REVIEWS AND APPROVALS

Prepared by:

_____ _____
xxxxx Date
Software Project Manager

_____ _____
xxxxx Date
End-User Representative

_____ _____
xxxxx Date
Security Manager

_____ _____
xxxxx Date
Software Project Manager's Manager

_____ _____
xxxxx Date
Vice President, Quality

Approved by:

_____ _____
xxxxx Date
Contracting Officer's Representative

_____ _____
xxxxx Date
Program Manager, DOE

SOFTWARE CONFIGURATION MANAGEMENT PLAN
CONTENTS

CONTENTS (continued)

LIST OF REVISIONS

Issue Date	Revision #	Page/Section #	Change/Description
September 2000	0	i, 2-29	Initial Release
September 2001	1	i	Added James O. Mitchell as Reviewer.

1.0 PURPOSE

The purpose of this Software Configuration Management Plan (SCMP) is to provide guidance for establishing and implementing configuration management for the NMMSS Upgrade. It will set forth the policies, procedures and processes used to accomplish system Configuration Management (CM) for this project.

2.0 REFERENCES

2.1 NAC Documents

[1] NAC 135-PL-001, *NMMSS Project Plan*.
[2] NAC 135-PL-100, *NMMSS Records Management Plan*.
[3] NAC 135-PR-305, *Software Change Request Procedure.*
[4] NAC 135-PL-330A, *Software Project Management Plan*.
[5] NAC 135-PL-400A, *Software Quality Assurance Plan*.
[6] NAC 135-PL-410, *Project Quality Assurance Plan*.
[7] NAC 135-PL-500, *NMMSS Employee Training Plan*.

2.2 Non-NAC Documents

[1] *ANSI/IEEE Std 1028-1997, IEEE Standard for Software Reviews*.
[2] *ANSI/IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology*.
[3] *ANSI/IEEE Std 828-1998, IEEE Standard for Software Configuration Management Plans*.
[4] *ANSI/IEEE Std 1042-1987, IEEE Guide to Software Configuration Management.*
[5] U.S. Department of Energy *(DOE) Software Engineering Methodology (SEM), dated March 1999.*

2.3 Definitions and Acronyms

2.3.1 Definitions

Acceptance Testing - Formal testing is conducted to determine whether a system satisfies its acceptance criteria and to enable the customer to determine whether to accept the system (Std 610.12-1990).

AIS Security Testing - The process of testing an Automated Information System (NMMSS) to verify that the system meets all security requirements and receives accreditation to process U.S. Government classified information.

Anomaly - Anything observed in the documentation or operation of software that deviates from expectations based on previously verified software products or reference documents (Std 610.12-1990).

Critical Anomaly - An anomaly that must be resolved before system operations can continue.

Baseline - A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for future development, and that can be changed only through formal change control procedures (Std 610.12-1990).

Computer Software Configuration Item (CSCI) - An aggregation of software that is designated for configuration management and treated as a single entity in the configuration management process (Std 610.12-1990).

Configuration - The arrangement of a computer system or component as defined by the number, nature, and interconnections of its constituent parts (Std 610.12-1990).

Configuration Control - An element of configuration management, consisting of the evaluation, coordination, approval or disapproval, and implementation of changes to configuration items after formal establishment of their configuration identification (Std 610.12-1990).

Configuration Control Board (CCB) - A group of people responsible for evaluating and approving or disapproving proposed changes to configuration items, and for ensuring implementation of approved changes (Std 610.12-1990).

Configuration Identification - An element of configuration management, consisting of selecting the configuration items for a system and recording their functional and physical characteristics in technical documentation (Std 610.12-1990).

Configuration Item (CI) - An aggregation of hardware, software, or both, that is designated for configuration management and treated as a single entity in the configuration management process (Std 610.12-1990).

Configuration Management (CM) - A discipline applying technical and administrative direction and surveillance to identify and document the functional and physical characteristics of a configuration item, control changes to those characteristics, record and report change processing and implementation status, and verify compliance with specified requirements (Std 610.12-1990).

Configuration Status Accounting - An element of configuration management, consisting of the recording and reporting of information needed to manage a configuration effectively. This information includes a listing of the approved configuration identification, the status of proposed changes to the configuration, and the implementation status of approved changed (Std 610.12-1990).

