Software Program Manual

Revision 2

Non-Proprietary

January 2018

REVISION HISTORY

Revision	Date	Page	Description
0	November 2014	All	First Issue
1	February 2017	xiii	Reflected the changed section title
		xvii and xviii	Added the acronyms, ICC, LCL, and SPDS
		xviiiii	Modified the abbreviation, QAM to reflected the supplemental response to RAI 261-8253, Question 07.01-30
		xviii	Deleted the acronym, SCI
		4	Added the corresponding reference
		12	Added the description for the analysis of pre-existing software to reflect the response to RAI 261-8253, Question 07.01-29
		23	Replaced the description for the software life cycle model to reflect the supplemental response to RAI 261-8253, Question 07.01-31
		70	Added the description for the scope of testing to reflect the supplemental response to RAI 261-8253, Question 07.01-32
		75	Modified the description for the clarification for the criteria in the section that will be followed by the digital safety I&C systems to reflect the response to RAI 261-8253, Question 07.01-33
		78	Fixed a typo of the reference number
		83	Modified the section title to reflect the response to RAI 261-8253, Question 07.01-33
		84	Added the description for the vulnerability assessment to reflect the response to RAI 261-8253, Question 07.01-33

Revision	Date	Page	Description
		96	Deleted the unnecessary item, “FDR affected” in the typical software change request form
2	January 2018	84	Modified the description for the vulnerability assessments to reflect the revised response to RAI 261-8253, Question 07.01-33 (Rev. 1)

ABSTRACT

This technical report (TeR) provides the software engineering process for digital computer-based instrumentation and control (I&C) systems of the APR1400.

This report describes the processes which ensure the reliability and design quality of the software throughout its entire life cycle.

By implementing the processes in this report, the digital I&C system software achieves the following:

- Desired level of quality and reliability required for nuclear power plants (NPPs)
- Safety-related I&C functions for protecting and securing the safe operation of the NPPs
- Satisfactory conformance to nuclear codes and standards

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SHEETS

ACRONYMS AND ABBREVIATIONS

ALMS	acoustic leak monitoring system
AMI	acoustic leak monitoring system
ANSI	American National Standards Institute
BTP	branch technical position
CBP	computer-based procedure
CCB	configuration control board
CD-ROM	compact disk – read only memory
CEA	control element assembly
CEAC	CEA calculator
COTS	commercial off-the-shelf
CPC	core protection calculator
CPCS	core protection calculator system
CPEN	computer program error notification
CPP	CEA position processor
CPU	central processing unit
CR	comment record
CRC	cyclic redundancy check
DC	design certification
DCN-I	data communication network – information
DCD	design control document
DCS	distributed control system
DIS	diverse indication system
DPS	diverse protection system
DRCS	digital rod control system
DS	design specification
DT	design team
EGS	engineering group supervisor
EPRI	Electric Power Research Institute
ESCM	ESF-CCS soft control module
ESF	engineered safety features
ESF-CCS	engineered safety feature – component control system
ESFAS	engineered safety features actuation system
FAT	factory acceptance test(ing)
FIDAS	fixed in-core detector amplifier system

FMEA	failure modes and effects analysis
FWCS	feed water control system
GC	group controller
GP	general purpose
I&C	instrumentation and control
I/O	input/output
ICC	inadequate core cooling
IEEE	Institute of Electrical and Electronics Engineers
IPS	information processing system
ITA	important-to-availability
ITP	interface and test processor
ITS	important-to-safety
IVMS	internal vibration monitoring system
KEPCO	Korea Electric Power Corporation
KEPCO E&C	KEPCO Engineering & Construction Company, Inc.
KHNP	KEPCO Engineering & Construction Company, Inc.
LAN	local area network
LC	loop controller
LCL	local coincidence logic
LDP	large display panel
LPMS	loose parts monitoring system
MTP	maintenance and test panel
NAPS	nuclear application programs
NIMS	NSSS integrity monitoring system
NIST	National Institute of Standards and Technology
NPCS	NSSS process control system
NPP	nuclear power plant
NRC	U.S. Nuclear Regulatory Commission
NSSS	nuclear steam supply system
OM	operator module
P-CCS	process – component control system
PCS	power control system
PHA	preliminary hazards analysis
PM	project manager
PPS	plant protection system
QA	quality assurance

QAM	APR1400 DC quality assurance manual
QIAS-N	qualified indication and alarm system – non-safety
QIAS-P	qualified indication and alarm system – P
RCPVMS	reactor coolant pump vibration monitoring system
RG	regulatory guide
RPCS	reactor power cutback system
RRS	reactor regulating system
RTM	requirements traceability matrix
SAT	site acceptance test(ing)
SBCS	steam bypass control system
SC	safety-critical
SCM	software configuration management
SCMP	software configuration management plan
SCR	software change request
SDD	software design description
SDL	serial data link
SDN	safety system data network
SDOE	secure development and operational environment
SDP	software development plan
SInstP	software installation plan
SIntP	software integration plan
SMP	software management plan
SOE	sequence of event
SOMP	software operation and maintenance plan
SPADES+	safety parameter display and evaluation system plus
SPDS	safety parameter display system
SPM	software program manual
SQAP	software quality assurance plan
SRS	software requirements specification
SSP	software safety plan
STP	software test plan
STrngP	software training plan
SVVP	software verification and validation plan
SysRS	system requirements specification
TER	test exception report
TeR	technical report

V&V	verification and validation
VT	V&V team

Page intentionally blank

1. INTRODUCTION

1.1 Purpose

The purpose of this technical report (TeR) is to specify a systematic approach to be used for the software engineering process of digital computer-based instrumentation and control (I&C) systems. This report provides generic guidance for the software program plans based on the Branch Technical Position (BTP) 7-14, “Guidance on Software Reviews for Digital Computer-Based Instrumentation and Control Systems” (Reference 1).

1.2 Scope

This report is intended to be applied to the application software of digital computer-based I&C systems. This report defines the software engineering process to be followed in creating and revising software of the I&C systems. The software engineering process and outputs are described in each section separately.

This report is applied to the design, production, and maintenance of software for the I&C systems. The software life cycle is implemented, operated, and maintained based on this report.

This report does not describe the following:

- Platform software engineering
- Human factors verification and validation (V&V)
- Hardware engineering

This report contains the following basic elements:

- Software management plan (SMP): The SMP is the basic governing document for the entire development effort. Project oversight, control, reporting, review, and assessment are all carried out within the scope of the SMP.
- Software quality assurance plan (SQAP): The SQAP describes the process and practice of developing and using software. The SQAP addresses software classification, software categories, software development process, software management, documentation, standards, practices and conventions, review requirements, problem reporting, and other software quality issues.
- Software verification and validation plan (SVVP): The SVVP describes requirements for the V&V process to be applied to software of the I&C systems. The SVVP addresses overview of V&V, V&V principles, V&V process, V&V testing strategy, and V&V activities.
- Software configuration management plan (SCMP): The SCMP describes the methodology used in managing the configuration control of software of the I&C systems.
- Software development plan (SDP): The SDP provides the necessary information on the technical aspects of software development that are required to be performed by the design team in order to carry out the project.
- Software integration plan (SIntP): The SIntP describes integration of software elements, the hardware/software integration process, and testing the result of the integrated product.
- Software installation plan (SInstP): The SInstP describes the process for installing the software product to the target hardware.
- Software training plan (STrngP): The STrngP describes the process for training the operators of the software.
- Software operation and maintenance plan (SOMP): The SOMP specifies the requirements for the operation and maintenance of computer software utilized for the systems after delivery to the user or customer.

- Software safety plan (SSP): The SSP describes the process and measures to maintain and improve the safety of the software.
- Software test plan (STP): The STP describes the process and measures to test intermediate, integrated, and/or final software products.
- Secure development and operational environment (SDOE): The SDOE describes the process and measures to ensure that the software and its development environments are protected from inadvertent operation and undesirable behavior of connected systems.

Separate documents do not need to be produced for each of these plans, for each system. Also, information to be produced may appear in documents with different titles, or may appear in an electronic database.

The generation and implementation of the SInstP, the STrngP, and the SOMP is under the responsibility of utility.

1.3 Overview

The software developers shall first become familiar with this report which is the guideline for all activities related to software.

The design team shall determine the class and category of all software to be used for the I&C systems as described in this report. The design team shall identify the applicable standards which must be followed for those classes and categories of software. The software tasks and responsibilities are outlined in this report based upon software classification and categorization.

The software tasks required to be performed in each software life cycle phase are described in this report. The narrative description, along with the corresponding tables, figures, and sheets, assists in making the required decisions concerning the appropriate tasks to be performed and the responsibility of the tasks.

In addition, the documents which must be produced for each software life cycle phase are discussed in this report. Some content of a document may appear in other document(s) and a common document for the project may cover several systems. However, the required content must be provided.

2. GENERAL PROCESS

The management and control of software applies to computer software and associated documentation developed or used for applications. Software shall be developed, acquired, procured, controlled, and maintained in accordance with this report.

2.1 Software Classification and Categorization

The software classification is based on the functionality and importance related to safety. The software that is developed and/or used within the I&C systems is assigned to one of the following classes:

- Safety-critical (SC; Protection): Software whose function is necessary to directly perform reactor protection system initiation actions, engineered safety features actuation system (ESFAS) control actions, and safe shutdown control actions.
- Important-to-safety (ITS): Software whose function is necessary to directly perform diverse protection system (DPS) control actions, software that is relied on to monitor or test SC functions, or software that monitors plant critical safety functions.
- Important-to-availability (ITA): Software that is relied on to maintain operation of plant systems and equipment that are critical to operate the plant.
- General purpose (GP): Software that is not assigned to SC, ITS, or ITA class.

This report makes distinctions regarding the methods applied to each class. Specific parts of the software in a single system may be assigned to different classes. Each part of the software has an assigned class. The tasks required for each class of software are described in Table 4-3.

This report also makes distinctions regarding methods applied to each of the following categories of software:

- Original, developed for the APR1400
- Existing, to be modified
- Existing, not to be modified

Every software item is assigned to one of the above categories. Software in several categories may be included in each I&C system. The tasks required for each category of software are described in Table 4-2.

2.2 Software Life Cycle

This report uses a software life cycle model consistent with RG 1.152 (Reference 4) and RG 1.173 (Reference 11). The software life cycle includes the following phases:

- Concept
- Requirements
- Design
- Implementation
- Test
- Installation and checkout
- Operation and maintenance (including retirement)

Figure 2-1 shows the relationship between each phase of the software life cycle and the software life cycle plans for the SC class software. The numbers in Figure 2-1 represent the section numbers in this report. For the other software classes, some of the plans in Figure 2-1 are not prepared. Refer to Table 4-3 for the software life cycle plans for each software class.

2.3 Indoctrination and Training

Personnel involved in software engineering of the I&C systems shall be trained as described in this report. Such training records shall be prepared and maintained in accordance with the requirements of the QAM (Reference 32).

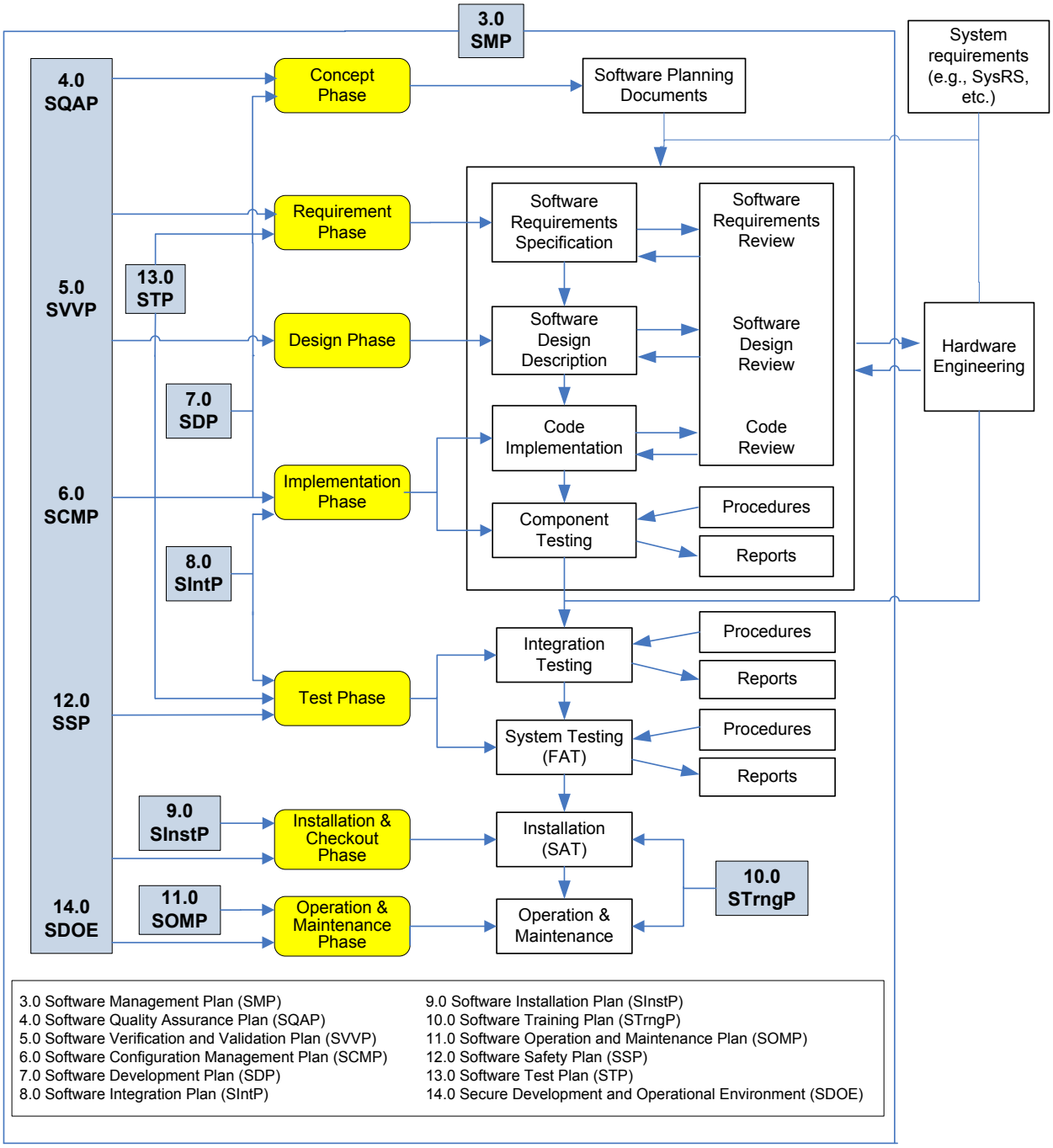

Figure 2-1 Overview of Software Life Cycle Plans and Engineering Process

2.4 Organization

The technical activities of the software engineering process are organized into the following two teams:

- Design team: The design team develops or obtains software of the I&C systems based on the system requirements. The design team develops the requirements, design, and code of original software.
- V&V team: The V&V team performs V&V activities on the software product. The V&V team is composed of competent individuals who have not performed the software design for the project. The individual responsible for the V&V is independent of the person responsible for the design.

Each V&V team for SC and ITS class software shall be independently organized of those responsible for the design. Each V&V task for SC class software shall be performed by the independent V&V team. Each V&V task for ITS class software shall be performed by the independent V&V team, or by the member of the design team who is not directly involved in the software design.

Each V&V team for ITA and GP class software is not required to be independently organized of those responsible for the design. Each V&V task for ITA and GP class software is performed by the member of the design team who preferably is not directly involved in the software design.

