DCPP Procedure CF2.ID2, Revision 10, "Software Configuration Management for Plant Operations and Operations Support "
(LAR Reference 50)

TITLE: Software Configuration Management for Plant Operations and Operations Support

INFO ONLY
EFFECTIVE DATE

PROCEDURE CLASSIFICATION: QUALITY RELATED

TABLE OF CONTENTS

1. SCOPE

1.1 The purpose of this procedure is to provide uniform, minimum acceptable requirements for preparing Software Configuration Management (SCM) and Software Quality Assurance (SQA) plans and maintaining configuration control of computer systems and applications that are used for the monitoring or operation of plant structures, systems, and components.

1.2 This would include any software providing automatic control, or software that provides indication of plant conditions to operate or make operational decisions about the plant. Also controlled by this procedure are systems providing collection, storage, and/or retrieval of plant parameters used to meet regulatory commitments or provide tuning of plant control or protection parameters.

1.3 All business related software comes under the control of CF2.ID3. CF2.ID9 provides additional guidance for developing new software applications and producing required documentation, including SCM and SQA plans.

1.4 This procedure applies to computer systems whose software design is under complete plant control, and those that are proprietary and maintained by a vendor. Requirements are provided for maintenance of computer software as applied to the operation and maintenance of power plant equipment. This procedure only applies to applications within the realm of the plant or those applications that use recorded or live plant data for operational or engineering decision making to the extent that the integrity of that data is not compromised.

1.5 This procedure provides guidance on the requirements for status control for in service computer systems and applications to assure the applications and computer systems that are being modified are properly removed from service so that they are not used by operations until such time as the modification is tested and complete.

1.6 Equipment setpoints, commercial grade dedication activities, and scaling calculations are not controlled via this procedure.

1.7 This procedure provides cyber security guidance for power plant applications and systems.

2. DISCUSSION

2.1 The guidelines set forth in this procedure apply to software that produces or manipulates data used directly in the design, analysis, and operation of plant structures, systems, and components. The application of specific requirements shall be prescribed if needed in each system's respective Software Configuration Management Plan (SCMP) and/or Software Quality Assurance Plan (SQAP).

2.2 The definitions listed in Section 3 pertain to this procedure only. Various items may have different meanings in different environments so in order to preserve the clarity of this procedure it was deemed more beneficial to list them here rather than on the more global level of CF2.

3. DEFINITIONS

3.1 Application Sponsor (AS): The AS is the individual assigned by plant management that is the owner of the functional requirements for a system. For plant systems, an AS is not usually required or assigned as the functional requirements are a part of the plant design and are controlled via CF3. For data acquisition, test, and data retrieval applications, the AS is an individual that represents the users of the system, and is knowledgeable of what is required of the system.

3.2 Baseline Configuration: A named collection of hardware, software components, and supporting documentation that is subject to change management and is upgraded, maintained, tested, statused, and obsolesced as a unit. The baseline configuration can only be changed through the formal change control methods provided within that system's software quality assurance plan. For purposes of this procedure, the term baseline will refer to a "release" baseline, i.e. software that has been fully V&V'd and is installed and operating in its final running configuration.

3.3 Configuration Item: Configuration items include hardware, software, data, or documents that are uniquely identified and controlled as a single entry in the configuration management process. Examples include an EPROM, a hardware test plan, application source code, or a database.

3.4 Continuity of Power System: Those systems having a direct immediate impact on continuity of operation, the ability to generate electric power. They are systems that may cause an immediate reactor trip or systems that are required by regulations for personal safety and other commitments that would force a decision to shut down the plant.

3.5 Critical Digital Asset (CDA): A digital device or system that plays a role in the operation or maintenance of a critical system and can impact the proper functioning of that critical system. A CDA may be a component or a subsystem of a critical system; the CDA may by itself be a critical system; or the CDA may have a direct or indirect connection to a critical system. Direct connections include both wired and wireless communication pathways. Indirect connections include pathways by which data or software are manually carried from one digital device to another and transferred using disks or other modes of data transfer.

3.6 Cyber Security: Cyber security is the program implemented to prevent damage to, unauthorized access to, and allow restoration of computers, electronic communications systems, electronic communication services, wire communication, and electronic communication, including information contained therein, to ensure its availability, integrity, authentication, confidentiality, and nonrepudiation. In control systems this would include unauthorized access that could affect operation of plant structures, systems, or components.

3.7 Data: Data is defined as representation of facts, concepts, or instructions in a manner suitable for communication, interpretation, or processing by humans or automatic means. Data may be distinguished from software in that they are not instructions (i.e., data alone cannot not be used to operate a computer system or provide a desired function or performance). Data is acted upon or is the result of computer program activities.

3.8 Defensive Model: A definition or graphical representation included in the defensive strategy based on the nuclear industry practices of defense-in-depth, including detailed criteria for connectivity within and between defensive layers.