Critical Software - Software whose failure could have an impact on safety, or could cause large financial or social loss (Std 610.12-1990).

Functional Configuration Audit (FCA) - An audit conducted to verify that the development of a configuration item has been completed satisfactorily, that the item has achieved the performance and functional characteristics specified in the functional or allocated configuration identification, and that its operational and support documents are complete and satisfactory (Std 610.12-1990).

Integration Testing - Testing in which software components, hardware components, or both are combined and tested to evaluate the interaction between them (Std 610.12-1990).

Physical Configuration Audit (PCA) - An audit conducted to verify that a configuration item, as built, conforms to the technical documentation that defines it.

Release - The activity of distributing an aggregation of CSCIs outside the development organizations.

Revision - The activity of making changes to a program that corrects errors in the logic or otherwise modifies software characteristics but does not affect the documented functional capabilities since none of the requirements have changed.

Significant Impact - A significant impact change is defined as any change that requires more than eighty (80) man-hours of software engineer effort.

System Requirements Validation Review - A review conducted to evaluate the completeness and adequacy of the requirements defined for a system.

Version - An initial release or re-release of a computer software configuration item, associated with a complete compilation or recompilation of the computer software configuration item.

Walk-through - A static analysis technique in which the responsible individual leads members of the development team and other interested parties through a segment of documentation or code. The participants ask questions and make comments about possible errors, violations of development standards, and other problems.

2.3.2 Acronyms

AIS	Automated Information System
ANSI	American National Standards Institute
ARF	Authority Reference File
CCB	Configuration Control Board
CI	Configuration Item
CM	Configuration Management
COTS	Commercial-Off-The-Shelf
CSCI	Computer Software Configuration Item
CSM	Computer Software Module
FCA	Functional Configuration Audit
IEEE	Institute of Electrical and Electronics Engineers
ISSM	Classified Information Systems Site Security Manager
LAN	Local Area Network
PCA	Physical Configuration Audit
QA	Quality Assurance
SAL	Software Archive Library
SCM	Software Configuration Management
SCMP	Software Configuration Management Plan
SCR	System/Software Change Request
SDD	System Design Description
SDL	Software Development Library
SML	Software Master Library
SPMP	Software Project Management Plan
SPR	Software Problem Report
SQAP	Software Quality Assurance Plan
SRS	System Requirements Specification
SRVR	System Requirements Validation Review
STP	System Test Plan
STCP	System Test Case/Procedures
TRR	Test Readiness Review
VDD	Version Description Document

3.0 CONFIGURATION MANAGEMENT ORGANIZATION

3.1 Project Organization

The organizational chart, in Appendix A of NAC 135-PL-001, for the NMMSS Project identifies the relationships of NAC project management, NAC software development personnel, and software quality assurance personnel as related to configuration management activities.

3.2 Configuration Management Authority

The authority for the implementation of the NMMSS Upgrade configuration management practices is derived from the Office of Plutonium, Uranium and Special Materials Inventory (SO-23) of the U.S. Department of Energy (DOE). The application of these practices and procedures will ensure each system change proposal meets program objectives, and is necessary or offers a significant benefit.

3.3 Software Configuration Management (SCM) Roles and Responsibilities

All authority for managing the NMMSS Project at NAC is vested with the Project Manager. The Senior Technical Manager and the Database Administrator are the primary interfaces between the NMMSS software development team, software testing, documentation management and software quality assurance activities. The Classified Information Systems Site Security Manager (ISSM) is designated as the Software Configuration Manager and is supported by the Database Administrator, Senior Technical Manager, Project Manager and the NAC Quality Assurance VP. The Software Configuration Manager administers the SCMP, provides configuration identification, configuration control and configuration status accounting services for the Project.

The general responsibilities for coordinating SCM activities include processing information needed to control changes to an application and associated documentation as they develop. These responsibilities include capturing the “as built” documentation, test data, test reports, and code for the NMMSS software baselines defined in Section 4.1 of this plan. The following matrix identifies the SCM activity responsibilities:

SCM Activity	Responsible Individual
Configuration Identification	Software Configuration Manager
Configuration Control	Software Configuration Manager
Configuration Status Accounting	Software Configuration Manager
Configuration Audits and Reviews	NAC QA Manager
Interface Control	Senior Technical Manager
Subcontractor/Vendor Control	Senior Technical Manager

3.4 Configuration Control Board (CCB)

The purpose of the CCB is to control major issues such as schedule, function, and configuration of the system as a whole. The CCB meets at the discretion of the chairperson. At a minimum, the CCB meets quarterly. As such, the CCB, in conjunction with the DOE Contracting Officer's Representative (for significant changes), is responsible for approval or disapproval of all changes to the system baseline. A copy of the CCB Charter is located in Appendix C of the plan.

