|  |  |  |
| --- | --- | --- |
| **SMURF.CO** | **LIBRARY MANAGEMENT SYSTEM**  **CONFIGURATION MANAGEMENT PLAN**  **August 2020** | **DOC NO: 135-PL-300A**  **Revision: 1**  **Effective Date:**  **(*submission date of SCMP)*** |
| **REVIEWS AND APPROVALS**  **Prepared By:**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  SHUHAILA BINTI MOHD SHAHRI Date  *Software System Analysts*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ADILAH ADRIANA BINTI ZAINUDDIN Date  *Software System Designer*  **Approved By:**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  NURUL ALIN BINTI KAMARUDIN Date  *Software Project Manager* | | |

SOFTWARE CONFIGURATION MANAGEMENT PLAN CONTENTS

[LIST OF REVISIONS 3](#_TOC_250000)

[1.0 PURPOSE 5](#_bookmark0)

* 1. [REFERENCES 5](#_bookmark0)
  2. [Definitions and Acronyms 5](#_bookmark0)
     1. [Definitions 5](#_bookmark0)
     2. [Acronyms 8](#_bookmark1)
  3. [CONFIGURATION MANAGEMENT ORGANIZATION …9](#_bookmark2)
  4. [Project Organization 9](#_bookmark2)
  5. [Configuration Management Authority 9](#_bookmark2)
  6. [Software Configuration Management (SCM) Roles and Responsibilities 9](#_bookmark2)
  7. [Configuration Control Board (CCB) 10](#_bookmark3)
     1. [CCB Roles and Responsibilities. 10](#_bookmark3)
  8. [Configuration Management Policy and Procedures 13](#_bookmark4)
  9. [Configuration Management Methodology and Tools 13](#_bookmark4)
  10. [SCM ACTIVITIES 13](#_bookmark4)
  11. [Configuration Identification (CI) 13](#_bookmark4)
      1. [Identifying Configuration Items 14](#_bookmark5)
      2. [Naming Configuration Items. 15](#_bookmark6)
  12. [Version Control 16](#_bookmark7)
      1. [Production Change Releases 16](#_bookmark7)
  13. [Configuration Control 18](#_bookmark8)
      1. [Requesting Changes 18](#_bookmark8)
      2. [Evaluating Changes 19](#_bookmark9)
      3. [Approving or Disapproving Changes 19](#_bookmark9)
      4. [Implementing Requests 19](#_bookmark9)
      5. [Testing Plan 20](#_bookmark10)
      6. [Effect of SCR on Project Documentation 20](#_bookmark10)
      7. [Review of SCR for Completeness 20](#_bookmark10)
  14. [Configuration Status Accounting 20](#_bookmark10)
  15. [Configuration Audits and Reviews 21](#_bookmark11)
  16. [Interface Control 21](#_bookmark11)
  17. [Subcontractor/Vendor Control 21](#_bookmark11)

[5.0 SCM SCHEDULE 21](#_bookmark11)

[6.0 SCM RESOURCES 22](#_bookmark12)

CONTENTS (continued)

* 1. [PLAN MAINTENANCE 22](#_bookmark12)
  2. [Plan Responsibility 22](#_bookmark12)
  3. [Plan Updates 22](#_bookmark12)
  4. [Plan Change Evaluation 23](#_bookmark13)
  5. [Plan Change Communication 23](#_bookmark13)

Appendix A - Version Description Document 24

Appendix B - NMMSS Configuration Control Board Charter 25-28

Appendix C - SCR Process Flow Diagram 29

# 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 constructing this Software Configuration Management Plan (SCMP) is to define the implementation and procedures in enabling to accomplish Configuration Management through the lifecycle of the previous system “Library Management System” project. With this document enable to figuring out the policies, procedures and processes used to achieve the goals system Configuration Management (CM) for this project.

# REFERENCES

This document was prepared with referring to the following documents as guidelines and input to be used into SCMP:

1. IEEE Std. 828-1983, IEEE Standard for Software Configuration Management Plans by Software Engineering Technical Committee of the IEEE Computer Society
2. IEEE Std 729-1983, IEEE Standard Glossary of Software Engineering Terminology.
3. NMMSS Project, NMMSS Upgrade Software Configuration Management Plan issued on September 2001.

# 

# Definition and Acronyms

# Definitions

# The technical terms used in this document are defined in IEEE Std. 828-1983 and IEEE Standard Glossary of Software Engineering Terminology.

# Acronyms

CCB Configuration Control Board

CI Configuration Item

CM Configuration Management

IEEE Institute of Electrical and Electronics Engineers

QA Quality Assurance

SCM Software Configuration Management SCMP Software Configuration Management Plan CR Change Request

SDD System Design Description

SRS System Requirement Specification

PM Project Manager

# CONFIGURATION MANAGEMENT ORGANIZATION

# Project Organization

The organizational chart will be displayed in Appendix A that will specify the roles of each members in conducting this project. All of these roles are crucial as each role as their own specified work in developing the project which will be related to all configuration management activities.

# Configuration Management Authority

The authority for the implementation of Library Management System configuration management practices is derived from the previous semester project. The previous system has specified their requirements in SRS documentation which able to be used in practices and procedures enable to ensure of any system change proposal meets its program objectives that would offer a substantial benefit.