The organizational structure of KEPCO E&C NSSS division to control software life cycle process for the safety I&C systems is described in Appendix B.

3. SOFTWARE MANAGEMENT PLAN

3.1 Overview

A concise summary of the objective for software development, the product to be delivered, major work activities, major work products, major milestones, required resources, and master schedule and budget is provided in this section.

3.1.1 Project Summary

3.1.1.1 Purpose, Scope, and Objectives

The SMP is the basic governing plan for the entire development efforts. Project oversight, control, reporting, review, and assessment for systems are described in the SMP.

The SMP shall be prepared in accordance with BTP 7-14 (Reference 1) and meets IEEE Std. 1058 (Reference 24).

The contents of the SMP can be roughly divided into several categories: overview, project organization, managerial process plans, technical process plans, supporting process plans, and additional plans.

This SMP, which may be developed when deemed necessary by the management, provides the guidance and detailed information to manage software development of the digital I&C system.

3.1.1.2 Assumptions and Constraints

The SMP describes the following:

- The assumptions upon which the project is based.
- The external events upon which the project is dependent.
- The imposed constraints on project factors such as schedule, resources, technology, etc.

3.1.1.3 Project Deliverables

The SMP shall list the work products that will be delivered to the customer, the delivery dates, delivery locations, and quantities required to satisfy the terms of the project agreement.

3.1.1.4 Schedule and budget summary

The SMP shall provide the schedule for the software development process including system functions, activities, and tasks which take into account the precedence relations and the required milestone dates, schedules expressed in absolute calendar time or in increments relative to a key software development process milestone and the allocation of budget and resources to the various project functions, activities, and tasks.

3.1.2 Evolution of the Plan

The SMP shall specify the plans for producing both scheduled and unscheduled updates to the SMP, and also the mechanisms used to place the initial version of the SMP under configuration management and to control subsequent changes.

3.2 Project Organization

The SMP shall address organizational issues; internal organizational structure, boundaries and interfaces, and project responsibilities. Software project organization is described in Section 2.4.

3.2.1 External Interfaces

The SMP shall describe the organizational boundaries between the project and external entities.

3.2.2 Internal Structure

The SMP shall describe the internal structure of the project organization to include the interfaces among units of the software development team. In addition, the organizational interfaces between the project and organizational entities that provide supporting processes shall be specified.

3.2.3 Roles and Responsibilities

The SMP shall identify and state the nature of each major work activity and supporting process and identify the organizational units that are responsible for those processes and activities.

3.3 Managerial Process Plans

The SMP shall specify the project management processes for the project, and also include project start-up plan, risk management plan, project work plan, project control, and project closeout plan.

3.3.1 Start-up Plan

The SMP shall specify the estimation, staffing, resource acquisition, and training plan. These plans may be incorporated directly or by reference to other plans, depending on the size and scope of the project.

3.3.1.1 Estimation Plan

The SMP shall specify cost and schedule as well as methods, tools, and techniques used to estimate project cost, schedule, resources requirements, and associated confidence levels.

3.3.1.2 Staffing Plan

The SMP shall specify the number of staff by skill level, the project phases in which the numbers of personnel and the types of skills that are needed, and the duration of need.

3.3.1.3 Resource Acquisition Plan

The SMP shall specify the plan for acquiring the resources in addition to personnel and shall include a description of acquisition process, including assignment of responsibility for all aspects of resource acquisition, and shall also include acquisition plans for equipment, computer hardware and software, training, service contracts, transportation, facilities, and administrative services.

3.3.1.4 Project Staff Training Plan

The SMP shall describe the training needed to ensure necessary skill levels, and include types of training, numbers of personnel, entry and exit criteria for training, and the training method.

3.3.2 Work Plan

The SMP shall specify the work activities, schedule, resources, and budget details for the software project.

3.3.2.1 Work Activities

The SMP shall specify the various work activities to be performed in the software project. A work breakdown structure shall be used to depict the work activities and the relationship among work activities. Work activities shall be decomposed to a level that exposes all risk factors and allows accurate estimate of resource requirements and schedule duration for each activity.

3.3.2.2 Schedule Allocation

The SMP shall provide scheduling relationships among work activities in a manner that depicts time-sequencing constraints and illustrates opportunities for concurrent work activities.

3.3.2.3 Resource Allocation

The SMP shall describe a detailed itemization of resources allocated to each major work activity. Resources shall include numbers and required skill levels of personnel for each work activity.

3.3.2.4 Budget allocation

The SMP shall describe details of necessary resource budgets for each major work activity. The activity budget shall include estimated cost for personnel, travel, meetings, computing resources, software tools, special test and simulation facilities, and administrative support.

3.3.3 Control Plan

The SMP shall specify control procedures for requirements, schedule, budget, and quality. The SMP shall also specify procedures for reporting and metrics collection.

3.3.3.1 Requirements Control Plan

The SMP shall describe the control mechanisms for measuring, reporting, and controlling changes to the product requirements.

3.3.3.2 Schedule Control Plan

The SMP shall describe control mechanisms to measure the progress of work at the major and minor milestones, and also specifies the methods and tools to measure and control schedule progress.

3.3.3.3 Budget Control Plan

The SMP shall describe control mechanism to measure the work cost, and also describe methods and tools to manage the budget.

3.3.3.4 Quality Control Plan

The SMP shall specify the mechanism to be used to measure and control the quality of the work processes and resulting work products.

3.3.3.5 Reporting Plan

The SMP shall describe reporting mechanisms, report formats, and information flows to be used in communicating the status of requirements, schedule, etc., within the project and to entities external to the project.

- Internal reporting: within the design team
- External reporting: to auditors and regulators

3.3.3.6 Metrics Collection Plan

The SMP shall describe methods, tools, and techniques in collecting and retaining project metrics. Refer to Section 4.4.3.

3.3.4 Risk Management Plan

The SMP shall specify the risk management plan for identifying, analyzing, and prioritizing project risk factors. Risk factors that shall be considered include contractual risks, technological risks, risks caused by the size and complexity of the product, risks in the development and target environments, etc. throughout the life cycle of the project.

3.3.5 Project Closeout Plan

The SMP shall contain the plan to ensure orderly closeout of software project, and also include preparation of final report and analysis of project objectives achieved.

3.4 Technical Process Plans

The SMP shall specify the development process model, the technical methods, tools, and techniques to be used develop the various work products; plans for establishing and maintaining the project infrastructure; and the product acceptance plan.

3.4.1 Process Model

The SMP shall define the relationships among major project work activities. The following specifications shall be provided:

- Timing of major milestones
- Software project baselines
- Timing of project reviews and audits
- Work products of the software project
- Project deliverables

3.4.2 Methods, Tools, and Techniques

The SMP shall specify all of the methods, tools, and techniques that will be used to develop the software. For the methods, tools, and techniques, refer to Section 7.

3.4.3 Infrastructure Plan

The SMP shall specify the plan for establishing and maintaining the development environment (hardware, operating system, network, and software), policies, procedures, standards, and facilities required to conduct the software project.

3.4.4 Product Acceptance Plan

The SMP shall specify the plan for acceptance of the deliverable products generated by the software project, for objective criteria for acceptability of deliverable product, and for any technical processes, methods, or tools required for product acceptance.

3.5 Supporting Process Plans

The SMP shall contain plans or references for the supporting processes that span the duration of the software project.

3.5.1 Configuration Management Plan

The SCMP shall include the methods that will be used to provide configuration identification, control, status accounting, evaluation, and release management. Refer to Section 6.

3.5.2 Verification and Validation Plan

The SVVP shall include scope, tools, techniques, and responsibilities for the V&V work activities. Refer to Section 5.

3.5.3 Documentation Plan

The SQAP shall include plans for generating non-deliverable and deliverable work products, and also a list of documents to be prepared, the controlling template or standard for each document. Refer to Section 4.3.

3.5.4 Quality Assurance Plan

The SQAP shall assure that the software project fulfills its commitments to the software process and the software product as specified in the requirements specification. It shall also indicate the relationships among the QA, V&V, review, audit, configuration management, system engineering and assessment processes, Refer to Section 4.

3.5.5 Reviews and Audits Plan

The SQAP shall specify schedule, resources, and methods and procedures to be used in conducting project reviews and audits. Refer to Section 4.5.

3.5.6 Problem Resolution Plan

The SQAP shall specify the resources, methods, tools, techniques, and procedures to be used in reporting, analyzing, and processing software problem reports generated during the project. Refer to Section 4.7.

3.5.7 Subcontractor Management Plan

The SQAP shall include plans for selecting and managing any subcontractors that may contribute work products to the software project. Refer to Section 4.10.

3.5.8 Process Improvement Plan

The SMP shall include plans for periodically assessing the project, determining areas for improvement, and implementing improvement plans.

3.6 Additional Plans

The SMP shall contain additional plans required to satisfy product requirements and contractual terms.

4. SOFTWARE QUALITY ASSURANCE PLAN

4.1 Introduction

4.1.1 Purpose

The SQAP describes the requirements and methodology to be followed by the responsible team in developing, acquiring, using, and maintaining software of digital computer-based I&C systems.

The design team shall document the appropriate software classification in its software requirements specification (SRS) and software design description (SDD).

The SQAP shall describe in more detail the requirements for software configuration management (SCM), V&V and coding standards to be used for the APR1400.

The SQAP shall be prepared in accordance with BTP 7-14 (Reference 1) and meets IEEE Std. 730 (Reference 16).

4.1.2 Scope

The SQAP is required for all software classifications defined for digital computer-based I&C systems: SC, ITS, ITA, and GP.

All software developed or used within a system shall be documented by class and category. This shall be performed by the design team and included in the SRS.

Software that is initially assigned to one software class can be reassigned to any other 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.

Existing software is software that has been created, but not under this report.

To qualify existing software for use under this report, the software must be evaluated by the design team to meet the following criteria:

- Existing commercial software may be used as SC or ITS class if it is qualified in accordance with EPRI TR-106439 (Reference 30). To qualify existing commercial software as ITA or GP class, the design team shall select applicable portions of EPRI TR-106439 to qualify the software.
- Existing NPP non-commercial software that has been actively used in a NPP may be used as the same class of software under this report provided it has been maintained under an acceptable quality plan with an active program for problem reporting and corrective action during the software life cycle. This software shall also have adequate design documentation, user documentation and well commented source code. This software shall be analyzed with the criteria for previously developed or sub-vendor software described in Section 5.3.3.2 of this report. 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 may be used if the review by the design team and the V&V team conclude that it is well organized and has adequate design documentation, user documentation, and source code commentary.

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 report. Modifications to this software may be made in accordance with prior documentation and code format.

Under the 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.

The SQAP describes the methodology by which all software and associated documentation is managed throughout the life cycle.

4.1.3 Software Development Process

The software development process for the I&C systems is shown in Figure 2-1, which shows the relationship between software and hardware, the process of software integration and testing, the design documentation produced, and the QA documentation required throughout the software life cycle.

Software quality is assured through the process of software reviews and testing at the different stages of development, and SCM during all phases of software development.

Software V&V activities are governed by SVVPs. Test procedures and test reports required are based on the level of the test and the class of the software.

4.2 Management

4.2.1 Organization

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

Software life cycle activities for the I&C systems shall be performed by the organization described in Section 2.4. The design team, V&V team, and QA team are involved in the execution of software QA tasks.

Management of the SQAP is overseen by the managers of design team and V&V team. Organizationally, verification of the implementation of QA requirements is performed by the QA team in accordance with the QAM.

The design team and V&V team shall ensure that software and associated documentation are developed in accordance with the standards specified in the SQAP. This includes ensuring that the coding standards (Section 4.4.1), testing standards established in the STP, and documentation standards (Section 4.3) are followed.

4.2.2 Tasks and Responsibilities in the Software Life Cycle

This section describes the specific tasks and responsibilities. Tasks are listed in the life cycle phase for which they will be performed. Typical tasks are: software design and development, software QA planning, verification reviews, audits, test planning, test execution, and test reporting. Tasks required are based on software category. Table 4-2 shows the software tasks for each category in each phase. Table 4-3 shows the division of responsibilities for document preparation and verification review activities between the design team and the V&V team. If required, the procedure for the activities shown in the Table 4-3 will be written as an internal document.

The following are some typical procedural actions that are performed to ensure traceability throughout the development and verification stages:

- The software design documents are dated and signed by the designer and the corresponding manager.
- Each original or revision of a software component written by a programmer is dated and signed by the programmer of the software component.
- The corresponding project software V&V report and software test documents are dated and signed by the author and the manager of the responsible team.
- Each software component is reviewed and documented.

4.2.2.1 Concept Phase

During this phase, SMPs, SQAPs, SVVPs, SCMPs, SDPs, and SSPs shall be prepared. SIntPs, SInstPs, STrngPs, SOMPs, STPs, and coding standards shall be prepared before they are to be used. For example, coding standards shall be prepared no later than the end of the design phase, and SInstP shall be prepared no later than the end of test phase.

4.2.2.2 Requirements Phase

During this phase, the SRS is developed. The SRS involves analyzing and documenting the requirements for the software. An SRS shall be developed in accordance with the system requirements which are pertinent to the software.

The design team shall be responsible for developing, maintaining, and updating its SRS. An SRS shall provide the detailed information sufficient to design the software. For software which has functions spanning more than one software class, the SRS shall be divided to describe software requirements for the software in each class. The SRS shall be developed in accordance with Section 4.3.1 of this report.

Each SRS shall be reviewed by the team indicated in Table 4-3. The review shall ensure that the system requirements are properly reflected in the SRS. The review of each SRS shall be performed in accordance with Section 4.5.1.

A prototype may be devised and coded in this phase following the management decision. All the activities related to the prototype development must end at a time before the start of or during the design phase. Further details regarding the prototype activities are provided in Section 4.2.2.3.

4.2.2.3 Design Phase

The design team shall be responsible for developing, maintaining and updating the SDD for each software unit. Each SDD shall be traceable to the requirements set forth in the SRS, and shall include details sufficient to begin coding in the implementation phase. All SDDs shall be developed in accordance with the requirements of Section 4.3.2.

Each SDD shall be reviewed by the team indicated in Table 4-3. The review shall ensure that the software requirements identified in the SRS are properly reflected in the SDD. The review of SDD shall be performed in accordance with Section 4.5.2.

During this phase, prototype software may be developed to prove a new principle or to help further define the functional design. As such, prototype software has a different (and usually shorter) software life cycle than the other categories of software. Specifically, prototype QA tasks shall include:

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

4.2.2.4 Implementation Phase

Original software development and modifications to existing software shall begin with the coding by the design team in accordance with the appropriate coding standards listed in Section 4.4.1. Existing software, which has been qualified as described in Section 4.1.2, may be integrated and tested during this phase.

The code review shall be performed by the team identified in Table 4-3. Details of the code review are described in Section 4.5.3.

Component testing shall be performed in accordance with the STP.

Component testing consists of module testing and unit testing.

The responsibility for component (module/unit) testing is assigned to the design team and V&V team as shown in Table 4-3. Module test cases, procedures, and reports are only required for SC class software. Unit test procedures and reports are only required for SC and ITS class software.

4.2.2.5 Test Phase

During this phase, the components of a software product are evaluated and integrated into the hardware, and the software product is evaluated to determine whether or not requirements have been satisfied.

During this phase, integration testing and system testing shall be performed.

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

- One method is to hierarchically assemble the software components and perform integration and system tests.
- Or, the test sequence can be performed in a series of expansions. This could be accomplished by continually adding successfully tested software components and testing after each addition until the complete software system is assembled and tested.