The CCB includes the following NAC and DOE personnel or their designees. Each position listed below has voting privileges:

Configuration Control Board (CCB) Members	Description
CCB Chairperson	Project Manager
Database Administrator	Database Administrator
Software Configuration Manager	Classified Information Systems Site Security Manager (ISSM)
Software Project Manager	Senior Technical Manager
Security	Information Systems Site Security Officer (ISSO)
End-User Representative	Project Engineer
Software Engineer Representative	NMMSS Software Engineer
DOE Representative	DOE Contracting Officer's Representative[1]
NAC Quality Assurance	NAC Quality Assurance Manager

[1] Over-sight capacity for SCRs that meet the significant impact criteria.

3.4.1 CCB Roles and Responsibilities.

3.4.1.1 Non-Standing CCB Members

Non-standing CCB members will be representatives of areas not included in the standing membership, but whose specialized knowledge of a support or functional area is essential for

enhancement of the decision process. These members do not have voting privileges.

3.4.1.2 Standing CCB Members

Standing CCB members will consist of representatives from the NMMSS functional areas. These members will have voting privileges.

The following are the Standing CCB members roles and responsibilities:

DOE Contracting Officer's Representative responsibilities include the review and concurrence on all changes having a work estimate of 80 hours or greater. This role will not be required until the maintenance phase of the NMMSS Upgrade has been reached.

CCB Chairperson responsibilities include, but are not limited to:

- Acts as the final authority for CCB decision for Software Change Request (SCR) with estimates of less than eighty (80) man-hours.
- Presides over the CCB meeting.
- Reviews SCR for completeness.
- Designates an alternate in his or her absence.
- Designates a scribe for each CCB meeting.
- Updates the Software Change Request System stamping approved or disapproved on Software Change Request records discussed during the CCB meeting.

Software Configuration Manager, with the support of the Database Administrator, maintains the NMMSS software libraries, whether in hardcopy or magnetic media, as described below:

- The Software Development Library (SDL) contains software source code. The SDL resides on the NMMSS Test Server controlled by PVCS Version Manager software. This is the library used by NMMSS Software Engineers for the enhancement and maintenance of the NMMSS software.
- The Software Master Library (SML) contains software executables. The SML resides on the NMMSS Production Server controlled by PVCS Version Manager software. The SML contains the current baseline software with approved changes for the NMMSS. The Software Configuration Manager will control the access to this software.

- The Software Archive Library (SAL) is located on the NMMSS Test and Production servers and consists of the current baseline software with approved changes and previous baselines. The Production SAL is stored separately under the control of the Software Configuration Manager.

Software Project Manager responsibilities include, but are not limited to:

- Assigns priorities, time estimates and assigns SCR to a Software Engineer.
- Monitors progress on item approved by the CCB.
- Reviews SCR for completeness before CCB Meeting.
- Designates an alternate in his or her absence.

Database Administrator responsibilities include, but are not limited to:

- Recommends priorities, time estimates and assignment of SCR to a Software Engineer.
- Monitors progress on item approved by the CCB.
- Reviews SCR for completeness before CCB Meeting.
- Prepares agenda items for CCB meeting.
- Designates an alternate in his or her absence.

CCB Members (Security, End-User and Software Engineer) responsibilities include, but are not limited to:

- Recommends priorities, time estimates and the assignment of SCR to a Software Engineer.
- Monitors progress on item approved by the CCB.
- Reviews SCR for completeness.
- Designates an alternate in his or her absence.

Scribe responsibilities include, but are not limited to:

- Documents discussions during the CCB meeting.
- Issues minutes of the meeting to attendees and absent board members.

3.5 Configuration Management Policy and Procedures

The NMMSS Upgrade SCMP policy is to provide configuration management services to the project consistent with current DOE requirements and guidelines, as defined in this plan and associated procedures. This configuration management plan and supporting procedures are written in compliance with current DOE SEM guidelines.