# 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 |

# 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 Representative1 |
| NAC Quality Assurance | NAC Quality Assurance Manager |
| 1 Over-sight capacity for SCRs that meet the significant impact criteria. | |

# CCB Roles and Responsibilities.

* + - 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.

# 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.

# 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.

# 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.

# SCM ACTIVITIES

# 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 |

# 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.

# 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 | |

# 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.

# Production Change Releases

* + - 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.

# 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.

# 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. |

# 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.

# 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.

# 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.

# 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.

# 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.

# 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.

# Effect of SCR on Project Documentation

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

# 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.

# Configuration Status Accounting

The status accounting is carried out in order to track the changes made to the system through the Change Request Form. The CR form is constantly update by the team because the Configuration Control Board will review and highlights the necessary or unnecessary feature that could be beneficial to the system. Once all the change request has been finalized and approved by the Project Manager and CCB will need to approve it.

The activities involve in the Configuration status accounting will be record and report

of the status project configuration items. All the recorded items will be updated into the project Github and will be review and discuss of any further issues of the Change request. This allows to track each status of change request including the implementation status of all the approved changes.

Every version will be released and will be store as a back-up in keeping track the constant update of the change request which this version contains the comments, approval from the project manager and CCB as well as status of the change request.

# Configuration Audits and Reviews

Configuration Audits and Review were not carried out for the Library Management System for now as auditing requires large amount of time which will cost more money and acquire much effort from a team group of three to four people to do auditing. Instead of doing configuration audit, the Quality Assurance team will do the test of the content distribution media in making sure that the approved version of the system modules is involved. The team will responsible with the Software Version Control and develop installation checklist in verifying all the hardware and system software approved of the Configuration Item is available. The checklist will be used by the developer team and returned to the system project for any verification and filing.

# Interface Control

The Project Manager, supported by System Software Designer will manage the changes to the system interface which all the changes is acquire from the final version of Change Request which mainly focuses on the design interface of the system and backup system changes.

# Subcontractor/Vendor Control

Configuration management is responsible in ensuring that all the project sub-contractors have the Configuration Management programs in maintaining the same level of Configuration management required of the prime contractor. In this project, there are no sub-contractor acquire as the project is mainly focuses on the management of the library which only require the book products from the library.

# 5.0 SCM SCHEDULE

# For this project, Iterative model is been chosen (Appendix B). This model focuses on the repetition in a cyclic manner of every step after every cycle of process. The key activities of this model are the first development will be construct in very small scale and all the steps are followed which are taken into consideration. Then every next iteration will focus on the features and modules that were designed, coded and tested. Once satisfied with the module, it will be added to the system. This cycle will be repeated for each module so each iteration the management team will go through a risk management to prepare for the next iteration.

# The SCM Schedule is mainly referred to the identification and approval of the Change Request by the Project Manager and Control Change Board. After finalizing the CR, development phase will start accordingly. The initial development will only scale down to parts as testing and verifying of the module is issued just like in the iterative model. So each of the module will be follow by each iteration before applying to the system hence, initial testing made by the team of each of the modules in the project occur. All the modules are integrated into a test case of the system which will be evaluate to determine whether the system requirements have been met. Finally, acceptance testing will be carried out tested to the targeted user. After completing the acceptance testing, a new product baseline will be included in the updated documentation as necessary and the system is ready to be deploy to the intended user.

# All the documentation of testing activities will be saved. Test procedures, test cases and test data sets will be available at Github. This allows for the team to have references for any further update of the system and for data set purposes. After the system has been deploy to the public, maintenance is needed to keep track the system performance and ensuring that the system is always available or necessary changes need to be done in the future. This is where the SCM activities will continue during this phase.

# 6.0 SCM RESOURCES

The Software Configuration Management activity resource allocation is included in the Mini Project Report. The SCM resources include the tools, software, hardware, equipment, personnel and training necessary for the implementation of the SCM activities.

# PLAN MAINTENANCE

In this section will describes the activities and responsibilities necessary in ensuring the continued SCM plan during the lifecycle of the project. The SCMP will be reviewed constantly for revision. Hence, the plan will be revised throughout the software lifecycle to view any changes to the process used in the management of the project configuration.

# Plan Responsibility

The Project Manager and Software System Analysts will maintain the Library Management System Software Configuration Management Plan.

# Plan Updates

The SCMP is reviewed as directed by the Project Manager in determine necessary update are required. The reviewers consist of at least the following:

* Software System Analysts
* Project Manager of this project
* Control Configuration Board
* Library (Stakeholders)
* Test User

# Plan Change Evaluation

Necessary changes to the SCMP are based on the approval of the project manager and CCB which the process consist of a lot of discussion between the stakeholders and the developer. The new revision of the SCMP is issued, reviewed and approved as appropriate.

# Plan Change Communication

The copy of the new SCMP, Minutes of Meeting, Change Request Form are issued to the project team and CCB. A master copy is maintained in the Github and copies is made available the team for them to review or take as references.

# Appendix A

# Project Organization Chart

# Appendix B

# Software Development Life Cycle (Iterative Model)

****