The test procedures and reports shall be documented in accordance with the STP, by the teams identified in Table 4-3.

The final software V&V report for the deliverable software shall be prepared during this phase. (Refer to Section 4.3.4.)

4.2.2.6 Installation and Checkout Phase

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 may consist of installing hardware, installing the program, and verifying that all components have been included.

An installation log shall be maintained during the installation and checkout process. After installation, the equipment and software shall be checked out in accordance with the site acceptance test (SAT) procedure.

The utility has the responsibility for this phase and associated requirement.

4.2.2.7 Operations and Maintenance Phase

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

- Remove identified 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, in conjunction with the SCMP, shall also be used to assist in the management of these activities and procedures.

The utility shall have the responsibility for this phase and associated V&V requirements.

4.3 Documentation

Documents required by the SQAP are listed in Table 4-2. This section provides guidance for the development of documents. All documents listed shall be made lifetime quality records in accordance with the QAM.

Software documentation shall be provided for all computer software to be used or delivered. The author of each software document shall provide trace information for each requirement or item of the document to each requirement or item of the upstream document. With this information, the V&V team performs a traceability analysis for the document in accordance with the SVVP.

4.3.1 Requirements Documentation

For the software engineering, documentation of system requirements and software requirements is required. The system requirements may be included in the system design documentation such as the design specification (DS) for the system, or be documented separately in the SysRS. The software requirements are documented in the SRS, based upon the documented system requirements.

The SRS is developed in accordance with RG 1.172 (Reference 10) which endorses IEEE Std. 830 (Reference 19). The SRS is used as the source document for design of the software, including:

- Description of major software components which reflect the software requirements
- Technical description of the software (i.e. control flow, data flow, control logic, data structures)
- Description of all interfaces and allowable ranges of inputs and outputs
- Any other design items which must be translated into code
- A description of the intended platform and programming language(s) expected to be utilized
- Data necessary for final implementation such as setpoints
- Abnormal conditions to be accommodated by the software shall be described, including resulting functional operations.
- Plant input signal transient conditions to be accommodated by this software shall be described.

Each software requirement documented in the SRS shall be traceable to the system requirement.

Each requirement in the SRS shall be defined such that its achievement is capable of being verified by the SVVP.

4.3.2 Software Design Documentation

The software design is documented in the SDD. The SDD is prepared in accordance with the IEEE Std. 1016 (Reference 22).

The purpose of the SDD is to depict how the software will be structured to satisfy the requirements of the SRS. 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 the 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 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 as documented in the SRS are to be implemented in code, and how the design is traceable to the requirements in the SRS.

4.3.3 Source Code Documentation

Source code documentation shall include source code listings which reflect the translated design representation into a programming language. Also, any associated documentation generated during the coding shall be included with the source code documentation.

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 component. All source code, whether developed or modified from existing software, shall be documented in accordance with the coding standards listed in Section 4.4.1.

4.3.4 Software Verification and Validation Documentation

Software V&V documentation shall include the SVVP and the software V&V reports, prepared in accordance with RG 1.168 (Reference 6) which endorses IEEE Std. 1012 (Reference 21).

4.3.4.1 Software Verification and Validation Plan

The SVVP shall identify the software items to be evaluated. The SVVP describes the V&V evaluation and reporting activities. Software review requirements and guidelines are described in Section 4.5 and Section 5.3. Software tests to be performed are described in a separate STP.

For original software to be developed, the SVVP shall be developed as early as during the concept phase.

For existing software to be modified, the SVVP shall include methods for verifying and validating modifications to this existing software.

The SVVP shall provide adequate planning for the following:

- Software V&V process for the various software classes;

- Software V&V process for existing software to be modified and to be used "as-is";
- Software V&V process for prototype software.

The SVVP shall also facilitate the tracking and recording of the hardware configuration pertinent to the software V&V process during all phases of the software life cycle.

4.3.4.2 Software Verification and Validation Report

The software V&V phase summary report shall be produced for each phase, describing the results of V&V tasks performed. It shall include the following:

- Description of V&V tasks performed,
- Summary of task results,
- Summary of anomalies and resolution, and
- Recommendations.

A V&V file shall be maintained by the V&V team throughout the software life cycle to document all V&V activities. It shall be an ongoing compilation of all test results, problem reports and corrective actions (Section 4.7), and review results (Section 4.5). This file shall form the basis for the development of the final software V&V report upon the installation and checkout phase.

Each final software V&V report shall be developed by the V&V team in accordance with the SVVP.

4.3.5 User Documentation

User documentation is prepared in accordance with IEEE Std. 26514 (Reference 28). 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 shall be developed to the extent practical during the test phase and delivered to the user during the installation and checkout phase.

User documentation shall consist of vendor documents and documents prepared as part of the project. Project prepared user documents shall be as follows:

- User 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 the design team.

4.3.6 Software Configuration Management Documentation

The design team shall be responsible for developing the SCMP in accordance with RG 1.169 (Reference 7) which endorses IEEE Std. 828 (Reference 17) to satisfy the unique requirements of their software development process.

An SCMP is described in Section 6. The SCMP shall provide the necessary means to enable traceability of software and documentation products following their origination.

4.3.7 Computer Code Certificate

The completion of preparing the final software V&V report is the basis for the issuance of a computer code certificate, Sheet 14.

A computer code certificate is only issued for SC class software.

The issuance of a computer code certificate allows the release of a configuration item for use in its intended application.

4.4 Standards, Practices, Conventions, and Metrics

4.4.1 Coding Standards

The design team shall provide guidance to ensure standardization, compatibility and maintainability of resulting software products. The guidance shall include a coding standard for each language, database, or software tool that allows author's discretion in establishment or use of convention. This requirement applies to the following typical programming languages:

- High Level Languages (e.g., C/C++, FORTRAN, etc.)
- Assembly Languages
- Database Query Languages
- Display Building Languages
- Functional Graphical Languages

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

- General
  - This area outlines general ideas and concepts used to guide the creation of software written under a specific language.
- Naming conventions
  - Filename extensions as far as how they are used to organize files.
  - Information pertaining to file organization within a system.
  - Variable naming.
- Internal documentation guidelines
  - Program identification header content, placement, type, quality, and quantity.
  - Revision history recording within each source file.
- Stylistic conventions
  - Issues that affect readability, such as indentation and use of white space.
- Use of specific language features
- Software tool usage guidelines
  - Information and use of automatic make facilities (e.g., automated code generator, disassembler, automatic data generator, etc.)
  - Appropriate compiler flag usage.
- Functions
  - Modularity
  - Naming

4.4.2 Software Testing Standards

Software testing methodologies, policies and practices shall be described in the STP. Specific format and content for test procedures, test cases, and test reports shall also be provided in the STP. Refer to Section 13.

4.4.3 Metrics

The following metrics are maintained:

- The errors discovered during component, and integration testing shall be identified through the use of test exception report (TER) forms so that all errors discovered can be resolved during the testing and that the number of errors discovered can be tracked. The overall goal is to identify a decreasing number and severity of errors as the testing progresses from component testing to SAT.
- The system testing errors shall be reported through the use of TERs and the number and severity shall be identified.
- Software errors discovered after system testing and before SAT shall be tracked through the use of TERs and the number and severity shall be identified.

4.5 Software Reviews

This section addresses the review requirements throughout the software life cycle.

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

The methodology of performing reviews shall be included in each SVVP. The reviews shall be performed by peers who have an equivalent knowledge of the topic and who satisfies the independence requirement as described in Section 2.4.

4.5.1 Software Requirements Review

The software requirements review shall examine the SRS to verify that it is clear, verifiable, consistent, modifiable, traceable, and usable during the operation and maintenance phase. The software requirements review shall include evaluation of the software requirements against the user's software application which is described in a higher level requirements document such as a system design specification or a SysRS.

Specific software requirements review items are described in Section 5.3.3 and shall be described in detail as necessary in the SVVP.

As a minimum, the following items shall be included:

- 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

4.5.2 Software Design Review

The software design review is to determine the acceptability of the detailed design as depicted in the SDD in satisfying the requirements of the SRS. Software design review items are described in Section 5.3.4 and shall be described in detail as necessary in the SVVP. The software design review shall be performed at the completion of the SDD and generally address the following items:

- Technical adequacy
- Completeness of design requirements
- Consistency
- Correctness of software design
- Traceability of software design to software requirements

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

4.5.3 Code Review

The objective of code review process is to verify:

- That the code implementation decisions are consistent with good software engineering practice based on expert opinion;
- That the code meets the coding guidelines and other requirements;
- The executable code listing against SDD to ensure that no errors were introduced during the code generation process.

4.5.4 Software Verification and Validation Plan Review

The SVVP shall be reviewed to evaluate the adequacy and completeness of the V&V methods defined in the SVVP.

4.5.5 Functional Audit

The functional audit is conducted prior to the software delivery to verify that all requirements specified in the SRS have been met.

4.5.6 Physical Audit

The physical audit is conducted to verify internal consistency of the software and its documentation, and their readiness for release.

4.5.7 In-Process Audits

The in-process audits of samples of the design are conducted to verify the consistency of the design, including:

- Code versus design documentation
- Interface descriptions in design documentation (hardware and software)
- Design implementations versus functional requirements
- Functional requirements versus test descriptions

4.5.8 Managerial Reviews

The managerial reviews are conducted to assess the execution of all of the actions and the items identified in the SQAP. The managerial reviews are performed by, or on behalf of, the management personnel having direct responsibility for the system.

4.5.9 Software Configuration Management Plan Review

The SCMP review is conducted to evaluate the adequacy and completeness of the configuration management methods defined in the SCMP.

4.5.10 Post-Implementation Review

The post-implementation review is conducted at the conclusion of the project to assess the development activities on that project and to provide recommendations for appropriate actions.

4.6 Test

Refer to Section 13.

4.7 Problem Reporting and Corrective Action

The purpose of a formal procedure of software problem reporting and corrective action is to ensure that all software errors and failures are acted upon promptly and in a uniform manner encompassing all project software. This procedure ties together with 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 report.

4.7.1 Problem Reporting

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 discrepancy found outside of the normal V&V test process. Table 4-1 illustrates the type of report required by each method:

Table 4-1 Problem Reporting Methods

Method	Report
Reviews	Comment Record (Sheet 1)
Tests	Test Exception Report (Sheet 15)
General Findings	Comment Record (Sheet 1)

The appropriate configuration identification data (refer to 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 a TER, each form shall include the 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 to the completed TER. Copies of all completed TER forms shall be distributed by the independent reviewer to the design team and managed by the V&V team. All TER forms shall also be placed in a lifetime records per the QAM. The steps which cause the discrepancy shall also be included on the TER form in order to reproduce the problem. These steps shall be noted as best as possible if the problem is not repeatable.

The extent of the retest shall be determined by the design team and V&V team based on the relative impact of the software change on the overall system operation. For SC and ITS class software, all changes require complete retest for the system or function, unless otherwise justified in writing including steps to ensure that new errors were not introduced.

4.7.2 Corrective Action

The design team and the V&V team shall establish, as a clear objective, the goal of resolving all test problems (via TERs) and review comments (via CRs) 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.

In addition, the design team shall adhere to the following corrective action methodology that:

- Problems are identified, evaluated, documented and, if required, corrected by the appropriate reporting mechanism (Section 4.7.1).
- Corrections or changes shall be controlled in accordance with the SCMP.
- Preventive actions and corrective actions are documented on the appropriate form and distributed to the design team.

Corrective actions shall be documented on TERs and CRs by the design team and shall be completed by the due date specified on the form.

4.8 Tools, Techniques, and Methodologies

During software development, a number of techniques will be used to help assure all software is designed, implemented, and documented in accordance with the 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 analysis and design techniques and methodologies helps to define and design systems from the "top-down", i.e., based upon the broad requirements of the end user. The data flow diagram describes only the functions themselves, without any mention of how or by what components they will be implemented. Structured methods also provide techniques and guidelines to aid in the software design. A structure chart breaks down functions into a sequence of tasks (corresponding to procedures or subroutines), with the input and output parameters shown. While the decomposition is left to the designer, the design methods recommend high cohesion with each software module (i.e., each software module performs only a single function) and simple coupling among the software modules.

The software development organization may use its own life cycle model provided that, within its work scope, the required activities and tasks described in this report are performed and the required outputs specified in this report are provided.

The use of automated tools for SCM shall be employed to the maximum extent possible.

4.9 Media Control

This section describes the methods and facilities used to protect computer program physical media from unauthorized access or inadvertent damage or degradation.

4.9.1 Media Identification

The portable media shall contain a serial number and description which lists the file(s) backed up on the medium. The media shall not be switched, renamed, or initialized without prior approval of the design team manager.

A log shall be maintained which cross references serial numbers to file(s) backed up. A log is not required if alternate methods are in place to identify software release version accurately.

4.9.2 Archival Requirements

A locked fire/water proof facility shall be installed and maintained at a separate premise from the computer facility to store all project software. This facility shall be able to accommodate the storage of all utilized types of media.

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

4.10 Supplier Control

This section describes the level of software QA measures to be applied to software supplied from parties other than the design team.

4.10.1 Existing Software

This report defines existing software as software which was previously developed to satisfy a general market need and may be considered for use on this project. The software may be subsequently modified prior to delivery, or it may be used "as-is".

Existing software includes commercial software which is integral to the delivered system and software which 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 which will be used in the safety I&C systems must meet the requirements established in EPRI TR-106439 and are based on the software's classification as follows:

- SC - All requirements
- ITS - All requirements, except as modified or reduced by the design team in the following areas:
  - Software service experience data
  - Software inspection and testing techniques
  - Software problem reporting
  - Software change control

For existing software which is modified prior to delivery, all requirements specified in the SQAP for original software shall be in effect for at least the modifications, except that modifications may conform to coding standards already in use.

The minimum V&V activities applicable for modifications to existing software are software modifications requirements review, software modifications design review, program modification documentation review, and testing.

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 in accordance with Section 4.1.2.

Once qualified for use, the software shall fall under the SCMP. Once installed, the software shall meet the following requirements:

- V&V during installation and operation per the SVVP,
- Configuration management during installation and operation per the SCMP,
- Documentation including: STPs, test procedures, software V&V reports, and user manuals,
- Problem reporting and corrective action procedures, and
- Records of delivered documents and software

4.10.2 Sub-Contracted Software/Services

Original software, which is developed by a contractor and purchased, shall adhere to the QA requirements for original software specified in the SQAP. This applies regardless of whether the software will be modified or not.

Where available, third party contractors who supply software are required to provide feedback of problems encountered by the supplier of other users of similar software. Where available, this feedback information shall be supplied automatically without request from the project. Also, the design team has the responsibility for informing the supplier of any problems encountered by the user in the use or maintenance of the software.

Additional requirements for subcontracted software and services are as follows:

- Software and services must be procured from approved suppliers in accordance with the QAM.
- Suppliers must have written QA policies which meet the principles and intent of the SQAP.
- Purchase orders shall require the supplier to make available documents which are evidence of compliance with the principles and intent of the SQAP.
- Purchase orders shall require the supplier to deliver adequate user documentation, test procedures and test reports.

4.11 Records Collection, Maintenance, and Retention

Records collection, retention, and maintenance shall be in accordance with the QAM. If required, documents listed in the SQAP shall be collected, maintained, revised and retained as lifetime records in accordance with the QAM.

4.12 Training

The managers of design and V&V teams shall ensure that there are sufficient, independent, technically qualified and trained resources to implement the requirements of the SQAP.