3.9 Defensive Strategy: Documented collection of technology, administrative processes and programmatic processes that ensure the appropriate level of security at each level in the defensive model.

3.10 Firmware: Firmware is the combination of a hardware device, computer programs, and data that are stored in read-only software on that device.

3.11 Functional Requirement: An item can be called a functional requirement only if its achievement can be verified and validated. Functional requirements shall be traceable throughout the remaining stages of the software development cycle.

3.12 Functional Requirements Specification (FRS): An FRS is a summary of requirements that specifies exactly what functions a computer system is required to perform. For plant systems, this is generally produced by design engineering. For data acquisition or retrieval systems, this document defines the specifics of what the user wants the system to do.

3.13 Hardware: Hardware is physical components of a computer system. This includes but is not limited to central processing unit(s) (CPU), math or other co-processors, volatile and non-volatile memory, keyboards, displays, printers and other input/output (IO) devices.

3.14 Incident Handling and Response Plan: An organized approach to addressing and managing the aftermath of an information security breach or cyber attack.

3.15 Nuclear Significant System: Those critical systems that can impact public health and safety through an adverse impact on safety, security or emergency response of nuclear power plants. Included in this category are (1) safety-related systems, including auxiliary systems that support safety systems and are required by the safety systems to accomplish their safety functions; (2) systems important to safety systems; or (3) site security; and (4) emergency response systems, including offsite communications.

3.16 Quality Assurance: A planned and systematic pattern of all actions necessary to provide adequate confidence that the structure, system or component will perform satisfactorily in service.

3.17 Software: Software consists of all portions of a computer system that are not hardware or data. Software includes application programs, operating systems, and support programs.

3.18 Software Change Package (SCP): The documentation developed by the system coordinator responsible for the software. The SCP tracks completion of all tasks required to transition from one baseline to another. This may be implemented either as a paper document or using the SAP notification or a minor maintenance or full scope SAP order.

3.19 Software Design Description (SDD): The SDD describes the specifics of the software design. All algorithms, equations, database, global memory structures and their utilizations should be documented here. Software objects and modules should have their public interfaces defined in this document. Hardware interfaces timing parameters, storage requirements, and protocols are detailed here. The SDD is detailed enough to serve as a programmer's guideline.

3.20 Software Design Specification (SDS): The SDS lays out the high level software design and lists all object and module interfaces. For small projects, this may be a part of the SDD. For large projects, especially those involving multiple programmers or programming teams, this will generally be a separate document that assures that the individual components will interoperate properly when combined during the integration phase of the project. If needed or useful, object diagrams or models, block diagrams, or flowcharts can be developed to show software module relationships.

3.21 Software Requirement: An item can be called a software requirement only if its achievement can be verified and validated. Software requirements shall be traceable throughout the remaining stages of the software development cycle.

3.22 Software Requirements Specification (SRS): A summary of software requirements for the system. This document may amplify or explain functional requirements in software terms and it may include additional requirements determined by the software designer to be necessary to proper operation of the system. This document may not be required for systems with detailed functional requirements, or extremely simple systems.

3.23 System Coordinator (SC): The individual(s) that coordinate(s) activities related to procurement, development, maintenance, and operation of a plant computer system.

3.24 User Manual: A document that provides the instructions and other information needed to operate the system to the end user.

3.25 Validation: The process of evaluating a software baseline to ensure compliance with functional and software requirements.

3.26 Verification: The process of determining whether or not the products of a given phase of the software development cycle fulfill the requirements established during the previous phase.

3.27 Out of Service (OOS): A computer system or application is OOS when the appropriate units SFM has approved working on it until the SFM has been notified that the change package is complete and the system or application is available for use by operations.

4.

RESPONSIBILITIES

4.1 The Application Sponsor (AS) is responsible for:

4.1.1 Functional requirements for assigned computer systems.

4.1.2 Evaluation of problem reports and change requests to determine further action is required.

4.1.3 Works with SC to ensure that V&V activities are adequate to ensure that a configuration baseline fully meets its requirements.

4.2 Configuration Management (CM) program owner is responsible for acting as a Single Point-of-Contact (SPOC) for Diablo Canyon Power Plant configuration management issues with full accountability for the Configuration Management Program.

4.3 Software Quality Assurance (SQA) program owner is responsible for acting as a single point-of-contact (SPOC) for Diablo Canyon Power Plant software quality assurance issues with full accountability for the SQA program.

4.4 Cyber security program owner is responsible for acting as a single point-of-contact (SPOC) for Diablo Canyon Power Plant cyber security issues with full accountability for the Cyber Security Program.

4.5 Design engineering is responsible for the hardware design and the functional requirements specification for all plant computer systems.