Deviations from the NMMSS Upgrade SCMP and procedures may only be made with the permission of the NMMSS Project Manager with the support from the Software Configuration Manager. Configuration management will be used to invoke the required discipline for ensuring compliance with project requirements and to ensure the integrity of the NMMSS products.

3.6 Configuration Management Methodology and Tools

The NMMSS Upgrade configuration management methodology is prescribed within this plan and associated procedures. Configuration management will use tools derived from the Commercial-Off-The-Shelf (COTS) software products to create repositories and libraries that minimize the assignment of personnel, but still accomplish their mission. Later in the project, when and if automated tools might become necessary, migration will be accomplished. Tools will be assigned an identifier and tracked as part of the configuration management environment.

4.0 SCM ACTIVITIES

4.1 Configuration Identification (CI)

Baselines are an effective mechanism to establish, develop, and maintain milestones for review, testing, and release on the Project. A Computer Software Configuration Item (CSCI) is defined as a computer program, COTS and associated documentation that satisfies an end use function or aggregate of a function. The NMMSS Upgrade CSCI is uniquely identified by an assigned label. The configuration identification of the CSCI is accomplished in the aggregate of specification, computer listings, and other documentation used to describe the characteristics of the CSCI. The acceptance of the documentation at specific times throughout the software life cycle is known as “establishing baselines.” A baseline, plus approved changes, constitutes the current configuration identification for the CSCI.

There shall be four (4) baselines:

- Functional Baseline
- Allocated Baseline
- Development Baseline
- Product Baseline

As a result of the Product Baseline completion, the Functional and Developmental Baselines are established. Each baseline represents the completion of a milestone and forms the basis for the next phase of the software development process. The Allocated Baseline is established by development and issuance of the submitted draft System Design Description (SDD). The items to be controlled on the NMMSS Project under each Baseline follow:

Baseline	Contents
Functional (or requirements)	System Requirements Specification (SRS)
Allocated (or design)	System Top Level Design Document Draft System Design Description (SDD) Associated SRS
Developmental (or testable)	System Test Plan (STP) System Test Case/Procedures (STCP) Associated SRS and Draft SDD Computer Software (source and executable code) Test Data Base(s) Developmental Hardware
Product (or delivered)	User Manual (Operating Procedures) Associated SRS, Final SDD, STP, STCP Source Code Source Code Listing Executable Code Initial Data Base(s) Test Data Base(s) Regression Test Database(s) Acceptance Test Report System Test Report Integration Test Report Operational Hardware

4.1.1 Identifying Configuration Items

The Functional Baseline (requirements baseline) is formally defined by the entry of the SRS into the SML immediately upon final approval of the SRS by the customer.

The Allocated Baseline (design baseline) will be initially defined by the entry of the SDD into the SML immediately upon approval by the customer.

The Developmental Baseline (testable baseline) comprises all internally approved software products, including those products pending customer approval. After customer review and approval of each software product or

item at the corresponding Walk-through and/or review, the Developmental Baseline is established as the source code and associated software products are entered into the SML. As software modules pass module test reviews or software items and software products are approved at Walk-throughs, they are entered into the SML for control as elements of the Developmental Baseline. The Developmental Baseline is modified, using the SCM change control process, as new and updated software products are created and approved. The Developmental Baseline becomes part of the delivered baseline (Product Baseline) after successful completion of the Test Readiness Review (TRR).

The NMMSS Product Baseline (delivered baseline) is established with the approval of the NMMSS FCA and PCA. The Product Baseline comprises all internally approved software products, including those pending NMMSS customer approval and is entered into the SAL.

4.1.2 Naming Configuration Items

All supporting documentation generated for the NMMSS Project is identified in a manner that is consistent with NMMSS Records Management Plan, NAC 135-PL-100.

The basis for labeling is by mnemonic labels assigned by the NMMSS Upgrade Database Administrator to each software product. The format for identification will be the DD-AA-000, where:

- DD is the two-character identifier for the software product, assigned from Table 1 below.
- AA is the two-character major version identifier, alpha or numeric.
- 000 is the three-digit minor version control suffix.