4.13 Risk Management

Risk management is included as part of the internal project review process.

The categories of risks that are reviewed include organization and interfaces, resources, technical work scope, schedule, and commercial/financial issues.

Risk that may occur in project execution shall be assessed during all phases of the software life cycle, and actions identified to mitigate identified risks. The mitigating actions shall be presented to project management.

Table 4-2 Tasks required for Software Categories

TS

Table 4-3 Software Tasks and Responsibilities

TS

5. SOFTWARE VERIFICATION AND VALIDATION PLAN

5.1 Purpose

The purpose of this section is to establish requirements for the software V&V process to be applied to the I&C systems. This section includes various V&V methodologies aimed to increase the software reliability and availability. Some of these methodologies employ systematic checks for detecting errors in the software, during the software development 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 software for the I&C systems. This SVVP complies with RG 1.168 (Reference 6) which endorses IEEE Std. 1012 (Reference 21).

The goals of the SVVP are to:

- Improve the software reliability and availability,
- Reduce costs by exposing errors as early as possible,
- Provide a systematic process of objectively evaluating the performance, and
- Demonstrate compliance with customer requirements, industry standards and licensing requirements.

V&V activities are integrated into the concept, requirements, design, implementation, test and installation and checkout phases described in Section 2.2. 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. V&V 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, test activities are performed. These activities determine that the operation of the software is consistent with the system requirements. Thus V&V activities are integrated with project activities from the beginning to the end.

5.2 Verification and Validation Overview

5.2.1 Organization

The organization is described in Section 2.4.

For SC and ITS class software, the V&V team is organized independently of the design team. The V&V team members shall have not been directly involved in the software design of the project.

For ITA and GP class software, the V&V team is not required to be organized independently of the design team. The V&V team members may have been involved in the software design of the project.

5.2.2 Master Schedule

The SVVP shall include the V&V schedule and required milestone delivery dates. This shall be developed in coordination with the V&V team leader.

5.2.3 Software Integrity Level Scheme

The V&V tasks, including tests, for specific software to be performed are determined by the classification of the software described in Section 2.1. The minimum V&V tasks for each software class are identified in Table 5-1.

TS

5.2.4 Resources Summary

The V&V team shall have all the same working resources (i.e.: facilities, tools, securities, access and documents) as the members of the design team.

The V&V plan shall summarize the resources to perform the V&V tasks, including staffing, facilities, tools, man-hours, and special procedures such as security, access rights, and documentation control.

5.2.5 Responsibilities

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

The emphasis shall be placed on assuring that the software documentation detailing the software functional requirements and performance specifications are clear, accurate and complete.

The documentation shall be reviewed looking for omissions, inconsistencies, inaccuracies and errors of omission/irrelevant requirements. Some significant functional requirements may be identified and monitored as development progresses.

The emphasis shall be placed on full independent analysis of the software requirements and design documentation, independent testing and evaluation for SC class software.

The actual assignment of responsible team for software tasks is shown in Table 4-3.

The requirements and the implementation of design shall be evaluated to ensure that the resulting software operation is functionally correct and meets the performance objectives.

5.2.6 Tools, Techniques, and Methods

5.2.6.1 Tools

Part of the V&V planning process includes the selection of appropriate tools. Commercial automated tools for software V&V may be selected to complete the required V&V tasks.

5.2.6.2 V&V Core Activities

The following V&V core activities are applicable to all software:

TS

5.2.6.3 Requirements Traceability Analysis

Throughout the software life cycle, a requirements traceability matrix (RTM, Sheet 10) is maintained and requirements traceability analyses are performed for each software system. The author of each software document to be used for the requirements traceability analyses is responsible for providing trace information from each item in the document to the items in the upstream document. The V&V team is responsible for analyzing the trace information provided. The V&V team shall review the trace information for the adequacy and accuracy.

The use of automated tools for the requirements traceability analyses is recommended to the maximum extent possible.

5.2.6.4 Risk Analysis

At the concept phase, a risk analysis is performed to identify risks and to provide recommendations to eliminate, reduce, or mitigate the risks.

At each of the succeeding phases, a risk analysis is performed to review and update previous analysis results and to provide recommendations to eliminate, reduce, or mitigate the risks.

Sheet 2 is a typical form for risk analyses based on the NUREG/CR-6430 (Reference 13).

5.3 Life Cycle Verification and Validation

This section describes the V&V tasks during the software life cycle.

Table 5-1 describes the V&V tasks required for each software class during software development process.

5.3.1 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:

- SVVP: Generate the SVVP outlining resources and schedule of the V&V activities.
- 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.
- 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.
- Review Support: Support management and technical reviews (e.g., software requirements review, software design review, code review, etc.). Identify key review support milestones in the SVVP and schedule V&V tasks to meet milestones. Establish methods to exchange V&V data and results with the design team.

5.3.2 Concept Phase V&V

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

The SVVP, including schedule and personnel requirements shall be developed at this time. The SVVP must reflect the V&V activities to be performed during each phase. Any specific tools to be used must be stated in the SVVP.

The concept phase checklist must be completed (Sheet 3).

The actual reporting of the concept phase V&V activities may be combined with the requirements phase.

TS

TS

5.3.3 Requirements Phase V&V

The intent of verifying the system requirements and software requirements is to ascertain that the requirements are correct, complete, accurate, testable and consistent.

Requirements phase V&V includes software requirements review.

The principal purpose of a requirements document is:

- To clearly define the objectives and needs of the software design and development process. Both the designer and the user must be able to understand and perform a meaningful assessment of the software.
- To serve as a means against which an implementation can be validated and the intermediate steps can be verified.

The goal of V&V activities during this phase is to ensure that the requirements documents do indeed serve the above purpose.

Extraneous issues which are not requirements shall not be in the SRS or it shall be explicitly stated that they are for information only.

TS

TS

5.3.4 Design Phase V&V

The purpose of the design phase V&V is to ascertain that the SDD represents a faithful translation of the SRS before the design is committed for implementation. Documentation of this V&V shall consist of completing the appropriate section of Sheet 5.

TS

5.3.5 Implementation Phase V&V

The purpose of the implementation phase V&V is to ascertain the implementation documents are clear, understandable, logically correct and a faithful translation of the SDD. 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 the design. In general the V&V activities during this phase are oriented towards evaluating the following:

- Does the implementation satisfy the SDD?
- Does the implementation follow established design standards?
- Does the implementation follow established documentation standards?
- Does the implementation serve production, test, use, transfer and other needs that motivated its creation?
- What is involved in testing the actual resulting product?

The implementation phase V&V activities also include the inspection of the source code program to assess the performance of either derived or resulting assembly. Also, the V&V team prepares or reviews detailed test procedures in preparation for the next stage of component testing. Component testing in this phase is to test that the executable code of each program behaves as specified. Test procedures and

reports for component testing shall be documented by the teams identified in Table 4-3. Component tests shall be conducted by the teams identified in Table 4-3.

TS

TS

5.3.6 Test Phase V&V

The overall objective of testing is to find faults in the software. The test phase covers integration testing and system testing. The V&V engineer ensures that the software requirements and system requirements allocated to software are satisfied by execution of the software.

Integration testing is to test that the software components, integrated together with the target hardware and pre-developed software, meet the requirements specified in the SRS. The testing is also to find errors in the software, hardware, and pre-developed software interfaces, errors in handling stress conditions, timing, fail-safe features, error conditions, and error recovery. Test procedures and reports for integration testing shall be documented by the teams identified in Table 4-3.

System testing is to test that the entire executable code meets the system requirements. System testing shall be conducted, per the STP, during this phase in the development environment when all of the software components have been integrated by design team. Test procedures and reports for system testing shall be documented by the teams identified in Table 4-3. Integration and system tests shall be conducted by the teams identified in Table 4-3.

All user documentation shall be developed during this phase in accordance with Section 4.3.5. The user documentation shall be reviewed by the V&V team.

TS

TS

5.3.7 Installation and Checkout Phase V&V

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.

Typical V&V tasks for the installation and checkout phase are:

- Review installation procedures and user manuals to ensure they are complete and correct.
- Complete the installation and checkout phase V&V checklist (Sheet 9).
- Records of all V&V activities shall be documented.

The installation and checkout phase V&V is the responsibility of the utility.

5.3.8 Operation and Maintenance Phase V&V

The operation and maintenance phase V&V is the responsibility of the utility.

Software may have to change in this phase for the following reasons:

- 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.

Typical V&V tasks for the operation and maintenance phase are:

- Evaluation of new constraints, proposed change assessment, and operating procedures
- SVVP revision
- Anomaly evaluation
- Migration assessment
- Retirement assessment

5.4 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).

5.4.1 V&V Phase Summary Report

This report is separately produced for each software life cycle phase until the test phase. Each report shall summarize the results of V&V tasks performed during each phase.

This report shall contain the following:

- Description of V&V tasks performed
- Summary of task results
- Summary of discrepancies and resolution
- Assessment of software quality
- Recommendations

5.4.2 Final V&V Report

This report shall be issued at the end of the project to summarize and document the V&V activities performed throughout all software development processes.

The report shall include:

- Summary of life cycle V&V tasks
- Summary of task results
- Summary of discrepancies and resolutions
- Assessment of overall software quality
- Recommendations for enhancements
- Code certificate

5.5 Verification and Validation Administrative Procedures

5.5.1 Anomaly Reporting and Resolution

Any discrepancies detected during any phase of the V&V process shall be immediately brought to the attention of the design team. 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.

5.5.2 Task Iteration Policy

If there are any changes (e.g., design changes, revision of approved document, etc.), the V&V tasks must be identified for the effects of the changes, and re-performed. The tasks must be identified in the reports identifying the rationale and the results of the tasks. This information shall be documented in the V&V reports to be produced.

5.5.3 Deviation Policy

If any deviation from the reviewed and approved V&V task plan is required, the change must be identified, rationale for the change provided, and a determination of effect on software quality provided. Any deviation must be approved by the V&V team leader and management.

5.5.4 Control Procedures

Procedures for V&V and software development in this report provide the controls for the activities associated with these efforts.

5.5.5 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 SVVP.

5.6 Verification and Validation Test Documentation

Refer to Section 13.

Table 5-1 V&V tasks required for Software Classes during Software Development Process

TS

6. SOFTWARE CONFIGURATION MANAGEMENT PLAN

6.1 Introduction

The SCMP shall be prepared in accordance with RG 1.169 (Reference 7) which endorses IEEE Std. 828 (Reference 17).

6.1.1 Purpose

The purpose of this section is to describe the methodology used in managing the control of software for the systems.

This control will be accomplished by:

- Maintaining historical records of software as it is developed and maintaining the relationship between that software and the corresponding functional documents.
- Controlling changes to software
- Recording and reporting change processing and implementation status.

The SCM forms described in this section are typical. Details will be provided in the SCMP. Systems may employ automatic configuration management software to support the requirements of this section. Such packages may produce documentation that is different in format but meets the intent of the forms in this section.

The SCMP shall establish the objectives that it intends to satisfy in order to support the realistic needs of the software life cycle. The following objectives shall be considered in developing SCMPs:

- To record and document the work in progress on each software item to facilitate understanding of current project status.
- To identify all software codes and data associated with a system, including revision level, completion status, test status and history.
- To maintain the association among software documents, code and data.
- To identify sets of software items that compose the system (baselines), test status and history, and readiness for release.
- To maintain the status of each released baseline, customer lists, and associated software problem reports.
- To support the evaluations of impact of a software problem on other baselines and customers.
- To maintain the association among software problem reports, change reports and the affected documents sections, lines of code and data items.
- To implement the appropriate controls and approvals for any changes to the software configuration.
- For safety critical software, identify all personnel with a given software item, problem report, or software change.
- To document the criteria for the generation of releasable baselines.
- To assure that designated prior revisions of individual software items and baselines can be reconstituted in the future.
- To backup software (completed and in progress) to protect against disaster.
- To plan for controlling access to computer software and protecting against software viruses.

6.1.2 Scope

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. Original software and modifications to existing software shall follow the configuration requirements for all life cycle phases. Existing software which is not modified shall only fall under configuration control procedures upon its introduction into the software systems.

6.2 Management

6.2.1 Organization

All SCM activities are performed by the design team. The design team leader is responsible for the implementation of adequate measures to manage and control the software configuration in accordance with the SCMP. A configuration control board (CCB) sets out the ground rules and policies of configuration control. The CCB members include the managers and leaders of the design team and the V&V team.

The system administrators are responsible for performing ongoing backup of work in progress for the system periodically in addition to the private backup by each engineer. The system administrators are responsible for developing procedures appropriate to their development facilities to meet the requirements of this plan.

6.2.2 Software Configuration Management Responsibilities

The design team leader is responsible for the implementation of adequate measures to manage and control the software configuration in accordance with the SCMP.

The librarian maintains all controlled software items as well as all control records. The librarian also maintains multiple backups of these items at alternate locations to assure sufficient disaster recovery capability. The librarian may be a member of the design team, but reports to the design team manager.

A single librarian may be used for more than one system under the cognizance of more than one manager.

A system administrator, who may also be the librarian, is responsible for performing ongoing backup of work in progress for the system.

6.2.3 Applicable Policies, Directives, and Procedures

The SCMP shall identify any external constraints placed on the SCMP by other policies, directives, and procedures, and state the impact and effect on the SCMP for each.

6.2.4 Management of the Software Configuration Management Process

The organization for the SCM process is described in Section 6.2.1.

The anticipated cost of the SCM process is included in the anticipated cost of the software development process.

The V&V team performs reviews for independent surveillance of SCM activities to ensure compliance with the SCMP.

6.3 Software Configuration Management Activities

6.3.1 Configuration Identification

Throughout the software life cycle, all versions of all software items shall be tracked along with their associations. For example, the association of a particular version of a code item and related design and requirements documents, test reports, etc. will be maintained in lifetime software records in accordance with the QAM. The association of software items with each particular software baseline shall also be maintained in lifetime software records.

6.3.1.1 Identifying Configuration Items

All software and documentation shall be uniquely identified by a system name, number, and corresponding revision date, appropriate category and appropriate software classification. Reviews, audits, problem reports, and test reports shall specifically identify the system name, number, revision number, and revision date to minimize confusion. This identification structure shall also have the ability to track resolution of test results and subsequent retesting of software components and integrated systems.

This configuration identification data shall be defined as part of the baseline at the completion of each major phase of the software project. In order for the baseline to change, it must incorporate all approved changes and represent the most recent approved software configuration.

Appropriate baselines shall be defined at control points within the project life cycle in terms of the following:

- The event that creates the baseline
- The items that are to be controlled in the baseline
- The procedures used to establish and change the baseline
- The authority required to approve baselined documents

6.3.1.2 Naming Configuration Items

Each software item shall have a numbering method that shall uniquely identify the item and its versions. The configuration identification scheme will be consistent with the numbering method for all software items. Software documents are identified by character string appearing on the cover sheet of each document. The software identifier shall be described in the SCMP.

6.3.1.3 Acquiring Configuration Items

The SCMP shall identify the controlled software libraries and describe how the baselined software products are to be controlled.

6.3.2 Configuration Control

Following the original baseline after the Implementation Phase, software and documentation shall be placed under configuration control by the design team. Configuration change control by the design team shall continue up to the point when the software is no longer expected to be changed. Software configuration control shall include limited access to master copies of documentation and software in either hard copy or digital form. Access to software and documentation shall be controlled by the librarian.

After the software is "system level certified" by the design team, it shall be delivered for integration testing.