4.6 Project engineering is responsible for developing the software requirements specification and the software design for all plant computer systems, with overall responsibility for the implementation phase including developing the Software Configuration Management (SCM) plans and Software Quality Assurance (SQA) plans for plant systems.

4.7 Digital systems engineering (EID) responsibilities include, but are not limited to:

4.7.1 Enforcing and maintaining Software Configuration Management Plans (SCMPs) and Software Quality Assurance Plans (SQAPs) for plant systems.

4.7.2 Maintaining the systems throughout the operations and maintenance phases.

This group will also work in conjunction with design engineering and project engineering during the requirements and design phases of system development.

4.8 Post maintenance testing engineering responsibilities include, but are not limited to, determining testing required to ensure correct system function prior to being placed in service.

4.8.1 This testing is generally required for new system installation or major modifications to existing plant systems.

4.9 System coordinators responsibilities include, but are not limited to:

4.9.1 Performing technical and LBIE reviews of plant computer system software additions/modifications.

4.9.2 Developing SCP(s) per this procedure and the respective system SCMP and SQAP.

4.10 Managers are responsible for reviewing all new software configuration management plans and software quality assurance plans within their respective organizations.

4.11 SCM/SQA expert panel, consisting of members from design engineering, digital engineering, and information technology, with other line organization representatives consulted on an as-needed basis, are responsible for:

4.11.1 Supporting the CM and SQA program owners in application assessments, program changes, and program information.

4.11.2 Being points of contact for plant personnel on program requirements or questions.

5. INSTRUCTIONS

5.1 This procedure will be used for SCMP and SQAP development in accordance with CF2.ID9 if the computer system in question will be used for plant operational purposes per the scope section of this procedure.

5.1.1 CF2.ID3 will be used for business related systems and applications.

5.2 Attachment 7.1 is provided as a guide in determining whether SCM and SQA plans are needed for a given system, component or application.

5.2.1 This is provided in a checklist format, but need not be retained as it is provided only as a guideline.

5.3 The evaluation of a computer based system or component that determines the need for SCM and SQA plans should be documented on a notification.

5.3.1 The notification should be assigned to the EID group with instructions to include the system/component in the plant software database.

5.4 For questions or assistance on SQA issues, contact the SQA program owner. For questions or assistance on SCM issues, contact the CM program owner.

5.4.1 The program owners may refer questions to the SCM/SQA expert panel when current rules and policy don't cover the answer.

5.5 Software development shall proceed in a traceable, planned, and orderly manner.

5.5.1 The number of phases and relative emphasis placed on each phase of software development will depend on the nature and complexity of the software.

5.5.2 Based on the specific characteristics of a software modification or the generation of a new application, some of the listed phases may not be required.

5.6 Each plant system SCMP/SQAP will determine what level of documentation is required for a specific modification.

5.6.1 A notification and software change package shall be developed for each new software application or software modification.

5.6.2 Database/parameter changes may be documented within a notification.

NOTE: Care must be exercised in determining what constitutes a software modification. For example, if a database point is used by a computer application in determining how an action or calculation is performed, it may constitute a change to the application requiring an SCP.

a. Some system databases may include point types that perform operations on other points where if taken in aggregate, the combination of database points constitute an application that performs a calculation.

1. An SCP may be required if a change to the database constitutes a change to the way that a calculation is performed.

5.7 Prior to implementing a plant system software addition/modification, a License Basis Impact Evaluation screen shall be completed per TS3.ID2 and attached to the software change package.

5.8 Each plant system SCMP/SQAP will determine what level of coordination is required for a specific modification.

5.8.1 Depending upon the complexity of the software modification and the potential impact the change may have on a system, design engineering shall be sent a notification requesting the determination of the impact of the software modification on the original system design and associated documentation.

5.9 Each plant computer system SCMP/SQAP will provide guidance on how status of the system or it's applications is controlled.

5.9.1 A method of positive status control will be provided such that a computer system or application is formally OOS from the time that the first modification is performed until the system or application is fully tested and ready for use by operations.

5.10 CF2.ID11 provides guidance on assessing cyber security for digital assets. For systems assessed as CDAs for either Continuity of Power or Nuclear Significant systems, the SCMP/SQAP should provide guidelines as to:

5.10.1 What defensive strategies will be used to provide cyber security for the system and how they will be maintained.

5.10.2 How cyber security breaches will be addressed as part of the overall Incident Handling and Response Plan.

5.11 Problem Identification

Problems associated with plant computer systems, be they hardware or software related, shall be reported and documented per OM7.ID1.

5.12 Vendor Supplied Software

5.12.1 Each plant system SCMP/SQAP will include details for handling vendor or contractor supplied software. Vendor supplied software falls into two distinct types.

a. The first is software developed under contract specifically for its intended application.