Configuration items have unique names that correspond in a meaningful way to the functions that they perform. A list of the two (2) character identifiers for the products are found in the table below:

Document Nomenclature	Mnemonic (DD)
Software Project Management	*
Software Quality Assurance Plan	*
Software Configuration Management Plan	*
Software Requirements Specification	*
Software Design Description	*
System Test Plan	*
Software User Manual (Operating Procedures)	*
Source Program	SP
Executable Program	EP
Program Documentation	PD
Software Documentation	SD
Software/System Test Data	ST
* identifier assigned in accordance with NMMSS Records Management Plan, NAC 135-PL-100	

4.2 Version Control

A version is an identified body of software and documentation. It is the initial release or re-release of a software configuration item. When a body of software is generated to satisfy a specific purpose, all component parts are identified at the level at which it exists at the time of release. The totality of that software release is called a version. The version is documented in the form of a Version Description Document (VDD), which accompanies the release. As a system matures, the baseline software is inevitably changed to accommodate problem fixes or new capabilities.

The CCB has the responsibility of approving the release of a new version and its contents. When a new version is released, CM will maintain copies of all superseded software versions for purposes of an audit trail or for the possibility that an older software version might need to be regenerated.

Version Control is the process of generating new software versions and maintaining control of older versions. Version Control will be in effect after establishing a baseline coded system.

4.2.1 Production Change Releases

4.2.1.1 System Modification

Once the baseline for the NMMSS Upgrade has been established, modifications to the system may be identified to further streamline the NMMSS Upgrade automated support of program operations. The modifications will involve changing the way in which the system supports existing functions. Examples of system modifications include, but are not limited to, the following:

- Changing the wording of an error message on a window.
- Deleting a display field from a window.
- Modifying the sequence in which columns of data appear on a report.

4.2.1.2 System Enhancements

Once the baseline has been established, additional system functionality that would improve the NMMSS Upgrade automated support of program operations may be identified. This type of change is referred to as an enhancement. System enhancements generally require a larger level of effort than for system modification. They are also planned and scheduled. Examples of system enhancements include, but are not limited to the following:

- Providing one or more additional standard reports.
- Development of an application outside of the NMMSS.
- Upgrade to improve performance of the NMMSS.

4.2.1.3 Software Modification and Enhancement Release Methods

The development team will make all changes to the existing NMMSS Upgrade baseline in the NMMSS test environment. All changes resulting from system problems, system modifications, and/or system enhancements will undergo a functional test by a designated member of the development team. The change will also undergo a full test by a designated end-user before being moved into the NMMSS production environment.

The software release method will be determined by the category of the SCR. One or more of the following methods will be employed depending on the conditions of the SCR and the preference of the CCB:

SCR Type	Description	Technical Term	Release Method
1	Immediate	Continuous Cutover	Changes to a baseline system are implemented continuously upon occurrence.
2	First Availability	Block Release	Changes are put into the baseline as a group, when warranted by the size of the group.
3	Monthly	Block Release	Changes scheduled to be placed into the baseline as a group at the end of the month.

Other software release methods that may be employed are as follows:

SCR Type	Description	Technical Term	Release Method
		Periodic Update	Baselines are revised in response to a regularly scheduled review performed to identify any required changes.
		Phase Approach	A large change is divided into manageable subtasks that are implemented sequentially to minimize disruption to the baseline.

4.3 Configuration Control

The configuration control functions are requesting, evaluating, approving or disapproving, and implementing changes to baseline CSCIs. Changes encompass both error correction and enhancement. The SDL is the NMMSS project's working library and is freely accessible at any time to the software development team. The Project Manager manages and controls all design documents.

The following sections describe the policies and procedures to be employed to effectively control changes to any configuration item that is a member of the SML or the SAL. The software change control system applies to software products and configuration items upon acceptance into these libraries as part of a baseline. Changes to the products and items under configuration control are made in accordance with the established change control procedures as defined in the Software Change Request Procedure, NAC 135-PR-305. Products or items in these libraries are not changed without an approved SCR. The following description summarizes the configuration control process. Appendix C contains the SCR process flow diagram.

4.3.1 Requesting Changes

Change requests shall include a description of the need for change and a description of the desired change. Affected configuration item(s) and revision(s) shall be identified along with the requestor name, date and an estimated priority or urgency of the change. Priorities are:

- **Critical** - a software change request considered essential to the system.
- **Very Important** - a software change request that would enhance system performance.
- **Important** - a software change request would be beneficial to system users but not impact system operation or performance.
- **Non-Critical** - a software change request that is desirable to implement as time permits.