The SCMP shall address how software changes will be managed. Software configuration control shall be performed in a single software environment or in multiple software environments such as the following:

- The development environment is where all software is written, changed, and tested, and controlled by the librarian.
- The control library is where all master copies of software components reside for integration test purposes, and are controlled by the librarian.
- The platform or FAT environment is where all software will be tested and integrated in preparation for final FAT on the deliverable equipment.

Changes to software shall be formally documented. This documentation shall contain a description of the change, the rationale for the change, and the identification of affected software element(s). Software V&V shall be performed on the change, in accordance with the SVVP, to ensure the necessary traceability, consistency, and correctness of the change.

Details of this control procedure shall be given in the SCMP. (The methods of configuration control may vary from those described above if an automated configuration management package is used. In this case the SCMP will describe the procedure from the programmer and librarian perspective.)

6.3.2.1 Requesting Changes

The SCR is the mechanism by which change requests are approved by the design team manager and presented to the librarian. This action allows a developer to check out software/documentation from the SCM controlled libraries. An SCR may be generated from an action documented in a TER. Refer to Section 4.7 for a description of problem reporting. The mechanism for requesting authorization is to present the SCR to the design team manager and request approval for work to begin. The SCR form shown in Sheet 11 is to be used.

The design team manager is responsible for assigning personnel to perform and develop requirements and procedures for configuration control.

6.3.2.2 Evaluating Changes

The design team leader evaluates proposed software changes to determine the impacts of the changes.

6.3.2.3 Approving or Disapproving Changes

The CCB determines whether the proposed software changes are feasible and appropriate or not. The CCB decides approval of each proposed change.

6.3.2.4 Implementing Changes

The design team implements approved software changes. The information for the implementation is recorded in the SCR and the software release history.

The V&V team performs V&V for the change implementation of SC and ITS class software, and the results are documented in the V&V reports.

6.3.3 Configuration Status Accounting

A software configuration status control log shall be maintained by the design team for a record of the revision history and current status of each controlled software element. This log shall include:

- Identification of the approved baseline configuration
- Status of proposed changes
- Status of approved changes

An automated configuration management system may be incorporated. If this is the case, the features of the configuration log described above shall be included in such a tool.

6.3.4 Configuration Audits and Reviews

Generally, there are a physical configuration audit and a functional configuration audit of configuration items prior to the release of a product baseline or an updated version of a product baseline. The physical configuration audit portion of the audit consists of determining that all items identified as being part of the configuration are present in the product baseline. The functional configuration audit portion is similar, in that someone acknowledges having inspected or tested each item to determine that it satisfies the functions defined in the specifications or contract for which it was developed.

The audits and reviews shall define ways to ensure that established configuration management procedures are followed.

6.3.5 Interface Control

Interface control is handled in the same manner as other types of software or documentation. Any difference between the SQAP and the SCMP must be resolved prior to the establishment of any baselines.

The SCMP shall identify the external items to which the project software interfaces.

6.3.6 Subcontractor/Vendor Control

Subcontractor/vendor control activities incorporate items developed outside the project environment into the project configuration items. Included are software developed by contract and software acquired in its finished form. Special attention shall be directed to these SCM activities due to the added organizational and legal relationships.

The SCMP shall define the activities to incorporate the externally developed items (subcontracted and acquired software) into the project configuration items and to coordinate changes to these items with their development organizations.

6.3.7 Release Management and Delivery

The software release history shall be documented. This will enable the configuration management program to maintain an accurate status of the software at all times and enable the reconstruction of a given release at any time in the future. The document will include the software version and indicate which components are affected in the revision.

6.4 Software Configuration Management Schedules

The project schedule shall include the SCM schedules.

The following are the major milestones or baselines:

- Configuration baseline based on available baseline software
- Intermediate baseline with requirements for major components
- Requirements implemented (ready for test and error correction)
- Test and error correction completed
- Final software release

6.5 Software Configuration Management Resources

The SCMP shall address the environment, infrastructure, software tools, techniques, equipment, personnel, and training necessary for the implementation of the specified SCM activities.

6.6 Software Configuration Management Plan Maintenance

The responsibilities for monitoring SCMP are defined in Section 6.2.

The SCMP shall be updated as deemed necessary by the design team.

Changes to SCMP shall be documented and approved to the same level as the original issue of the document and transmitted to each engineer for software configuration control effectiveness.

7. SOFTWARE DEVELOPMENT PLAN

7.1 Introduction

The SDP provides necessary information on the technical aspects for the software development that are required by the design team in order to carry out the project.

The design team shall follow the software life cycle model described in Section 2.2, which is in accordance with RG 1.173 (Reference 11) that endorses IEEE Std. 1074 (Reference 25) for information on life cycle processes.

This section provides the guidance and the detailed information to develop software of the digital I&C systems.

7.2 Life Cycle Processes

The SDP shall describe the software life cycle processes that will be used on the project. The SDP shall discuss the various processes that make up this life cycle. For each process, the SDP shall give the input information required in order to carry out the process, a description of the actions that must take place during the process, and the output information produced by the process. Since the output of one process is likely to be used as input to another, a data flow diagram will be appropriate.

TS

TS

7.3 Methods, Tools, and Techniques

The SDP shall describe the methods and techniques that will be used to develop the software, and the tools that will be used in connection with those methods and techniques. For the methods, tools, and techniques, refer to Section 4.8.

7.4 Standards

The list of all international, national, and company standards that must be followed in the project shall be founded in the section of the SDP.

7.5 Schedule and Milestones

The SDP shall provide the list of the technical milestones that must be met and describe what is expected at each milestone.

7.6 Technical Documentation

The SDP shall provide the list of the technical documents that must be produced during the software development and define milestones, baselines, reviews, authors and sign-offs for each document.

8. SOFTWARE INTEGRATION PLAN

8.1 Introduction

Software integration consists of three major phases: integrating the various software components together to form single programs, integrating this with the hardware, and testing the resulting integrated product. During the first phase, the various object components are combined to produce executable programs. These programs are then loaded in the second phase into test systems. The final phase consists of testing the integrated system, and these test results are created in the test phase of software life cycle.

This section provides the guidance and the detailed information to integrate the digital I&C system.

The SIntP shall include the system build document that describes precisely how the system hardware and software components are combined into the specific operational system. And it shall describe how the system is assembled, including hardware and software component names and versions, the particular hardware components, the method by which the hardware components are connected together and to the sensors, actuators, and terminals, and the assignment of logical paths connecting software components to hardware communication paths.

The SIntP shall be prepared in accordance with BTP 7-14 (Reference 1).

8.2 Identification of the Integration Process

8.2.1 Integration Level

Multiple levels of integration may be necessary, depending on the complexity of the software system that is being developed. Several integration steps may be required at some levels.

The SIntP shall provide the different levels of integration and the scope of each integration step at each level. And the SIntP shall provide a general description of the various objects that will be included in each step at each level.

8.2.2 Integration Objects and Strategies

The SIntP shall provide a complete list of all objects, computer hardware, instrumentation, software, and data that will be included in each integration step.

The SIntP shall describe the strategy that will be used for each integration step.

8.3 Integration Conditions

8.3.1 Integration and Testing Environment

The SIntP shall describe the environment that will be used to perform and test each integration step.

The SIntP shall provide the lists of the tools that will be used in each integration step.

8.3.2 Priorities

The SIntP shall provide the priority allocation for each integration step, based on schedule, dependence along the integration products, risk, and the allocation of any other factors deemed important.

8.3.3 Risks

The SIntP shall provide the risks that refer primarily to budget and schedule and other forms of risk that can be considered, at the option of the design team.

If any integration step involves significant risk, the SIntP shall describe the potential problems and the preventive measures that will be taken to avoid them.

8.3.4 Other Conditions

If an additional condition will be required, the SIntP shall describe it.

8.4 Organization of Integration

8.4.1 Integration Network Plan

The SIntP shall provide the integration steps listed by a time sequence. This will be determined primarily by the dependencies among the integration steps. And the SIntP shall provide the detailed level steps that will generally be required to complete successfully before a general level steps can be performed. Other factors can be considered.

8.4.2 Personnel and Responsibilities

- List of the personnel who will be involved in the integration steps
- Means to keep this list up to date

8.5 Integration Procedures

8.5.1 Required Products

The SIntP shall provide the lists of the products of each integration step.

8.5.2 Integration Instructions

A procedure shall be prepared which defines the integration process that is to be followed. The organization that is responsible for integration shall prepare the integration procedure. The procedure shall address any unique requirements imposed by the computer environment on the integration process. The procedure shall also address the recording and analysis of metrics related to integration errors (as may be required by the SQAP).

The integration may occur gradually, in a number of discrete steps throughout the implementation phase, or it may occur as a single activity. The responsible organization shall determine the most appropriate approach and timing for integration, and shall prepare the integration procedure accordingly.

The procedure shall be issued prior to commencing the integration activities and the plan shall consider the following items:

TS

8.5.3 Special Handling

If any special handling item will be required, the SIntP shall be revised for the special handling item.

9. SOFTWARE INSTALLATION PLAN

9.1 Introduction

Software installation is the process of installing the finished software products in the production environment. The SInstP will describe the general procedures for installing the software product. For any particular installation, modifications or additions may be required to account for local conditions.

The SInstP shall provide all of the installation requirements included as an actual plan. Installation testing shall be included in the SInstP and/or in the SVVP.

The SInstP shall consist of the transportation and installation of software from the development environment to the installation environment. The SInstP shall include the formal customer acceptance requirements.

The SInstP provides the guidance and the detail information to install software of the digital I&C system.

The SInstP shall be prepared in accordance with BTP 7-14 (Reference 1).

9.2 Identification of the Installation Environment

9.2.1 Application Environment

The SInstP shall describe the environment within which the software product is expected to perform.

9.2.2 Computer Hardware and Instrumentation

The installation environment description in the SInstP can include the reactor itself, the reactor protection system, and the protection system instrumentation and computer hardware. This description shall be limited to those items required for successful installation and operation.

9.3 Installation Package

9.3.1 Installation Software

The SInstP shall describe all of the materials that must be included in the installation package.

9.3.2 Installation Documents

The SInstP shall describe the software products, the media that contain them, and associated documents and, if alternatives are available, describe each of the software products, the media that contain them, and associated documents.

9.4 Installation Procedures

The SInstP shall describe completely the procedure for installing the software in the operational environment. This shall be a step-by-step procedure, written for the customer (contractor). Anticipated error conditions shall be described, with the appropriate recovery procedures. The following installation activities shall be categorically described in the specific SInstP;

TS

10. SOFTWARE TRAINING PLAN

10.1 Introduction

The STrngP shall describe the procedures that will be used to train the operators of the software. The operators, maintenance personnel, and management personnel will need to be trained in use of software of the I&C systems.

Because the actual training requirements depend to a great extent on the actual software product, development team, maintenance team and customer (utility), they are provided in the specific STrngP. This section describes only the contents of the generic STrngP.

The STrngP shall be prepared in accordance with BTP 7-14 (Reference 1).

10.2 Required Activity

The STrngP shall describe the following activities:

- Training program
- Training material (includes the operation manual)
- Training program validation and results report
- Training program implementation
- Training records

Training is essential for operators, technical support staff, and maintenance staff. Therefore, an operator training program shall be required so that the operators may learn the correct ways to use the system. The training manual is an important part of the training program. It will be provided by the system developer and/or the customer.

The operations manual provides all of the information necessary for the correct operation of the system. This includes normal operation, off-normal operation, and emergency operation. Start-up and shut-down of the system shall be discussed. All communications between the system and the operator shall be described, including the time sequencing of any extended conversations. All error messages shall be listed, together with their meaning and corrective action by the operator. The operations manual structure is dependent on the system characteristics.

10.3 Customer Training Need

The training need for the customer operation and maintenance personnel are described in this section. When an agreement is reached between the system designer and the utility, the content of this section shall be used as the guideline in preparing the customer training material.

- Training shall consider the following types of customer training needs:
  - Operators training needs
  - Maintenance personnel training needs
  - Management personnel training needs
- Training shall allow the operating personnel to witness installation activities and the site installation testing activities for practical familiarization with the software/system and its characteristics in the site environment.
- If practical, user classroom training shall be completed prior to installation. This will allow the operating personnel to support and operate the software/system immediately after it is installed.
- Training shall be in accordance with the training policy and procedures.

- Training materials shall be produced and delivered to the training location sufficiently in advance of the scheduled training activities to assure they will be available at the start of training.
- The V&V team shall review training materials to assure they properly reflect the safety aspects of the software/system.
- Instructors shall be certified for training in accordance with the applicable policy.
- Any training aid (such as videos or CDs) shall be prepared, controlled and distributed in accordance with the training policy and procedures.
- Training records shall be collected and retained for each operating personnel. Electronic copies of the training records shall be maintained as part of the software safety records.
- Consideration shall be given to dividing the training into two (2) sessions. Pre-installation class room training, and post-installation training in which the installed system will be utilized to provide additional hands-on practical training, using the actual delivered system.

11. SOFTWARE OPERATION AND MAINTENANCE PLAN

11.1 Introduction

The SOMP is to specify the requirements for the operation and maintenance computer software utilized for nuclear related functions.

The requirements described in this section shall be met by the utility.

The SOMP shall be prepared in accordance with BTP 7-14 (Reference 1).

11.2 Operation Phase

The SOMP shall contain the elements listed below. The SOMP will most likely contain little detail at the beginning of the project. As the system develops, details will be added until, at project completion, it is fully detailed.

TS

11.3 Maintenance Phase

The activities necessary to maintain the software, to remove uncovered errors, respond to new or revised requirements or to adapt the software to changes in the operating environments shall be coordinated.

Software maintenance work shall be coordinated in accordance with IEEE Std. 14764 (Reference 27).

Software modifications shall include the activities of the development cycle phases as appropriate, including V&V. These activities and their documentation shall be limited to those portions of the program that are actually modified except that the test phase shall include testing as necessary to verify that the modifications have not caused unintended adverse effects.

It shall be ensured that errors reported by users or the software developer are evaluated, documented and reported to other users, and to the software developer for correction if required.

The TER (Sheet 15) is prepared to document a problem with the system. If the problem is determined by the V&V team to be a defect in the software, the design team shall notify all users that are affected by the defect.

TS

TS

11.4 Retirement Phase

When a software product is no longer needed or has been superseded by a later revision, the users shall be formally notified of the termination of the computer code certificate (Sheet 14) and necessary steps to prevent access to the program shall be taken.

12. SOFTWARE SAFETY PLAN

12.1 Introduction

12.1.1 Purpose

The goal of the SSP is to enable development of SC class software for the digital I&C systems. The SSP provides reasonable assurance that any software defects do not present severe consequences to public health and safety.

12.1.2 Scope

The safety objective of the SSP is to provide procedures and methodologies for the development, procurement, maintenance, and retirement processes of SC class software that the engineers will follow to mitigate the potential of a software defect jeopardizing the health and safety of the public.

Any acceptable risks shall be defined in the relevant plan for a given system implementation.

The SSP shall be prepared in accordance with BTP 7-14 (Reference 1) and IEEE Std. 1228 (Reference 26).

The SSP applies to all SC class software whose failure could result in severe consequences to public health and safety.

12.2 Definitions, Acronyms and Abbreviations, and References

Refer to the acronyms and abbreviations, definitions, and Section 17 of this report.

12.3 Software Safety Management

12.3.1 Organization and Responsibilities

The organization of software activities is described in Section 2.4.

The software safety organization is composed of the following three teams:

- The design team is responsible to ensure that the requirements of the SSP are met throughout the software life cycle.