1. Vendor or contractor supplied software of this type shall comply with this procedure's guidelines or a very similar software quality assurance plan.

2. The vendor should supply documentation at a level of detail and scope equivalent to what would be produced "in house" as part of the SCMP/SQAP process controlled by this procedure.

3. All software applications and/or modifications to said software performed by non-plant personnel shall be documented by an SCP or equivalent documentation.

b. The other type of vendor supplied software is Commercial Off the Shelf (COTS) software.

1. This is software that is in general use and not developed especially for the nuclear industry.

2. COTS applications contain pre-existing software that was developed to varying commercial standards, often through a more evolutionary than structured or pre-planned process and with less documentation than would be required by this procedure.

3. When COTS software is used in safety related plant systems, it shall be dedicated for its intended use per EPRI TR-106439, "Guideline on Evaluation and Acceptance of Commercial Grade Digital Equipment for Nuclear Safety Applications." This will normally be done as part of the design change process.

4. For non-safety related applications, the same type of evaluation shall be done at a level commensurate with the importance and complexity of the system.

5.12.2 The system SCMP/SQAP shall indicate what actions will be required to accept the software for its intended use. It may be desirable to apply the full EPRI evaluation for a turbine control system for the main generator.

a. An SCMP/SQAP for a non-quality digital recorder may require only a means for controlling the current firmware/software version and configuration.

5.13 Media Control

5.13.1 A system SCMP plan shall document how its media is controlled (labeling, storage, etc.).

5.13.2 The products associated with each software baseline should be stored and labeled to distinguish one from another.

5.14 Configuration Management

5.14.1 A system SCMP shall document how its software design, modifiable source, database, data files, and hardware are to be maintained and controlled.

a. The structure of each system is different, so the SCMP for each system will have to detail what is specifically required.

b. The process for system traceability must be clearly outlined so that anyone can readily trace from the current baseline configuration back to the initial baseline.

5.14.2 Each system SCMP shall define its initial baseline and the event that created it.

a. For most plant systems, the initial baseline will be documented as part of the design, and included in the plant drawings.

b. Items such as Functional Requirements and hardware design and configuration are a part of the living design of the plant and are modified per the requirements of CF3.

c. Lower level software design documents such as SRS, SDD, and Source Code may also be included in the drawing system. These are usually identified as archival drawings, and are maintained to document the initial baseline configuration of the system.

5.14.3 For new systems, lower level software design documents should be incorporated as configuration items into the current baseline configuration as separate entity from the archival version incorporated in the plant drawings.

a. When a new configuration baseline is established, all configuration items should be updated.

5.14.4 For existing systems, lower level software design documents should be brought into the baseline configuration as practicable, and updated to show new changes.

a. If earlier changes have been made, a notation should be made to indicate that earlier changes have been made that are not incorporated.

b. The SCMP should indicate the storage location of documentation for pre-existing changes.

5.14.5 The configuration items shall be identified and it shall be stated how each item and its various versions are to be uniquely identified.

a. A description of the activities performed to define, track, store, and retrieve configuration items shall be provided.

5.14.6 Instructions for maintenance of the cyber security defensive strategy for the system or application and its specific defensive model should be included in the system specific SCMP, as applicable.

a. The defensive model for a system should take into account the physical security of the plant and the physical security and defensive strategy of any system it is connected too.

5.14.7 A Software Configuration Management Plan (SCMP) shall identify:

- The software products to which it applies
- Show the current software configuration of the application/system is documented and maintained
- The organizations responsible for performing the work and achieving software quality and their tasks and responsibilities
- Required documentation
- Standards, conventions, techniques, or methodologies which shall guide the software development, as well as methods to assure compliance to the same
- The required software reviews
- Methods for maintaining cyber security of the system
- Methods for assuring proper status control for the system and it's applications during the modification process.
- The methods for error reporting and corrective action

5.14.8 Once an SCMP or SQAP is developed and approved, it shall be stored and maintained under revision control within EDMS in the NPG Library:/Engineering/Digital Systems Engineering directory.

5.14.9 Changes to an approved SCMP or SQAP require the performance of a LBIE Screen, per TS3.ID2, and approval of the owner/supervisor of the SCMP/SQAP.

5.15 Cyber Security

5.15.1 Cyber security defensive strategy shall be incorporated into the design of all new computer based components installed at DCPP that meet the definition of a CDA.

a. Defensive strategies developed for CDAs already installed at DCPP should have their design documents updated to include defensive strategies as appropriate.

b. Design documents shall be updated if a modification is performed that requires a change to the Functional Requirements of the System or Application.

5.15.2 Where systems communicate between computers, defensive strategies shall be developed to prevent data corruption from effecting important displays or calculations. This may range from simple checksum verification to encryption technology depending on the importance of the data and the system.

