

**UNIVERSITY OF KUALA LUMPUR**

**MALAYSIAN INSTITUTE OF INFORMATION TECHNOLOGY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of Course** | Software Configuration Management (SCM) | **Course Code** | ISB42303 |
| **Lecturer** | | Ms. Nurul Sharaz Azmanuddin | |

|  |  |  |
| --- | --- | --- |
| **ID Number** | **Student Name** | |
| 52213116280 | MUHAMAD FAIZ BIN HISHAMMUDDIN | |
| 52213116286 | AFIQ IZZUDDIN BIN MOHAMAD BUKHAREE | |
| 52213116044 | SYED ZARUL ZAQUAN BIN SYED MUHAMAD ZULKIFLI | |
| 52213116292 | WAN MOHAMAD NAIMAN BIN WAN MAYU OTHMAN | |
| **Program** | | BSE |
| **Submission Date** | | 24 May 2019 |

|  |  |
| --- | --- |
| **Assessment** | SCM Plan |

Table of Contents

**1.0 IDENTIFICATION1**

1.1 DOCUMENT OVERVIEW2

1.2 ABBREVIATION AND GLOSSARY3

1.3 CONVENTIONS3

**2.0 ORGANIZATION4**

2.1 ACTIVITIES AND RESPONSIBILITIES5

**3.0 CONFIGURATION IDENTIFICATION4**

3.1 IDENTIFICATION RULES5

3.2 CONFIGURATION BASELINE MANAGEMENT5

**4.0 CONFIGURATION CONTROL4**

4.1 CHANGE MANAGEMENT5

**5.0 CONFIGURATION SUPPORT ACTIVITIES4**

5.1 CONFIGURATION STATUS ACCOUNTING5

5.2 TOOLS5

5.3 REVIEWS5

5.4 CONFIGURATION MANAGEMENT PLAN MAINTENANCE5

1. **IDENTIFICATION**

This Configuration Management Plan (CMP) is a tool used to establish the overall approach for the Configuration Management requirement for the Llama Plug system. The scope of this plan extends to Software Configuration Items (SWCIs) developed or implemented for the system’s life cycle. The CMP will be a dynamic document, and will be updated as work on the Llama Plug system proceeds and the necessity arises. This document shall be maintained by the project manager of this project, Faiz.

* 1. **DOCUMENT OVERVIEWS**

The Llama Plug project scope of work requires the development of a Configuration Management Plan (CMP). The CMP will describe the overall technical and administrative direction and control for the Llama Plug system during the total system life cycle, including operational phases. The following provides a summary of each section contained within this Configuration Management Plan.

• Section Introduction identifies, describes the purpose, introduces the objectives, and summarizes the contents of the document.

• Section Referenced Documents lists the referenced materials to which this document refers for further information.

• Section Configuration Management outlines and describes the project’s Configuration Management practices, which are based on the generally accepted industry standards in manufacturing and software development.

• Section Configuration Identification identifies the Baselines and the Configuration Items (CIs) comprising the Baselines.

• Section Configuration Control defines the policies and methods used to establish and control the Configuration Items identified in section 4 by the formally constituted Configuration Control Board (CCB).

• Section Configuration Management Status Accounting describes the plans for status accounting, which track changes and document them for historical reference.

The application of this CMP shall be as follows:

• Initial implementation and changes to the Llama Plug system (comprised of software assets and communications system) will be implemented in a controlled manner via a CCB.

• The aim is to keep the number of configuration changes to a minimum, and in this context, it must be kept in mind that technical or engineering feasibility are not in themselves sufficient grounds for change authorization.

The CMP is a living document and as a result additions, deletions, and modifications will occur as it is utilized. It will be updated as additional configuration activities are defined as the work proceeds and the necessity arises. Changes to this document after acceptance of the initial version require approval of the CCB.

* 1. **ABBREVIATION AND GLOSSARY**

**Baseline.** A Baseline is a Configuration Identification term formally designated and applicable at a specific point in an items life cycle. A version of a piece of software, which has been benchmarked, is re-creatable, and controlled in a configuration management system from which all changes are measured against. Baselines, plus approved changes from those baselines, constitute the current configuration identification.

**CMP.** Configuration Management Plan

**Configuration.** The functional and/or physical characteristics of hardware/software as set forth in technical documentation and achieved in a product.