- The V&V team independently performs the software safety activities as the following:

  - To ensure there are sufficient, independent, technically qualified and trained resources to implement the requirements of the SSP.
  - To coordinate software safety task planning and implementation with the design team.
  - To ensure that records are kept in accordance with the SVVP and the QAM.
  - To participate with the QA team on any audits within the purview of its responsibilities.

- The QA team, which is independent and outside of the design and V&V team, generates QA procedures and directives that are followed by all involved members. The QA team provides oversight by way of periodic audits to verify that the design team and the V&V team is correctly

abiding by both the procedures and directives it generates, and those generated by the design team and the V&V team.

12.3.2 Resources

An early understanding of the resources required to develop SC class software shall be developed, so that these resources are put in place when they are required. The design team manager and the V&V team manager shall determine and assign the resources required to implement the SC class software. The following resources shall be considered:

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

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

A project schedule, which includes the software safety activities, shall be established and maintained to monitor progress throughout the project.

12.3.3 Staff Qualification and Training

One of the most important factors in developing reliable software is the development and use of a qualified staff. SC 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 is also a key factor in establishing and maintaining a qualified staff. In assessing the training requirements, the manager shall consider that:

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

The V&V team members shall be trained in the tools, techniques and methodologies described in Section 5.2.6.

All personnel participating in the following tasks for SC class software shall be qualified to perform their assigned tasks:

- Safety requirements definition
- Design and implementation of SC class software
- Software safety analyses
- Testing SC class software
- SSP implementation audit
- Process certification

The manager assesses the capabilities of candidates and appropriately selects qualified personnel based on their experience.

In determining whether any candidate is qualified, the design team manager considers whether the candidate:

- Understands the system and its potentially hazardous effects;
- 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 SQAP, SCMP, and SVVP;
- Is able to produce reliable software, good documentation, and can implement required QA 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 manager periodically reassesses the qualifications of all personnel working on SC class software, particularly when specific changes to the project become known. The manager may direct additional training before the changes are effective, in order to maintain a fully qualified project team.

12.3.4 Software Life Cycle

The phases of the software life cycle are described in Section 2.2. Specific software safety tasks and activities in the each life cycle phases are described in Section 12.4.

12.3.5 Documentation

The documentation for SC class software shall be prepared in accordance with the requirements in Section 4.3, and incorporates the software safety documentation requirements. The change and approval process for SC class software portions of the project documentation is the same as for other documentation as specified in Section 4.7.

12.3.5.1 Software Project Management

Section 3 identifies project organization and management activities that will be used to coordinate both the system development activities and the QA activities to ensure that both are done according to prescribed procedures and that adequate resources are allocated for their proper execution.

12.3.5.2 Software Configuration Management

Section 6 describes the configuration management of SC class software.

The design team manager will be responsible for developing its own SCMP to satisfy the unique requirements of software development process. It will address the following areas as a minimum:

- Description of software covered
- Change control procedures
- Documentation required
- Baseline identification and configuration

The SCMP for each design team covers the following phases of life cycle:

- Concept

- Requirements
- Design
- Implementation
- Test

The formal SCM activities during the installation and checkout life cycle phase is the responsibility of the utility.

12.3.5.3 Software Quality Assurance

Section 4 describes the requirements and methodology to be followed by the design team and the V&V team in developing, acquiring, using and maintaining software to be used for the design and operation of software systems.

12.3.5.4 Software Safety Requirements

The SRS includes the safety requirements to be met by the software to avoid or control system hazards.

12.3.5.5 Software Safety Design

The SDD includes descriptions of the software design elements that satisfy the software safety requirements.

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

The standards, practices, metrics and conventions to be applied to the I&C system software are defined in Section 4.4.

12.3.5.7 Test Documentation

Section 13 describes the test documentation.

12.3.5.8 Software Verification and Validation

Section 5 describes how the software safety will be verified and validated.

12.3.5.9 Reporting Safety Verification and Validation

Section 5.4 describes how the results of software safety-related V&V activities are documented.

12.3.5.10 Software User Documentation

Section 4.3.5 describes the software user documentation which may include information probably significant to the safe installation, use, maintenance, and/or retirement of the system.

12.3.5.11 Results of Software Safety Requirements Analysis

The results of the software safety requirements analysis as described in Section 12.4.2 shall be included in the SRS, STP, and hazard analysis.

12.3.5.12 Results of Software Safety Design Analysis

The results of the software safety design analysis as described in Section 12.4.3 shall be included in the following documents:

- Methods - Sections 6 and 4.7, and correction of this document
- Hazards - SRS and SDD
- Classification - design documentation such as SRS
- Architecture evaluation - V&V report (requirements traceability analysis)
- Requirements compliance - requirements traceability analysis
- Required tests - STP
- Modifications to coding and testing techniques - final V&V report

12.3.5.13 Results of Software Safety Code Analysis

The results of the software safety code analysis as defined in Section 12.4.4 shall be in the V&V report for the implementation phase of the software life cycle.

12.3.5.14 Results of Software Safety Test Analysis

The results of the software safety test analysis as defined in Section 12.4.5 shall be included in the V&V report for the test phase of software life cycle.

12.3.5.15 Results of Software Safety Change Analysis

The results of the software safety change analysis as defined in Section 12.4.6 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.

12.3.6 Software Safety Program Records

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 prelease or installed system is recorded in the final V&V report. Results of audits performed on software safety program tasks are documented in the V&V phase summary reports and in the final V&V report. Results of safety tests conducted on all or any part of the entire system are documented in the test reports. A record of training provided to software safety program personnel is generated and maintained by the responsible team in accordance with the QAM. Computer code certificate is included in the final V&V report.

Retention of software safety program records is in accordance with the QAM.

The tracking system used to ensure that hazards and their status are tracked throughout the software life cycle through retirement is the RTM and requirements traceability analysis as described in Section 5.2.6.3.

12.3.7 Software Configuration Management Activities

A key factor in developing reliable software is strict and detailed configuration management. SCM activities for SC class software are described in Section 6.

12.3.8 Software Quality Assurance Activities

Software QA activities for SC class software are described in Section 4.

12.3.9 Software Verification and Validation Activities

Software V&V activities for SC class software are described in Section 5.

12.3.10 Tool Support and Approval

The use of software tools for SC class software development follows the requirements specified in IEEE Std. 7-4.3.2 (Reference 14). Tools may produce better program structure and more reliable software through the automation of repetitive or time consuming tasks. The manager approves the use of any tool. This approval is based on an evaluation of the tool's readiness for use on a project involving SC class software. This evaluation considers:

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

12.3.11 Previously Developed or Purchased Software

Section 4.10.1 describes the requirements for using existing software, including previously developed or purchased software, as SC class software. In addition, EPRI TR-106439 (Reference 30) provides dedication requirements that must be met when a commercial off-the-shelf software package is to be used as SC class software. Procurement of software is done in accordance with EPRI TR-106439.

12.3.12 Subcontract Management

Section 4.10.2 specifies the provisions for ensuring that subcontractor software meets established software safety program requirements.

12.3.13 Process Certification

The computer code certificate is included in the final V&V report and is used to document that delivered software produced for the safety I&C system is approved for design use (see Section 4.3.7).

12.4 Software Safety Analyses

12.4.1 Software Safety Analysis Preparation

TS

TS

12.4.2 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 tests to be performed or the implementation of certain hazard recovery techniques. When there are more than one software system environment incorporated in the whole system under development, IEEE Std. 1228 requires that a different software system analysis may be performed.

Refer to Section 4.5.1 and Section 5.3.3 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.

12.4.3 Software Safety Design Analysis

Refer to Section 4.5.2 and Section 5.3.4 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.

12.4.4 Software Safety Code Analysis

Refer to Section 4.5.3 and Section 5.3.5 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.

12.4.5 Software Safety Test Analysis

Refer to Section 5.3.6 and Section 13 for a description of the software safety test analyses performed. These activities provide reasonable assurance that each system and software safety requirement is tested.

12.4.6 Software Safety Change Analysis

Refer to Sections 4.7 and 6 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.

12.5 Post Development

All post development activities are the responsibility of the utility.

In accordance with contractual requirements, the following post development activities may be provided to assist in training, deployment, monitoring and maintenance, and retirement of SC class software that are necessary to ensure the continued safety of the system after its deployment and until its orderly retirement

12.5.1 Training

The design team and the V&V team may provide or assist on the job training of utility personnel during software development and/or acceptance testing.

A plan will be developed by the utility to provide formal classroom training for plant operators and maintenance personnel.

12.5.2 Deployment

Manpower and other required resources shall be deployed by the utility for the installation of the software. Guidelines for the start-up and transition operations will be provided by the system and software development organization.

- Installation: Generation of software installation guidelines.
- Startup and transition: Generation of guidelines to start-up and operate the systems in their various modes.
- Operations support: Software documentation, software technical manuals and/or software user manuals shall be provided.

12.5.3 Monitoring

Section 11 contains requirements for monitoring the use of delivered software and associated problem reporting.

In addition, SC 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 are documented. Methods are considered that provide automatic verification of the system during operation.

12.5.4 Maintenance

Software changes are maintained in accordance with the SCMP.

12.5.5 Retirement and Notification

Section 11 describes the retirement of software and associated notification to current recipients.

12.6 Plan Approval

The SSP shall be reviewed by the design team and the V&V team. The SSP shall be approved by the manager of responsible team. It shall be reviewed against requirements specified in NUREG/CR-6430 and IEEE Std. 1228.

13. SOFTWARE TEST PLAN

13.1 Introduction

The purpose of the STP is to describe the scope, approach, resources, and schedule for the software 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.

It also complements the SVVP and provides additional details and minimum information requirements for the test activities.

The STP shall be developed in accordance with RG 1.170 (Reference 8) which endorses IEEE Std. 829 (Reference 18).

The overall objective of testing is to find faults introduced in the software.

Each testing process includes:

- Preparing test procedures to specify the manner and environment in which the code is tested detailing the steps to be followed by the tester, and specifying the set of inputs, execution conditions, and expected results.
- Verifying the test procedure with the objective to ensure that:
  - The required test coverage is provided, and
  - The procedures are specified so tests can be repeated.
- Verifying the software referenced in the test procedures is qualified for usage.
- Implementing the test procedure, recording the resultant outputs, and assessing the test results.
- Producing the test result report.
- Verifying the test result report with the objective to ensure that:
  - Tests were executed as specified in the test procedures,
  - Test results were recorded and properly analyzed, and
  - Test results reporting checklist (Sheet 8) were completed.

13.2 Overview

13.2.1 Organization

The organization for software test is described in Section 2.4.

13.2.2 Schedule

The STP shall describe the test activities within the project life cycle and milestones.

The STP shall summarize the overall schedule of the testing tasks, and describe the task iteration policy for the re-execution of test tasks and any dependencies.

The responsible team leader shall prepare and update test schedule.

13.2.3 Integrity Level Scheme

Refer to Section 5.2.3.

13.2.4 Resources Summary

The STP shall summarize the test resources, including staffing, facilities, tools, and special procedural requirements (e.g., security, access rights, and documentation control).

The testers shall not have been involved in the software design. The independence requirements described in Section 2.4 apply to the testers.

The development facilities are utilized for the testing. The design team leader and the V&V team leader shall coordinate the schedule for use of the development facilities.

13.2.5 Responsibilities

Each testing task is prepared and performed by the responsible entity. (Refer to Table 4-3.) The responsible entity establishes the test methods and procedures. The responsible entity selects the test input conditions and defines the test output acceptance criteria in the form of documentation and test practices. The responsible entity is responsible for defining and implementing practical tests.

The V&V team reviews all software testing activities including test documentation performed by the design team. It ensures adequate test coverage for the requirements, including operation under failure conditions.

13.2.6 Tools, Techniques, Methods, and Metrics

The STP shall describe the documents, hardware and software, test tools, techniques, methods, and test environment to be used in the test process.

The STP shall document the metrics to be used by the test effort, and describe how these metrics support the test objectives.

13.3 Test Processes

13.3.1 Component Testing

The objective of component testing is:

- To evaluate that executable code of each program behaves as specified in the SDD, and
- To ensure that the executable code of each program does not perform unintended functions.

13.3.2 Integration Testing

The objective of integration testing is:

- To evaluate that the software components integrated together meet the requirements specified in the SRS with target hardware and pre-developed software, and
- To find errors in handling stress conditions, timing, fail-safe features, error conditions, and error recovery.

Integration tests are designed to check all software interfaces and all interfaces between software and hardware.

13.3.3 System Testing

The objective of system testing (factory acceptance testing; FAT) is to verify that the entire executable code, including newly developed code, modified code, and any commercial or existing code, that have been integrated with the target hardware, successfully functions and fulfills all the system requirements related to the software.

A comprehensive system test is conducted for any safety I&C system. The system test evaluates the system performance in an environment that is real, or as close to real as can reasonably be created. Typically this environment is at the factory. A fully integrated system with the hardware and software, that is similar to the site environment, is required. The system test process demonstrates and proves correct and successful implementation of safety requirements and correct functioning of hardware-software combined system.

System testing includes the following tasks:

TS

System test procedures are prepared based upon the requirements of the design documents. The test specifications (if any) and procedures include test cases including the full range of data expected of the system.

13.3.4 Site Acceptance Testing

The objective of site acceptance testing is to confirm that the system was not damaged during shipment or installation.

The utility has the responsibility of site acceptance testing.

13.4 Test Documentation

All of formally issued test documentation shall be signed-off by all designated stakeholders.

All of formally issued test documentation shall be maintained as quality records and under the control of the SCM system.

13.4.1 Test Plan

The purpose of a test plan is to specify scope, approach, resources, and schedule of the testing activities for the software system. The test plan shall identify the items being tested, the features to be tested, the testing tasks to be performed, the personnel responsible for each task, and the associated risks.

The test plan shall describe the scope of the system/software to be tested.

A test plan shall contain the following information:

- Test items: The object of testing, e.g., specific attributes of the software, including their version/revision level.
- Features to be tested, features not to be tested, approach: All software product or software-based system features and combinations of software of system features to be test, all features and known significant combinations of features that will not be tested and the rationale for exclusion, and the overall approach for each level of testing. These three types of information are commonly combined in a table. This table contains a unique identifier for each requirement for the test (e.g., system and/or software requirements, design, or code), an indication of the source of the requirement (e.g., a paragraph number in the source document), and a summary of the requirement and an identification of one or more generic methods of test. Some examples of possible methods are as follows:
  - Black box: test inputs can be generated and the outputs captured and completely evaluated from the outside of a test item; i.e., test cases are developed from the test item specification, only without looking at the code or design.
  - White box: Considers the internal structure of the software (e.g., attempts to reach all of the code). Commonly requires some kind of test support software.
  - Analysis: Just viewing the outputs cannot confirm that the test executed successfully; some kind of additional computations, simulations, studies, and so on will be required.
  - Inspection: This is a static test; the code or documentation is read and examined without being executed.
- Item pass/fail criteria: The criteria to be used to determine whether each test item has passed or failed testing.
- Suspension criteria and resumption requirements: The criteria used to suspend all or a portion of the testing activity on the test items associated with this plan, with the testing activities that must be repeated when testing is resumed.
- Test deliverables: All documents to be delivered by the test activity such as test procedures and test reports.
- Test management: Planned activities and tasks, test progression, environment/infrastructure, responsibilities and authority, interfaces among the parties involved, resources and their allocation, training, schedules, estimates, costs, risk, and contingency
- Test coverage requirements: The requirements for indications of the degree to which the test item has been reached or covered by the test cases. The test coverage is often expressed in terms of percentage of requirements or code tested.