5.15.3 A defensive strategy should be developed and maintained as part each phase of the software life cycle. This should define the system level interfaces and boundaries used by the system or application as a part of the overall defensive strategy and model developed for the site.

a. Functional and software requirements should specify the required level of connectivity, separation, and protection of the system/application including:

1. Define requirements for level of acceptable risk at each layer within the system.

2. Define connectivity required for acceptable operation of the system or application.

3. System detection of and response to unauthorized access.

b. The design phase should develop a defensive strategy utilizing the defensive model for connected systems as well as a defensive model developed for the system or application under development.

1. This should include both the hardware and software designers as applicable.

c. The implementation phase should implement the cyber security design as it would any other component of the design.

1. Protective features implemented may include procedures, training, or user manuals as well as software and hardware based protections.

5.15.4 V&V should include verification that appropriate defensive strategies have been incorporated into the design, and validation that the protections specified in the requirements specification and the design are present and meet the requirements.

a. Validation testing by simulated attack on systems or applications shall only be performed on lab or development systems or on installed systems that are out of service and isolated in such a fashion that failure of the defensive strategy to protect the system will result in no adverse effects on inservice plant structures, systems, or components.

5.16 Software Life Cycle

5.16.1 The steps listed below describe the software life cycle for both a new power plant application and a modification to an existing application. Depending upon the complexity of a system, its safety significance, and the modification in question, all of the steps or a subset thereof shall be discussed within the respective SQAP.

5.16.2 Problems or inconsistencies discovered during any phase of the life cycle may require revisiting earlier phases. The earlier in the life cycle that problems are discovered and corrected, the less the impact on the overall development cycle and on overall system quality they will have. The life cycle should be considered an iterative process where problems and inconsistencies discovered in a given phase are fed back to appropriate earlier phases to eliminate them so that the overall consistency of the design is maintained.

5.16.3 If an incremental development process is used where the development phases of the design lifecycle is performed multiple times for a project (identified subsets of the overall requirements are added each cycle), care must be taken to assure that scope is not increased in the process.

a. Project scope should be defined at the outset.

1. Any increase in scope should be:

a) Agreed to by the project team.

b) Approved by appropriate management.

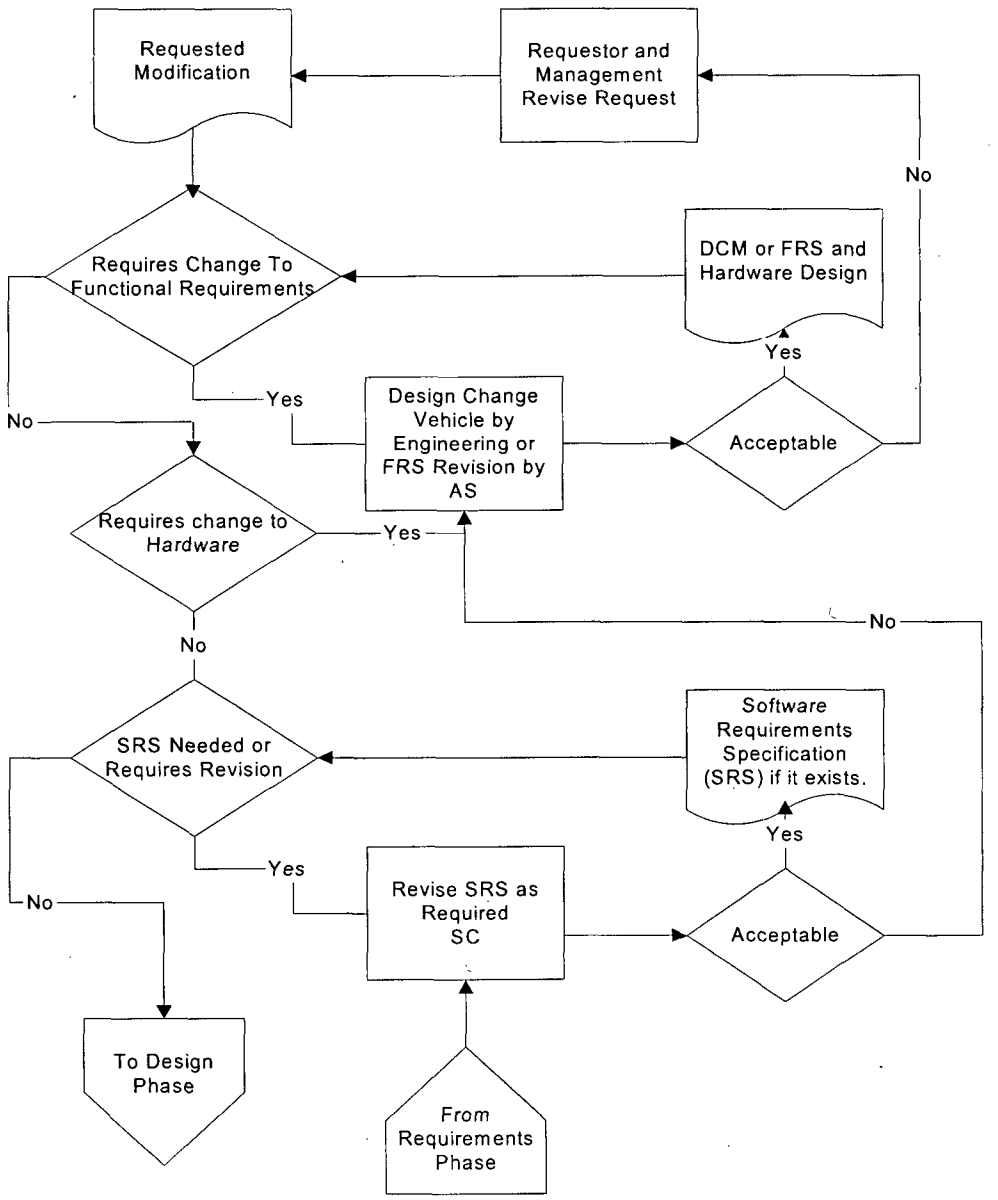
2. If increased scope is not vital to the success of the project, added scope should be:

a) Deferred.

b) Added after project completion. This is especially true as the project nears completion.

5.16.4 Requirements Phase – Figure 1 provides a flow chart for requirements phase tasks.

Figure 1: Requirements Phase Tasks



a. Functional Requirements

1. The functional requirements define what a system must do. They are the basis for both the hardware and software design.

a) The functional requirements shall include all interfaces including user input and output, hardware input and output, and interconnections to other systems including hardware and software protocols.

b) Functional requirements shall also include system response to changing inputs including indication, alarms, trips, storage, and outputs to other systems.

c) The functional requirements should also include performance requirements such as maximum processing cycle time, maximum response time for output due to significant input changes, transient response, and data storage amount and rate.

2. The functional requirements are the responsibility of the design engineering group for plant systems or the AS for data acquisition and support systems that fall under this procedure.

a) Having the design engineering group or the AS that requested the modification sign off on the requirements document/section produces a contract between the implementing and requesting groups which becomes the "officially agreed upon guideline" for the rest of the project.

b) Having an accurate requirements document will prevent the life of a project from increasing unnecessarily – saving both time and money and ensuring that the quality and completeness of the final product can be assured.

b. Software Requirements

1. The SRS may be used to further define how the software will function. It is normally the responsibility of the implementing organization. It may provide specific requirements for hardware interfaces, operating system, modularity, algorithms and data management and storage.

2. Each requirement shall be defined such that its achievement is capable of being objectively verified and validated by a prescribed method.

3. Documentation generated during this phase may include a functional requirements specification and/or a software requirements specification.

4. The software verification and validation plan may take form at this stage but will be finalized within the testing phase.

5.16.5 Design Phase Figure 2 provides a flow chart for Design Phase Tasks.

NOTE: Products of the requirements phase should be frozen at this point. Requirements should only be changed if the design phase determines that the requirements are inadequate or in error. Adding requirements subsequent to entering the design phase may cause significant additional resources and rework, and can make the difference between a successful project and a project that comes in late and over budget.

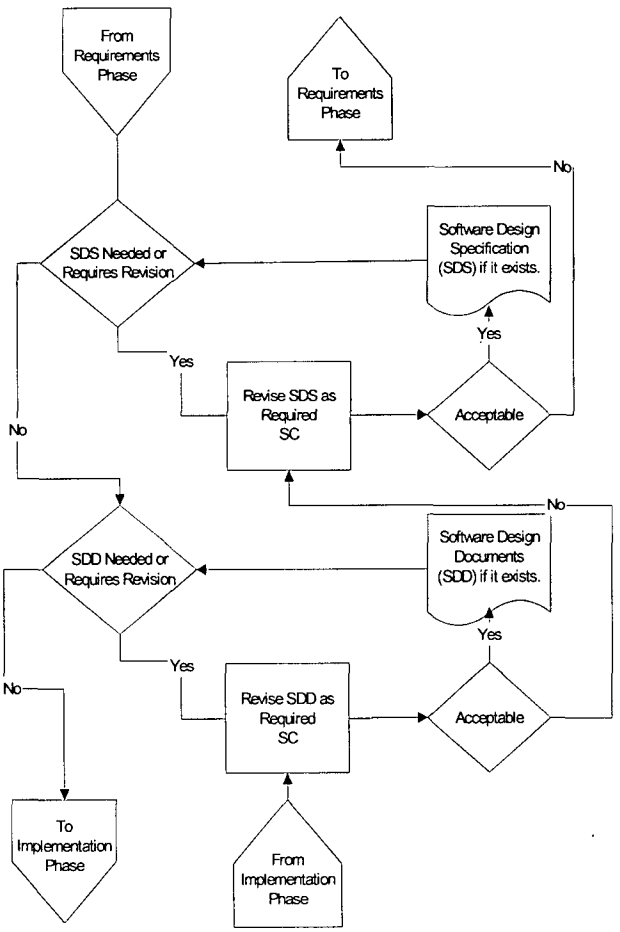
a. A software design based on the requirements shall be developed, documented, and reviewed.