**Configuration control.** The systematic evaluation, coordination, approval/disapproval, tracking, and dissemination of proposed changes and implementation of all approved changes in the configuration of any item after formal establishment of its configuration Baseline.

**Configuration Control Board (CCB).** A board composed of technical and administrative representatives who review and authorize changes to an approved baseline.

**Configuration identification.** The current approved or conditionally approved technical documentation for a configuration item as set forth in specifications, drawings, and associated lists, and documents referenced therein.

**Configuration item (CI).** A configuration item is an aggregation of hardware, software, or communications equipment, that satisfies a discrete end use function and is designated for separate configuration management.

**Hardware/Communications configuration item (HCCI).** A specific CI which is an aggregation of hardware and communication items that satisfies an end use function and is designated for separate configuration management. This aggregation includes documentation and other artefacts associated with the HCCI that are essential to the operation and maintenance of the HCCI.

**Life cycle.** The phases a software product goes through between when it is conceived and when it is no longer available for use. The software life-cycle typically includes the following: requirements analysis, design, development, testing (validation), installation, operation, maintenance, and retirement.

**Problem Report (PR).** A report to the CCB detailing problems or errors encountered in the Llama Plug system.

**System Change Request (SCR**). A proposed modification to the baseline of the Llama Plug system (SWCI(s)) that is to be managed by the Change Control process.

**Software configuration item (SWCI).** A specific type of CI that relates to a software item that is identified for configuration management. SWCI may be documentation, data, or other artefacts that are created or procured as part of the software development lifecycle.

**Status accounting.** The process of documenting and tracking the correct approved actions/status of the system, including a historical record of the development of CIs and all approved changes.

1. **ORGANIZATION**
   1. **ACTIVITIES AND RESPONSIBILITIES**

Configuration management is a systems development discipline that promotes the proper identification of the configuration, control of changes, and records the change implementation status of the physical and functional characteristics of the Llama Plug.

Configuration Management identifies what is required, designed, and produced. It also provides for the evaluation of changes including effects on technical and operational performance. This leads to making the configuration visible and understood by all the parties involved with the project.

Configuration management covers three basic essential interdependent activities:

1. Error! Reference source not found. – Configuration identification is for the formal step of identifying the configuration of an item (i.e., name, location, version), and documenting its’ functional and physical characteristics.

2. Error! Reference source not found. – Configuration control is the exercising of established procedures to classify, approve or disapprove, implement, and confirm changes to the agreed upon specifications and baselines.

3. Error! Reference source not found. – Configuration accounting is the formal recording and reporting of data relating to configuration identification, approval status of proposed changes, and implementation status of approved changes during all phases of the project. These three activities will be described in more detail in the following sections.

For this project, each team members have their own responsibilities and position. There are 4 team members working on this project which are the project manager, configuration manager, developer, and tester. Below are the details of the activities and responsibilities of each position.

**Project manager.** Manage on the entirety of the project in the sense of making sure the project is working smoothly, everything is according to schedule, works are done efficiently, and managing every team member.

**Configuration manager.** Configuration Managers maintain information about configuration items required to deliver an IT service, including their relationships. This information is managed throughout the lifecycle of the configuration item. Configuration management is part of an overall service asset and configuration management process.

**Developer.** Develop the system of product that is asked from the client according to their requirements. He also needs to do some changes to the system if there if any change request that has been approved.

**Tester.** Test the final product whether it meets the requirements agreed between the client and the project team.

Basically, all activities that revolve around the project will be documented. There are also many documents that will be recorded for references and presented to the clients.

1. **CONFIGURATION IDENTIFICATION**
   1. **IDENTIFICATION RULES**

Configuration identification consists of setting and maintaining baselines of each individual Configuration Item (SWCI) that define the Llama Plug at any point in time. Depending on the system life cycle phase, different baselines are progressively established. Details of each baseline established throughout the system life cycle shall be maintained.

* 1. **BASELINE MANAGEMENT**

The objective of establishing a baseline is to define a basis for further system life cycle process activity and allow reference to, control of, and traceability among configuration items and to requirements. It serves as the common reference that all system development activity is built on and dictates to the development team the changes that are to be implemented.

• Baselines shall be established for the configuration items. Developmental baselines will be established to aid in controlling the software development life cycle processes.

• A Production baseline shall be established upon implementation of the first phase of the Llama Plug. Further changes to the Production baseline require review and approval by the CCB (Leader).