SCRs also include a description of the circumstances surrounding the occurrence of the problem and the hardware configuration in which it was experienced. SCRs are numbered for status tracking purposes. Status and disposition of change requests are indicated on the SCR form.

The customer, users, or NMMSS employees may request a change. Changes to software products controlled in the SML or SAL are identified by an SCR that either documents a problem or describes a requested change. The requestor enters the change request into the electronic SCR tracking system. The Database Administrator reviews each request for completeness and prepares the SCR for evaluation.

Tracking of SCRs is accomplished by use of an electronic log.

4.3.2 Evaluating Changes

The Database Administrator or designated Software Engineer analyzes the SCR and performs a preliminary investigation for resolution. Each change request is evaluated in terms of enhancement to system functionality and its effect on project resources. Any potentially significant impact to schedule or cost is clearly identified by the Database Administrator or designated Software Engineer and documented on the SCR. Significant impact is defined as 80 man-hours or greater of effort. The SCR is presented to the CCB with recommendations and a change of priority, if appropriate. Additionally, any recommended changes affecting system security are evaluated by the ISSM.

4.3.3 Approving or Disapproving Changes

The CCB reviews and accepts or rejects proposed changes. If there is agreement on the change(s), the change request is approved by the CCB. If the request is denied by the CCB, the reason is noted on the form and filed in the SCR binder as well as electronically, and the requestor is notified. It takes a quorum consisting of a Chairperson, Database Administrator or Software Project Manager, Security and End-User to approve or disapprove a SCR.

DOE Program Manager approval of an SCR is required on any proposed change estimated to have significant impact as defined in Section 3.3.2 above. The CCB will meet a minimum of once monthly.

4.3.4 Implementing Requests

Necessary activities for verifying and implementing an approved change include producing a revision of the configuration item (software, hardware

and/or document), and either testing or reviewing the affected item(s). The updated configuration item(s) has revised release and/or revision numbers.

After an SCR has been approved, implementation proceeds. A change is deemed complete after it is thoroughly tested and updates have been made to the effected baseline and inserted into the SML or SAL, as appropriate, by the Software Configuration Manager.

For those changes that require an update to the system documentation, the Software Configuration Manager provides a copy of the SCR to the Project Administrator who maintains a file and ensures all updates to documentation are incorporated into the plan or procedure no later than the annual review.

4.3.5 Testing Plan

The Software Engineer effecting the changes required by the SCR and End-user develops a testing plan and documents this plan on the SCR. The selected End-user executes the test plan and indicates the results on the SCR. The completed and tested SCR is then returned to the CCB for approval before being placed into production.

4.3.6 Effect of SCR on Project Documentation

The SCR is not approved until any needed change in project documentation has been identified for update.

4.3.7 Review of SCR for Completeness

The CCB reviews each SCR for completeness of required information before approving the change for transfer to the production database.

When project documentation requires an update, a copy of the SCR with the CCB's final stamp of approval is placed in the document revision file. The Project Administrator informs the Software Configuration Manager of the SCRs that had document revisions completed. The Software Configuration Manager or his designee updates the SCR to denote the change.

4.4 Configuration Status Accounting

Status accounting is performed by tracking the changes to software products using the SCR form. The SCR form, NAC 135-PR-305 – Software Change Request Procedure, is updated upon release of each revision of the software product. Status of the CSCI is reported to the Project Manager on request.

Configuration status accounting activities record and report the status of project configuration items. Status accounting is performed at the configuration item level and is reported as required by the ISSM. Each configuration item is tracked by the ISSM for the status of requested changes, including the follow on implementation status of approved changes.

When a new version has been released, the first weekly back-up tape will contain the release and the revision, the components of the system, and their current change level on the transmittal cover.

4.5 Configuration Audits and Reviews

A System Requirements Validation Review (SRVR is held to ensure the adequacy, accuracy and completeness of the SRS). The SRVR is conducted by NAC as a formal presentation allowing all project team members to review and comment on the SRS.

A Test Readiness Review (TRR) is conducted to evaluate preliminary test results for the CSCI to verify that the test procedures for the CSCI are complete, comply with test plans, describe test requirements, and satisfy test requirements.