1. The design shall specify the overall structure (control and data flow) and the reduction of the overall structure into physical solutions.

2. The design may necessitate the modification of the requirements specification(s).

3. The design shall be reviewed to verify that all of the requirements are being addressed.

Figure 2: Design Phase Tasks



b. Hardware, system software and support software should be examined for potential interfacing effects on the project. The following should be defined:

- The nature of the interface
- The affected organizations
- How the interface code, documentation, and data are to be controlled

c. Documentation produced during this phase includes the software design specification and the software design description, and a LBIE screen.

5.16.6 Implementation Phase

NOTE: Products of the design phase should be frozen at this point. The design should only be changed if the implementation phase determines that the design is inadequate or in error. Revising the design subsequent to entering the implementation phase may cause significant additional resources and rework, and can make the difference between a successful project and a project that comes in late and over budget.

a. The designs shall be translated into a programming language and the implemented software shall be analyzed to identify and correct errors.

b. User documentation and manuals, programmer aids, and other implementation dependent documentation shall be produced/modified during this phase.

c. Verification activities shall consist of the examination of computer program listings to assure adherence to agreed upon coding standards and conventions and that all documentation is correct and clear and is consistent with the software.

5.16.7 Testing Phase

a. The design as implemented in code shall be exercised by executing the test cases.

1. Failure to successfully execute the test cases shall be reviewed to determine if modifications of the requirements, design, implementation, or the test plans and test cases are required.

2. The V&V plan that may have been initialized during the requirements phase will be finalized during this time frame.

b. Testing phase activities shall consist of code validation to assure adherence to the requirements and that the software produces correct results for the test cases.

1. To evaluate technical accuracy, the software test case results can be compared to results generated via alternative methods such as:

- Analysis without computer assistance
- Other previously validated computer programs
- Experiments and tests
- Standard problems with known solutions
- Confirmed published data and correlations

5.16.8 Software Verification and Validation Plan

a. Software verification shall be performed throughout the development process to ensure that the products of the current life cycle phase comply with the requirements of the previous phases.

b. Software validation ensures that the end product satisfies all the functional requirements specified in the design documents.

c. Software V&V activities must be planned and performed for each system configuration that may impact the software.

1. Software V&V should be performed by individuals other than those that programmed the software and the results documented.

d. V&V activities shall:

- Ensure that the artifacts of each phase of the development cycle are correct and are consistent with the artifacts of earlier phases
- Ensure that the software adequately and correctly performs all required functions
- Ensure that the software does not perform any unintended function that either by itself or in combination with other functions can degrade the entire system
- Ensure that all requirements have been addressed
- Ensure that changes to user documentation, manuals, and procedures are complete and correct

5.16.9 Installation and Checkout Phase

a. The new/modified software becomes part of a system incorporating applicable software components, hardware, and data. The process of integrating the software may consist of installing hardware, installing the program, reformatting or creating databases, and verifying that all components have been included.

b. V&V activities shall consist of:

- The execution of tests for installation and integration
- The documentation of the approval of the software for operational use

c. The requestor of the modification will sign off that the software is acceptable. This establishes the current released baseline. If this is a new plant application/system it will now be added to the component database.

d. If the change required changes to user documentation, manuals, or procedures, these changes shall be incorporated into the appropriate drawings via the FCT process and plant manuals via an OTSC or Revision as required per the appropriate procedures.

5.16.10 Operation and Maintenance Phase

a. At this point the software has been approved (certified) for operational use. Further activity shall consist of:

- Corrective maintenance – removal of latent errors
- Perfective maintenance – responses to new or revised requirements
- Adaptive maintenance – adapting the software to changes in the operating environment

b. Requirements for software modifications shall be documented per OM7.ID1. The modifications will be approved, documented, verified and validated, and controlled per the respective system SCMP and SQAP.

c. Periodic testing of plant software is not required if all functions of the software have been tested prior to baselining the system, and any modifications have been created and tested per this procedure and the corresponding system SCMP and SQAP.

1. Plant computer systems are controlled per OM7.ID1, CF1, and the family of CF2 procedures.

2. No changes are made without the cognizance of the system coordinator.

5.16.11 Disaster Recovery

a. Methods shall be established to prevent the loss of software, data, and other pertinent components in case of a disaster or computer failure.