13.4.2 Test Cases

The purpose of a test case is to define the information needed as it pertains to inputs to and outputs from the software or software-based system being tested.

A test case is a set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement. Test case documentation specifies inputs, expected results, and a set of execution conditions for a test item.

Test cases for a test item may be included in a separate document, or incorporated into test procedure for the test item.

A test case shall contain the following information:

- Test case identifier: The unique identifier needed by each test case so that it can be distinguished from all other test cases. An automated tool may control the generation of the identifiers.
- Objective: Brief description of the special focus or objective for the test case or a series of test cases. This is much more detailed than the test items in the STP.
- Inputs: Specific values (with tolerances where appropriate) or names (e.g., constant tables, transaction files).
- Outcomes: All outputs and the expected behavior (e.g., response time) required of the test items. The exact value(s) (with tolerances where appropriate) for each required output and expected behavior shall be provided.
- Environmental needs: Description for the test environment needed for test setup, execution, and results recording. These are commonly documented per scenario or group of scenarios. This information includes the characteristics and configurations of the hardware required to execute the test case, all software configurations required to execute the test case, and any other requirements (e.g., unique facility needs, specially trained personnel, and third-party provided environments) if there are any. This information is not documented in the test case documents if the STP or the test procedure contains the same information.
- Special procedural requirements: Any special constraints on the test procedures that execute the test case such as pre- and post-conditions and/or processing. This may reference the use of automated test tools. This provides exceptions and/or additions to the test procedure, not a repeat of any of the information contained in the procedure.
- Interface dependencies: The identifiers of test cases that must be executed prior to the test case. This includes the summary of the nature of the dependencies. This may not be needed if test cases are documented in the order in which they need to be executed.

13.4.3 Test Procedures

The purpose of a test procedure is to specify the steps for executing a set of test cases or the steps used to exercise a software product or software-based system item in order to evaluate a set of features. Also, the purpose of the test procedure is to specify any refinements of the test approach which is described in the STP and to identify the features to be tested by the test.

A test procedure provides detailed instructions for the setup, execution, and evaluation of results for a given test case, or specifies a sequence of actions for the execution of a test. Also, the test procedure specifies the details of the test approach for a software feature or combination of software features and identifies the associated tests.

A test procedure for a test item may include test cases for the test item.

A test procedure shall contain the following information:

- Features to be tested: This information is more detailed than that of the STP.
- Approach refinements: The refinements to the approach described in the corresponding STP. This information includes specific test techniques to be used. The method of analyzing test results shall be identified (e.g., comparator tools, visual inspection, etc.).
- Test identification: The list of the identifiers and brief description of test cases.
- Feature pass/fail criteria: The criteria to be used to determine whether the feature or feature combination has passed or failed. This information is not needed if it is covered by the STP and there have been no subsequent changes to the criteria.
- Inputs, outputs, and special requirements: All that is needed to execute the tests, including test cases, databases, automated tools, and external and/or third-party systems. Any special requirements that are necessary for the execution of the procedure such as prerequisite procedures, special skill requirements, or special environmental requirements.

- Ordered description of the steps to be taken to execute the test cases.

13.4.4 Test Reports

The purpose of a test report is to summarize the results of the designated testing activities and to provide evaluations and recommendations based on these results.

A test report (or a regression test report) is issued for each testing performance. Typically, a regression test report does not cover all test cases or procedures in the corresponding test documentation.

A test report shall contain the following information:

- Overview of test results: The summary of the evaluation of the test items. This information includes the items tested with their version/revision level, the environment in which the testing activities took place, and its impact (if any).
- Detailed test results: The summary of the results of testing. This information includes the following:
  - All resolved anomalies and the summary of their resolutions (or available references)
  - All unresolved anomalies
  - The summary of major testing activities and events
  - The summary of relevant metrics collected
  - Any variances of the test items from their specifications, any variances from the test documentation (e.g., test changes or tests not executed), and the reason for each variance (or recorded references)
  - The evaluation of the comprehensiveness of the testing process (e.g., coverage metrics, if specified) against the comprehensiveness criteria specified in the STP of the test procedure
- Rationale for decisions: The issues that were considered for any decisions and the reasons for the selection of the conclusions
- Conclusions and recommendations: Overall evaluations of test items including their limitation, based on the test results and the item level pass/fail criteria, and the recommendations the status relative to availability for production use and under what circumstances (e.g., immediately, with a specified subset of anomalies resolved, or never).
- TERs: The TER logs and copies of the TERs shall be attached. Any outstanding TERs shall be brought to the attention of the management. Refer to Section 4.7.1 for description of TERs.

13.5 Test Administration

13.5.1 Anomaly Resolution and Reporting

The STP shall describe the method of reporting and resolving anomalies. Refer to Section 4.7.

13.5.2 Task Iteration Policy

When the test item or the test documentation is changed (e.g., re-performing tests after the anomaly resolutions), the test cases and/or procedures must be identified for the effects of the changes, and re-performed. The test cases and/or procedures must be identified in the test reports identifying the rationale and the results of the testing. This testing is referred as regression testing.

13.5.3 Deviation Policy

If any deviation from the reviewed and approved test documentation is required, the change must be identified, rationale for the change provided, and a determination of effect on software quality provided. Any deviation must be approved by the responsible manager.

13.5.4 Control Procedures

All test documents described in the STP or other plans shall be prepared and retained as QA records.

13.5.5 Standards, Practices, and Conventions

The STP shall describe the standards, practices, and conventions that govern the performance of testing tasks including internal organizational standards, practices, and policies.

13.6 Test Reporting

The STP shall specify the purpose, content, format, recipients, and timing of test reports.

Refer to Section 13.4.4.

14. SECURE DEVELOPMENT AND OPERATIONAL ENVIRONMENT

14.1 Introduction

This section describes the establishment of the SDOE for the digital safety I&C systems during the software life cycle phases in accordance with RG 1.152 (Reference 4). The phases include concept, requirements, design, implementation, and test, as defined in Section 2.2. The digital safety I&C systems will be developed in accordance with this section.

14.2 System Design Features for Security

The safety I&C system is designed with features that minimize the potential for security vulnerabilities. The system uses a combination of personnel access controls, and hardware and software design features to protect the safety-related software from unauthorized alteration and to ensure the deterministic performance of the safety I&C system. This is achieved through a combination of features within the common digital platform and features within the system applications. These features prevent the system from adverse operating caused by either intentional or unintentional sources. RG 1.152 (Reference 4) security guidance is implemented during the software development process.

This section describes the design features that prevent unintended changes to hardware or software code, and to detect those unintended changes if they occur. These design features are based on the assumption that potential intruders have a detailed knowledge of the safety I&C system, and that unintended functions can be included during initial development or through unintended modifications.

14.2.1 Physical Access Controls

The hardware or software of the safety I&C system can only be altered by physically accessing the safety I&C system. Security controls that prevent changes through electronic access are described in Sections 14.2.2 and 14.2.3.

TS

14.2.2 Electronic Access Controls

TS

14.2.3 Software Alteration Controls

TS

TS

14.2.4 Deterministic Performance Controls

Deterministic behavior is a key attribute of the safety I&C system that ensures predictable performance for all abnormal plant conditions. This section describes the key features of the safety I&C system that achieve deterministic performance, the security design features that protect that performance against external threats, and the security features that ensure any disruption to deterministic performance is detected and alarmed.

TS

TS

14.3 Software Development Processes for Security

This section describes the processes implemented during each software life cycle phase for the safety I&C system, which prevent inclusion of unintended functions in documents and software code. Unintended functions can be included during initial development or through unintended modifications. This section assesses the effectiveness of those processes.

These processes are based on the assumption that potential intruders have a detailed knowledge of the safety I&C system, including development tools and development processes. All types of unwanted designs and software code are considered in the assessment of the effectiveness of these processes.

14.3.1 Concept Phase

TS

TS

14.3.2 Requirements Phase

The SRS fully documents the security design requirements established during the concept phase. Requirements applicable to the common digital platform or to the safety I&C system applications are clearly distinguished. These documents are generated by the design team based on peer review, such as design review meetings. The system requirements documents undergo formal independent technical review and formal technical management review.

TS

TS

14.3.3 Design Phase

During this phase the SDD is created. The SDD provides the necessary software design detail to implement all software functions, including all security features described in Section 14.2. The SDD reflects the requirements documented in the SRS, which is provided from the requirements phase.

TS

TS

14.3.4 Implementation Phase

Software code is created during the implementation phase for the common digital platform and for the software that is implemented on each target processor for each safety I&C system. The software is documented in source code listings and in graphical diagrams that show the interconnection of various common platform software function blocks.

The software code reflects the detailed designs documented in the SDD, which is provided from the design phase. Software is configured in units that represent logical groupings of platform or application functions.

TS

TS

14.3.5 Test Phase

During this phase, common digital platform operating system and application software is integrated with the common digital platform hardware for each system. The test phase covers two types of testing:

- Integration testing: This testing confirms that the software components, which are integrated together with the actual common platform hardware, meet the requirements specified in the SRS. The testing looks for errors in the software, errors in the hardware and software interfaces, errors in timing and fail-safe features, errors in handling self-test detected failure conditions, and errors in failure recovery. Integration tests are conducted on multiple units that form a logical test entity or subsystem.
- System testing (FAT): This is a test of the fully integrated system to validate that the system meets the system requirements related to the software. The system test is conducted with all connected system interfaces, including those that pose security threats. The system testing demonstrates that all system requirements related to the software function correctly in the final integrated system. The FAT is the final test performed by the equipment supplier, prior to delivery of the equipment.

TS

TS

14.3.6 Security Measures for Document and Software Storage

TS

14.4 Security Processes for Commercial-off-the-Shelf Software

There are two kinds of COTS software:

- System software: This software runs within the operating system of the safety I&C system processors.
- Engineering tools: This software is used to create original software in accordance with this report, which runs within the safety I&C system processors.

This section describes the processes employed to (1) achieve high assurance that COTS software products are free of unwanted code that could degrade the security of the safety I&C system, and (2) to ensure that any unwanted code that may be introduced into the safety I&C system by the COTS software products is detected and eliminated.

TS

14.5 Vulnerability Assessment

Sections 14.2, 14.3 and 14.4 describe the security features and processes planned during the concept phase. As described above, the adequacy of these design features and processes are reassessed during each subsequent life cycle phase to ensure they remain adequate to mitigate any new threats that may be identified.

The safety I&C system exhibits susceptibility to unwanted changes and introduction of unwanted functions throughout the software development life cycle. Without SDOE, the safety I&C system software may be adversely affected by personnel engaged in the development process. Potential threats arise from:

TS

TS

The assessment shall address each of these threats, and mitigation activities appropriate for the each life cycle phase. The assessment assumes that potential undesirable changes originate from personnel with significant knowledge of the safety I&C system and the life cycle processes. There is no type of undesired change that is excluded from the assessment, whether intentional or unintentional.

Recurring vulnerability assessments confirm the adequacy of the safety I&C system design features and development processes in mitigating the potential threats and any new threats that may be identified.

15. SHEETS

This section provides typical forms referred in this report for software tasks.

Sheet 1. Comment Record (Typical)

Doc. No.		Rev. No.		Date of Issue	
Doc. Title					

No.	Sec./Page	Comment	Comment Resolution
		Reviewed by: Name Signature Date Approved by: Name Signature Date	Resolved by: Name Signature Date Approved by: Name Signature Date

Sheet 2. Risk Analysis Worksheet (Typical)

System: _____ Subsystem: _____

- Software class:

  Safety-critical ( ) Important-to-safety ( )

- Hazard severity category (based on Figure 3 of NUREG/CR-6430):

  Catastrophic ( ) Critical ( )

  Marginal ( ) Negligible ( )

- Hazard probability level (based on Figure 4 of NUREG/CR-6430):

  Frequent ( ) Probable ( )

  Occasional ( ) Remote ( )

  Improbable ( )

- Risk level (based on Figure 5 of NUREG/CR-6430):

  High ( ) Medium ( ) Low ( )

- Recommendations for eliminating, reducing, or mitigation the risks:

Prepared by:
Name: _____ Signature: _____ Date: _____

Sheet 3. Software V&V Checklist – Concept Phase (Typical)

System: _____

Documents/items reviewed:

Check results:

No.	Check Point	Yes/No
1	Are the concept documents complete?	
2	Are the concepts correct?	
3	Are the concept documents consistent and complete?	
4	Are the concepts clear and unambiguous?	
5	Are the concepts feasible?	
6	Are the concepts testable?	
7	Are the concept documents properly entered into the requirements traceability tool?	
8	Are the concept document requirements marked per traceability requirements?	

Comments (optional):

Prepared by:
Name: _____ Signature: _____ Date: _____

Sheet 4. Software V&V Checklist – Requirements Phase (Typical)

System: _____

Documents/items reviewed:

Check results:

No.	Check Point	Yes/No
1	Are the requirements documents complete?	
2	Are the requirements correct?	
3	Are the requirements documents consistent and complete?	
4	Are the requirements clear and unambiguous?	
5	Are the requirements feasible?	
6	Are the requirements testable and related to performance goals?	
7	Are the requirements documents properly entered into the requirements traceability tool?	
8	Are the requirements documents marked per traceability requirements?	
9	Are the requirements for interfacing with other equipment consistent and complete?	
10	Are process or digital data input requirements identified properly?	
11	Are initialization requirements identified?	
12	Are in-service tests or diagnostic capabilities defined?	
13	Do the requirements specify handling of abnormal events?	
14	Has a requirement analysis been performed to address potential hazards?	

Comments (optional):

Prepared by:
Name: _____ Signature: _____ Date: _____

Sheet 5. Software V&V Checklist – Design Phase (Typical)

System: _____

Documents/items reviewed:

Check results:

No.	Check Point	Yes/No
1	Are the design documents complete?	
2	Are the design documents properly entered into the requirements traceability tool?	
3	Are the design documents marked per traceability requirements?	
4	Is the design traceable to the requirements?	
5	Is the design complete?	
6	Is the design correct?	
7	Is the design internally consistent?	
8	Is the design clear and unambiguous?	
9	Is the design feasible?	
10	Is the design testable?	
11	Is software architecture adequately addressed?	
12	Are input/output interfaces adequately addressed?	
13	Is the testability of the system adequately addressed (e.g., response time)?	
14	Is algorithm design adequately addressed (e.g., base functions addressed)?	
15	Is information flow adequately addressed (communication between subsystems, data management and signal conversion to engineering units)?	
16	Are operating modes/sequences adequately addressed (e.g., initialization, startup, test system mode of operation)?	
17	Does the design adequately address potential hazards?	

Comments (optional):

Prepared by:
Name: _____ Signature: _____ Date: _____

Sheet 6. Software V&V Checklist – Implementation Phase (Typical)

System: _____

Documents/items reviewed:

Check results:

No.	Check Point	Yes/No
1	Does the source code conform to specified standards and procedures?	
2	Is the source code traceable to the functional design requirements?	
3	Are the comment statements provided sufficiently to give ad adequate description of each routine and data structure?	
4	Is the source code understandable?	
5	Is the source code consistent with the design?	
6	Are all the variables properly specified and used?	
7	Is there satisfactory error checking?	
8	Have potential software hazards been adequately addressed, including implementation of prevention and/or control techniques (e.g., logic, data, interfaces, constraint, non-critical code, timing/sizing)?	
9	Do all subroutine calls transfer data variables correctly?	
10	Is the data correctly passed between components, and/or integrated subsystems or systems?	
11	Do the database components adequately and correctly reflect the program and general content?	
12	Do component test reports indicate correct execution of critical software elements?	
13	Do procedures exist (as necessary) to: a. Configure the application programming tool(s)? b. Generate and upload the application code? c. Maintain application source code configuration control?	