An FCA is conducted to verify the software satisfies the requirements as outlined in the SRS. Additionally, a PCA is conducted to verify the design and reference documents accurately represent the installed software. The FCA and PCA occur following the successful completion of the System Acceptance Test and with the release of a new version.

4.6 Interface Control

The Project Manager, supported by the Senior Technical Manager, coordinates changes to the Project CSCI resulting from external factors, such as telecommunications interfaces or backup system changes.

4.7 Subcontractor/Vendor Control

Configuration management is responsible for ensuring that all NMMSS sub-contractors and vendors have CM programs that maintain at least the same degree of CM required of the prime contractor. Sub-Contractor software is software that has been specifically designed under contract for NMMSS, and will be treated like internally developed software. Sub-Contractors are required to comply with the NMMSS Upgrade CM requirements as presented in this plan in developing their software. In addition, the COTS software is tested prior to being placed under SCM. All vendor software is placed under configuration control.

5.0 SCM SCHEDULE

The code and test phase is that period of time in the software lifecycle during which the software modules of the NMMSS Upgrade software are tested. All modules are integrated into the CSCI and the system is evaluated to determine whether the system requirements have been met. Upon successful completion of integration testing, the CSCI is placed in the SML and system testing begins. When system acceptance testing is complete, and a new product baseline to include updated documentation as necessary is established, the CSCI is ready to be placed into the operational environment.

The documentation of testing activities is retained. Test procedures, test cases, and test data sets are retained in the Project Library. Classified data produced during testing is kept in the vault. Because of the need to have extensive data for data sets, the historical data in the vault is used for data set purposes.

The operational and maintenance phase is that period of time in the software lifecycle following system acceptance testing. During this time, the NMMSS software is deployed into its operational environment, monitored for satisfactory performance, and modified as necessary to respond to changing system requirements. SCM activities continue during the operations and maintenance phase.

6.0 SCM RESOURCES

The SCM activity resource allocation is included in the Software Project Management Plan, NAC 135-PL-330A. The SCM resources include the tools, techniques, equipment, personnel and training necessary for the implementation of the SCM activities specified in Section 3 of this plan.

7.0 PLAN MAINTENANCE

This section identifies the activities and responsibilities necessary to ensure continued SCM planning during the lifecycle of the project. At a minimum, the SCMP is reviewed annually for revision. In addition, the plan is revised throughout the software lifecycle to reflect any changes to the process employed in the management of the NMMSS configuration.

7.1 Plan Responsibility

The Senior Technical Manager maintains the NMMSS SCMP.

7.2 Plan Updates

The SCMP is reviewed as directed by the Project Manager to determine if updates are needed. The reviewers consist of at least the following:

- Software Configuration Manager
- NMMSS Project Manager
- Vice President, Quality

- Vice President, Energy Information Products
- DOE Project Manager

7.3 Plan Change Evaluation

Any necessary changes to the SCMP are agreed upon using the walk-through process. In this process, walk-through participants ask questions and make comments about possible errors, violations of development standards, and other problems. A new revision of the SCMP is issued, reviewed and approved, as appropriate.

7.4 Plan Change Communication

Copies of the new SCMP are issued to the CCB Standing Members and the contract administration authority. A master copy is maintained in the Project Library and copies are made available to other parties upon request to the Project Administrator.

Appendix A
Version Description Document

CPCI Name:	**Version:**	**Date:**
Physical Media and Documentation:		
SPR/BCR Changes Incorporated:		
Installation Procedures:		
Known Discrepancies/Additional Instructions:		
Project Manager: **Date:**	**QA:** **Date:**	

Appendix B
NMMSS Configuration Control Board Charter

1. Purpose

The purpose of this charter is to establish the NMMSS Configuration Control Board and define its scope of authority, identify the membership, and define their responsibilities.

2. Scope

The NMMSS CCB is established to provide a board charged with the responsibility for ensuring all proposed changes to NMMSS are reviewed to ensure that they are viable and will not adversely impact the operation of the existing system or subsystem. The Software Change Request (SCR) will be the primary change instrument used to propose changes to the NMMSS Software Configuration Items (SWCIs).