1. Each SCMP/SQAP should include guidance on frequency and scope of backup activities to meet the above requirement.

2. A minimum of two copies of each backup should be made, each stored at separate geographic locations.

b. The frequency of backups will depend upon the number of software modifications made and will be made at the discretion of the system coordinator.

c. Each plant software quality assurance plan will describe the methodology used to implement disaster recovery actions.

5.16.12 Retirement Phase

a. The software product is removed from operational use and support for it is terminated. All users of the software product shall be notified and the component database updated accordingly.

6. RECORDS

6.1 A logbook should be maintained for each plant computer system by the respective system coordinator for systems that would benefit from having all the software, hardware, and database changes located in one place.

6.1.1 Any system modification (hardware or software) will be documented within the logbook.

6.1.2 The current logbook shall be located in the immediate vicinity of the respective plant computer system.

6.1.3 Past logbooks should be retained by the system coordinator in system files for the life of that plant computer system.

6.1.4 Plant computer system logbooks do not require storage on micro film (RMS).

6.2 The following documentation shall be maintained per AD10.ID1 "Storage and Control of Quality Records:"

- SCM Plans
- SQA Plans
- Disaster Recovery Plans

7. ATTACHMENTS

7.1 "SCMP/SQAP Requirement Checklist," 06/16/10

8. REFERENCES

8.1 AD3.ID5, "Procedure Distribution and Control"

8.2 CF1, "Configuration Management"

8.3 CF2, "Computer Hardware, Software, and Database Control"

8.4 CF2.ID3, "Software Management for Business Information Computer Systems"

8.5 CF2.ID9, "Software Quality Assurance for Software Development"

8.6 CF2.ID11, "Cyber Security Assessment of Critical Digital Assets"

8.7 OM7.ID1, "Problem Identification and Resolution"

8.8 TS3.ID2, "Licensing Basis Impact Evaluations"

8.9 ASME NQA-1-1997, "Quality Assurance Requirements for Nuclear Facility Applications"

8.10 10 CFR 50 Appendix B, "Quality Assurance Criteria for Nuclear Power Plants and Fuel Reprocessing Plants"

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

8.12 IEEE Std. 730-2002, IEEE Standard for Software Quality Assurance Plans

8.13 EPRI TR-106439 – Guideline on Evaluation and Acceptance of Commercial Grade Digital Equipment for Nuclear Safety Applications

8.14 QA Commitments

8.14.1 FSAR Chapter 17.2, 17.3

8.14.2 Reg Guide 1.64, 4.15

NUCLEAR POWER GENERATION
CF2.ID2
ATTACHMENT 7.1

TITLE: SCMP/SQAP Requirement Checklist

		Yes	No
1.	Determine if the system/software falls under CF2.ID2		
	a. Is the software installed in a Safety Related system?	[ ]	[ ]
	b. Is the software used to control SSCs in the power plant?	[ ]	[ ]
	c. Is the software used to provide information to plant operators that is used to operate or make operational decisions about the plant?	[ ]	[ ]
	d. Is the software part of a fixed or portable data acquisition system that is connected to plant equipment whose output is used to make operational or engineering decisions important to the safe and reliable operation of the plant or to meet a regulatory commitment?	[ ]	[ ]
	e. Is the software part of a data storage or retrieval system used to store real time or historical plant data that may be used to make operational or engineering decisions important to the safe and reliable operation of the plant or to meet a regulatory commitment?	[ ]	[ ]
	A yes answer to any of the above questions indicates that CF2.ID2 is applicable to the software. If all questions are answered no, the software should be evaluated per the requirements of CF2.ID3.		
2.	Determine if SCMP and SQAP are required		
	a. Can the software program be changed by the company?	[ ]	[ ]
	b. If the vendor is not on the Qualified Suppliers List as an approved vendor for this software, is the software safety related or graded quality?	[ ]	[ ]
	c. Is the software Commercial Off-The-Shelf (COTS) software used in safety related systems where there is no other process in place to assure proper dedication of the software to its safety related function?	[ ]	[ ]
	d. Is there a configuration database associated with the system that is not adequately controlled by other plant procedures or drawings?	[ ]	[ ]
	e. Are there configuration items (both installed and backup) that are not already identified in plant drawings or procedures as to name, location, and version so that the current configuration of the system can be identified and if needed restored in the event of a system failure. In the event of a system failure, is there a requirement to load software and/or configuration data where there is insufficient guidance in plant procedures and drawings on how this is done. (Disaster recovery)?	[ ]	[ ]
	f. Is there any other reason that an SCMP and SQAP should be created for this software? This could include regulatory commitments, non-obvious testing requirements, or complexity of the system. Document in the operation.	[ ]	[ ]
	A yes answer to any of the above question indicates that an SCMP and SQAP should be created for this system/software.		