Comments (optional):

Prepared by:
Name: _____ Signature: _____ Date: _____

Sheet 7. Software V&V Checklist – Test Phase (Typical)

System: _____

Documents/items reviewed:

Check results:

No.	Check Point	Yes/No
1	Is the test plan description complete?	
2	Are the test case definitions adequate and complete?	
3	Is each testable requirement adequately covered?	
4	Is the plan for evaluating and reporting test results adequate?	
5	Does the test plan specify the required test environment (hardware or software)?	
6	Have all the elements of an integrated program been identified?	
7	Does the plan require a test case log?	
8	For program maintenance changes, have adequate regression tests been specified to verify that the modifications have not caused adverse effects on unmodified code?	

Comments (optional):

Prepared by:
Name: _____ Signature: _____ Date: _____

Sheet 8. Software V&V Checklist – Test Results Reporting (Typical)

System: _____

Documents/items reviewed:

Check results:

No.	Check Point	Yes/No
1	Do the test results comply with the format specified in the test plan?	
2	Do the test results provide and accurate statement of the testing performed?	
3	Has each section of the test procedure been completed accurately?	
4	Have all test cases been executed correctly?	
5	Have test results been evaluated as acceptable?	
6	Does the test report provide a summary of test results and recommendations?	
7	Have any deviations from the test plan or expected test result (criteria) been properly documented in test exception reports?	
8	For program maintenance changes, have adequate regression tests been specified to verify that the modifications have not caused adverse effects on unmodified code?	

Comments (optional):

Prepared by:
Name: _____ Signature: _____ Date: _____

Sheet 9. Software V&V Checklist – Final (Typical)

System: _____

Documents/items reviewed:

Check results:

No.	Check Point	Yes/No
1	Is the information in the user documentation consistent with that in the requirements?	
2	Is the information in the user documentation consistent with that in the design?	
3	Is the information in the user documentation an accurate reflection of the coded program?	
4	Is the information in the user documentation internally consistent?	
5	Is the information in the user documentation clear and unambiguous?	
6	Is the user documentation installation package sufficient to install the software on the delivered hardware?	
7	Have all discrepancies and V&V findings been resolved to the satisfaction of the V&V team?	
8	Is the corrective action process in place for the user to report errors?	
9	If required, have installation/integration tests been successfully performed and documented?	
10	Have access controls on the program source been implemented?	
11	Is the software identified by a unique code name and version number which appears on its hardcopy output?	
12	Are all applicable elements of the V&V report present?	
13	a. Cover sheet b. Design description, or reference c. Requirements document, or reference d. Test plan, or reference e. Test results evaluation f. Test results hardcopy computer output where available/appropriate g. User documentation, or reference h. Source code listing i. Computer code certificate j. V&V checklist	

Comments (optional):

Prepared by:
Name: _____ Signature: _____ Date: _____

Sheet 10. Requirements Traceability Matrix (Typical)

RTM No.	System Requirements	Software Requirements	Design	Implementation	Test

Sheet 11. Software Change Request (Typical)

SCR NUMBER: ISSUE DATE: Page 1 of 1

Plant	
System Name	
Software Component	
Version	
REFERENCES	
DESCRIPTION	
REASON FOR CHANGE	
DOCUMENT AFFECTED	
DOCUMENT ATTACHMENT	

Prepared by:
Name: _____ Signature: _____ Date: _____
XXX system engineering group

Reviewed by:
Name: _____ Signature: _____ Date: _____
XXX system engineering group

Approved by:
Name: _____ Signature: _____ Date: _____
XXX system engineering group

Sheet 12. Computer Program Error Notification (Typical)

Program Name: _____ Version(s): _____

Error Number for This Program: _____

Program Executable File Identification

No.	Computer System	Permanent File Name	Owner ID	Cycle
1				
2				
3				

Error Description:

Possible Impact of Error:

Recommendation to User (Error Avoidance and/or Remedial Action):

Prepared by:
Name: _____ Signature: _____ Date: _____
XXX system engineering group

Reviewed by:
Name: _____ Signature: _____ Date: _____
XXX system engineering group

Approved by:
Name: _____ Signature: _____ Date: _____
XXX system engineering group

Sheet 13. Computer Program Error Notification Response Receipt (Typical)

Program Name: _____ Version(s): _____

Error Number for This Program: _____

Receiving EGS: _____ Date: _____
XXX Department

I acknowledge receipt of the attached error notice, and have assessed the impact on design analyses as indicated. When the error impact is yes, appropriate actions will be taken, documented and made a quality record.				
User Name	Signature	Date	Error Impact	
			No	Yes

Receiving EGS is to return this signed form within sixty (60) days to:

EGS: _____ Date: _____
XXX Department

Sheet 14. Computer Code Certificate (Typical)

The following code, as noted by its name, version number and executable file identification, is approved for design use.

Code Name: _____

Version Number: _____

Executable File Identification: _____

Computer(s): _____

List any limitations on use, special hardware considerations, etc.:

Software V&V Report Number: _____

EGS: _____ Date: _____
XXX Department

Sheet 15. Test Exception Report (Typical)

TER NUMBER:

System Name		Plant	
Procedure Name			
Procedure Number		Rev.	
Tested By		Date	
Reviewed By		Date	
Approved By		Date	
Summary of Exception			
S/W Class			
Resolution			
Implementation			
Software Change Number			
Implemented By		Date	
Reviewed By		Date	
Approved By		Date	
Re-test Result			
Re-tested By		Date	
Reviewed By		Date	
Approved By		Date	

16. REFERENCES

1. NUREG-0800, BTP 7-14, Revision 5, “Guidance on Software Reviews for Digital Computer-Based Instrumentation and Control System,” March 2007.
2. 10 CFR 50, Appendix B, “Quality Assurance Criteria for Nuclear Power Plants and Fuel Reprocessing Plants.”
3. Regulatory Guide 1.97, Revision 4, “Criteria for Accident Monitoring Instrumentation for Nuclear Power Plants,” June 2006.
4. Regulatory Guide 1.152, Revision 3, “Criteria for Use of Computers in Safety Systems of Nuclear Power Plants,” July 2011.
5. Regulatory Guide 1.153, Revision 1, “Criteria for Safety Systems,” June 1996.
6. Regulatory Guide 1.168, Revision 2, “Verification, Validation, Reviews, and Audits for Digital Computer Software Used in Safety Systems of Nuclear Power Plants,” July 2013.
7. Regulatory Guide 1.169, Revision 1, “Configuration Management Plans for Digital Computer Software Used in Safety Systems of Nuclear Power Plants,” July 2013.
8. Regulatory Guide 1.170, Revision 1, “Test Documentation for Digital Computer Software Used in Safety Systems of Nuclear Power Plants,” July 2013.
9. Regulatory Guide 1.171, Revision 1, “Software Unit Testing for Digital Computer Software Used in Safety Systems of Nuclear Power Plants,” July 2013.
10. Regulatory Guide 1.172, Revision 1, “Software Requirements Specifications for Digital Computer Software and Complex Electronics Used in Safety Systems of Nuclear Power Plants,” July 2013.
11. Regulatory Guide 1.173, Revision 1, “Developing Software Life-Cycle Processes for Digital Computer Software Used in Safety Systems of Nuclear Power Plants,” July 2013.
12. NUREG/CR-6101, “Software Reliability and Safety in Nuclear Reactor Protection Systems,” June 1993.
13. NUREG/CR-6430, “Software Safety Hazard Analysis,” February 1996.
14. IEEE Std. 7-4.3.2-2003, “IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations.”
15. IEEE Std. 603-1991, “IEEE Standard Criteria for Safety Systems for Nuclear Power Generating Stations.”
16. IEEE Std. 730-2002, “IEEE Standard for Software Quality Assurance Plans.”
17. IEEE Std. 828-2005, “IEEE Standard for Software Configuration Management Plans.”
18. IEEE Std. 829-2008, “IEEE Standard for Software and System Test Documentation.”
19. IEEE Std. 830-1998 (Reaffirmed 2009), “IEEE Recommended Practice for Software Requirements Specifications.”
20. ANSI/IEEE Std. 1008-1987 (Reaffirmed 2009), “IEEE Standard for Software Unit Testing.”
21. IEEE Std. 1012-2004, “IEEE Standard for Software Verification and Validation.”
22. IEEE Std. 1016-2009, “IEEE Standard for Information Technology – Systems Design – Software Design Descriptions.”
23. IEEE Std. 1028-2008, “IEEE Standard for Software Reviews and Audits.”
24. IEEE Std. 1058-1998, “IEEE Standard for Software Project Management Plans.”
25. IEEE Std. 1074-2006, “IEEE Standard for Developing Software Life Cycle Processes.”
26. IEEE Std. 1228-1994 (Reaffirmed 2002), “IEEE Standard for Software Safety Plans.”
27. IEEE Std. 14764-2006, “Software Engineering – Software Life Cycle Processes – Maintenance.”
28. IEEE Std. 26514-2010, “IEEE Standard for Adoption of ISO/IEC 26514:2008 Systems and Software Engineering – Requirements for Designers and Developers of User Documentation.”
29. ISO/IEC/IEEE 24765:2010, “Systems and Software Engineering – Vocabulary.”
30. EPRI TR-106439, “Guidance on Evaluation and Acceptance of Commercial Grade Digital Equipment for Nuclear Safety Stations,” October 1996.
31. NIST Special Publication 800-82, Revision 1, “Guide to Industrial Control Systems (ICS) Security,” May 2013.
32. APR1400 DC Quality Assurance Manual (QAM).

17. DEFINITIONS

The following definitions are used in this report. The definitions of all other terms not described here can be found in Reference 29.

1.	Accident	An unplanned event or series of events that result in death, injury, illness, environmental damage to or loss of equipment or property
2.	Application software	Software designed to fulfill specific needs of user
3.	Baseline	Software and related documents which have been formally reviewed and that can be changed only through formal change control procedures
4.	Component	One of the parts that make up a system
5.	Component testing	Testing of individual components
6.	Configuration control	The process of identifying and defining the configuration items in a system, and controlling the release and change of these items
7.	Configuration item	A collection of hardware or software elements treated as a component for the purpose of configuration control
8.	Control point	A project agreed on point in time or times when specified agreements or controls are applied to the software configuration items being developed, e.g., an approved baseline or release of a specified document/code
9.	Factory acceptance testing	Final testing performed to demonstrate that deliverable system satisfies acceptance criteria as a condition of its transfer to a receiving authority
10.	Hazard	A condition that is a prerequisite to an accident
11.	Integration testing	Testing in which software components, hardware components, or both are combined and tested to evaluate the interaction between them
12.	Milestone	A scheduled event used to measure progress. Examples of major milestones for software development process may include an acquirer or managerial sign-off, baselining of a specification, completion of system integration, and product delivery. Minor milestones might include baselining of a software component or completion of a chapter of the user's manual.
13.	Module testing	Testing of a single subroutine, or a main routine with stubs
14.	Product	Any output of the software development activities; e.g., document, code, model
15.	Project agreement	A document or set of documents baselined by the acquirer and the supplier that specifies the conditions under which the project will be conducted. A project agreement may include items such as the scope, objectives,

	assumptions, management interfaces, risks, staffing plan, resource requirements, price, schedule, resource and budget allocations, approved products, and acceptance criteria for the approved products. Documents in a project agreement may include some or all of the following: a contract, a statement of work, user requirements, system engineering specifications, software requirements specifications, a software development process management plan, supporting process plans, a business plan, a project charter, or a memo of understanding.
16. Project deliverable	A work product to be delivered to the customer. Quantities, delivery dates, and delivery locations are specified in contract. Approved products may include the following: operational requirements, functional specifications, design documentation, source code, object code, test results, installation instructions, training aids, user's manuals, product development tools, and maintenance documentation. Approved products may be self-contained or may be part of a larger system's deliverables.
17. Regression testing	Testing replication performed as a result of changes to the previously tested design specified regression analysis
18. Risk	A measure that combines both the likelihood that a hazard will cause an accident and the severity of that accident
19. Software life cycle	The period of time that starts after a software product is conceived and ends when the software product is no longer available for use. The software life cycle typically includes a concept phase, requirements phase, a design phase, an implementation phase, a test phase, an installation and checkout phase, and an operation and maintenance phase.
20. Software module (or module)	The smallest software entity that is subjected to testing
21. Software project	The set of work activities, both technical and managerial, required for satisfying the terms and conditions of a project agreement. A software project has specific starting and ending dates, well-defined objectives and constraints, established responsibilities, and a budget and schedule. A software project will span all the software life cycle to develop the software of a specific digital I&C system.
22. Software safety program	A systematic approach to reduce software risks
23. Software unit (or unit)	A separately testable software element that is the integration of several modules
24. Software validation	The test and evaluation of the completed software/configuration item to ensure compliance with software requirements
25. Software verification	The process of determining whether the product of a given phase of the software development cycle fulfills the requirements imposed by the previous phase

26. Supplier	An organization that develops some or all of the approved products for an acquirer. Suppliers may include organizations that have primary responsibility for project deliverables and subcontractors that deliver some part of the project deliverables to a primary supplier. In the latter case, the primary supplier is also an acquirer.
27. System testing	Testing of a system during or at the end of the development processes to determine whether it satisfies specified requirements
28. Unit testing	Testing of individual hardware or software units or groups of related units to witness a function
29. Utility	Combined license applicant
30. Vulnerability	Weakness in a system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source

APPENDIX A. THE I&C SYSTEM SOFTWARE CLASSES

A.1 Assignment of Software for the I&C Systems to Classes

The following table describes the assignment of software for the I&C systems to specific classes defined in Section 2.1.

Table A-1 Assignment of Software for the I&C Systems to Classes

TS

TS

A.2 Software Classification for the Safety I&C Systems

The software for the safety I&C systems is classified as one of the two software classes: SC class and ITS class. Both classes of the software for the safety I&C systems are implemented on Class 1E hardware.

IEEE Std. 603 (Reference 15), Clause 5.3 requires that components and modules be of a quality that is consistent with minimum maintenance requirements and low failure rates. RG 1.152 (Reference 4) endorses IEEE Std. 7-4.3.2 (Reference 14) which supplements IEEE Std. 603 for computer-based safety I&C systems. IEEE Std. 7-4.3.2, Clause 5.3.3, “Verification and validation” states that software V&V effort shall be performed in accordance with IEEE Std. 1012 (Reference 21) and the IEEE Std. 1012 V&V requirements for the highest integrity level (level 4) apply to systems developed using IEEE Std. 7-4.3.2. Performing adequate V&V is a critical part of ensuring a high quality development process for safety I&C systems.

In the safety I&C system design, for ITS class software, V&V activities are not performed in accordance with the V&V requirements for the highest integrity level as stated in IEEE Std. 7-4.3.2. However, the assurance is given to demonstrate that ITS class software is of sufficient quality and reliability to maintain the safety function of the software and does not present a hazard on SC class software.

TS

TS

APPENDIX B. SOFTWARE PROJECT ORGANIZATION FOR THE SAFETY I&C SYSTEMS

TS

TS

APPENDIX C. CONFORMANCE TO IEEE STD. 1012-2004

TS

APPENDIX D. FPGA DEVELOPMENT PROCESS

TS

TS

TS