CCB responsibilities include, but are not limited to:

- The review and disposition of proposed modifications to current baselines. Approve resources for system maintenance and/or enhancements.
- The review and propose prioritization, assignment of due dates, and disposition of proposed SCRs.
- Ensuring that introduction of the proposed change will not have a negative impact on current operations.
- Determining approved SCR implementation schedules.
- Advise Project Manager on deviations from standards and schedules, on a case-by-case basis.
- Inform and approval from DOE Project Manager of SCRs requiring more than 80 man-hours of software engineer time.

3. Authority

The NMMSS CCB derives its authority from the Office of Plutonium, Uranium and Special Materials Inventory, SO-23. The CCB will review each proposed SCR against its real or perceived impact on existing services and operations. The CCB has final authority over all issues brought before it for review. CCB decisions are documented in the CCB minutes and are implemented within the scope of its authority.

4. Membership and Responsibilities

The CCB consists of the CB Chairperson; the CMB Members; and the CMB Scribe.

4.1. Configuration Management Board Chairperson

CCB Chairman responsibilities include, but are not limited to:

- Acts as the final authority for CCB decision for Software Change Request (SCR) with less than eighty (80) man-hours of programming effort.
- Presides over the CCB meeting.
- Reviews SCR for completeness.
- Designates an alternate in his or her absence.
- Designates a scribe for each CCB meeting.
- Updates the Software Change Request System stamping approved or disapproved on Software Change Request records discussed during the CCB meeting.

Software Project Manager responsibilities include, but are not limited to:

- Assigns priorities, time estimates and assigns SCR to a Software Engineer.
- Designates an alternate in his or her absence.
- Reviews SCR for completeness before CCB Meeting.
- Monitors progress on item approved by the CCB.

Database Administrator responsibilities include, but are not limited to:

- Prepares agenda items for CCB meeting.
- Reviews SCR for completeness before CCB Meeting.
- Monitor progress on item approved by the CCB.
- Recommends priorities, time estimates and the assignment of SCR to a Software Engineer.
- Monitors progress on item approved by the CCB.
- Designates an alternate in his or her absence.

CCB Members responsibilities include, but are not limited to:

- Reviews SCR for completeness.
- Monitors progress on item approved by the CCB.
- Recommends priorities, time estimates and the assignment of SCR to a Software Engineer.
- Designates an alternate in his or her absence.

Scribe responsibilities include, but are not limited to:

- Takes notes during the CCB meeting.
- Issues minutes of the meeting to attendees and absent board members.
- Appointed by the Chairman.

5. Protocols

CCB meetings will be held at the direction of the CCB Chairperson. Emergency meetings can be scheduled by the CCB Chairperson within one week of the requestor's notice. Agendas will be distributed as approved by the CCB Chairperson. Items not contained in the agenda may be discussed, at the discretion of the Chairperson.

The CCB Scribe will record pertinent information, including Members in attendance and decisions/actions associated with each proposed change. CCB minutes from the previous meeting will be distributed within three working days of the meeting. Change packages will be distributed to Members as soon as possible before a scheduled meeting. Action items assigned at CCB meetings will be monitored by the Chairperson to assure timely closure.

5.1 Change Management Process

Proposed SCRs are entered into the electronic Software Change Request system for processing. Upon creation, the Software Change Request system assigns a tracking number to the proposed SCR. An anticipated CCB review date is decided upon and the Database Administrator drafts a proposed agenda. A change package consisting of all SCRs contained in that agenda, and the agenda are distributed to all CCB Members and others, as directed by the Chairperson. Members provide their recommendations, and a decision to approve or disapprove is resolved, with a proposed implementation date scheduled. When all agenda items have been exhausted, the Chairperson closes the meeting. The Scribe distributes minutes within three working days.

CCB MEETING SIGNATURE SHEET

Date _____ **Time** _____

Position[1]	ATTENDEE	JOB TITLE
CP	Name:	
	Signature:	
DBA	Name:	
	Signature:	
SCM	Name:	
	Signature:	
SPM	Name:	
	Signature:	
EU	Name:	
	Signature:	
S	Name:	
	Signature:	
QA	Name:	
	Signature:	
SE	Name:	
	Signature:	
	Name:	
	Signature:	
	Name:	
	Signature:	
	Name:	
	Signature	

[1] CP – Chairperson, DBA – Database Administrator, EU – End-user, NS – Non-Standing Member, S – Security, SE – Software Engineer, SPM – Software Project Manager, QA – Quality Assurance, SCM - Software Configuration Manager

Appendix C
SCR Process Flow Diagram