Baselines are established in a system development effort to define a formal departure point for controlling future changes that affect performance or design. A baseline, once defined and approved, is placed under CM, after which any changes in the baseline should be formally documented and approved. Each package build should have a unique release label. Product baselines should be reviewed and approved with an approval memo and attachments for the description of any discrepancies that are part of the release.

The following items should go into the baseline

• All related requirement documents

• All related design documents

• All related test plans and test plan results

• All non-proprietary development code required to build the system components

• All related data and configuration files

1. **CONFIGURATION CONTROL**

Configuration control covers the evaluation of all Change Requests and Problem Reports and their subsequent approval or disapproval. This includes providing methods and procedures for the systematic proposal, justification, evaluation, coordination, and approval or disapproval of proposed changes to the Llama Plug.

The following outlines the method to avoid the possibility of a change being implemented without due consideration of its effect on the baselines, including logistics impact, costs, schedules, performance, or interface.

To enable the configuration control process to operate correctly and effectively, it is necessary for the CCB to oversee changes having the purpose of:

* Providing the relevant information for best decisions on changes to be made;
* Determining and implementing decisions;
* Reviewing and controlling changes in accordance with policy established by the CCB.
  1. **CHANGE MANAGEMENT**

The configuration control process provides for an orderly incorporation and documentation of approved changes to the formal configuration baseline. Changes can originate as enhancements to existing functionality, hardware problem reports, software problem reports, or notifications of necessary hardware or software upgrades and/or patches that may impact the Llama Plug. Note that there are three types of change requests that may be submitted. These are outlined within this section and example forms are detailed in. One form will be used to capture the change requests and problem reports. The three types of reports are defined below:

• Software Change Request (SCR)

This will document the nature and functional requirements addressed by a proposed change to the software or Interface Control Documents. The process utilized to review, deny, or approve the change is outlined by the form structure and the basic procedure outlined below.

• Problem Report (PR)

o Problem reports will be the basic mechanism for centers to report data or functionality problems;

• Maintenance Change Request (MCR)

o Regular and periodic upgrades to COTS products needed to keep the system operational.

1. **CONFIGURATION SUPPORT ACTIVITIES**
   1. **CONFIGURATION STATUS ACCOUNTING**

Configuration management status accounting provides the necessary reporting mechanism to ensure the integrity of the Llama Plug configuration at any time. With proper configuration management status accounting, the current and previous configurations of the Llama Plug can be reported to the CCB and managed appropriately.

* 1. **TOOLS**

Configuration management status accounting for source code SWCI(s) developed by IBI shall be stored in a version management (VM) repository. VM tools such as CVS, PVCS, Telelogic CM Synergy, or Visual Source Safe are examples of a VM repository. A VM tool such as these allow for managing and controlling access to multiple versions of an object or file such that each is identified by a unique SWCI identifier which includes version numbers and disallows or inhibits changes to the same SWCI version by more than one person or process. VM tools are listed here as examples of tools that can be used by organizations that develop Llama Plug and which groups are required to utilize a VM tool within the development lifecycle.

* 1. **CONFIGURATION AUDITS**

Unless specifically requested by the MTC or CCB, configuration audits are not intended to be part of this plan. If desired, appropriate staff may be contracted to perform this function.

* 1. **CONFIGURATION REPORTING**

Through use of version management tools, software development vendors or Centers developing their own software will be required to produce reports of the content of any pending, current, or prior version of the baseline. Baseline reports may be made available via the collaboration tool at the behest of MTC or the CCB.

* 1. **REVIEWS**

During the project, there will be a review to establish the baseline of the project thus will make the project runs smoothly. The review will be conducted after each and every phase of the software development to ensure that nothing being left behind.

Configuration Manager must prepare an agenda for the reviews to be discuss among the team. This is to ensure that the project can run smoothly as possible. This is also important to detect the program’s drawback during the development of the project.

* 1. **CONFIGURATION MANAGEMENT PLAN MAINTENANCE**

A weekly maintenance will be conducted to ensure the Llama Plug can operate smoothly without any bugs or defect. This is also needed to be done to clear out cache and to debug the program to ensure that there is no anomaly occurring during usage of program. This is also to prevent and keep the program up to date to the virus and malware database. This is because of the program are web-based thus important to ensure the user’s information does not leak out by malicious malware.