SCM BLOG SCM Plan

VERSION 1.0

6 OCTOBER 2014

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Doc. Version | Description | Author |
| 4 October 2016 | 0.1 | First draft | Juniada Bendesa |
| 6 October 2016 | 1.0 | Baselined | Juniada Bendesa |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1. Introduction 4](#_Toc463366571)

[1.1 Organization and Responsibilities 4](#_Toc463366572)

[1.1.1 Configuration Manager 4](#_Toc463366573)

[1.1.2 Change Control Board 4](#_Toc463366574)

[1.1.3 Change Approver 4](#_Toc463366575)

[1.2 Tools, Environment, and Infrastructure 4](#_Toc463366576)

[1.2.1 Tool 4](#_Toc463366577)

[1.2.2 Environment 5](#_Toc463366578)

[1.2.3 Infrastructure 5](#_Toc463366579)

[2. The Configuration Management System 5](#_Toc463366580)

[2.1 Configuration Identification 5](#_Toc463366581)

[2.1.1 Naming and Versioning Standard 5](#_Toc463366582)

[2.1.2 CIs Items under CM 6](#_Toc463366583)

[2.1.3 Access Control 6](#_Toc463366584)

[2.1.4 Repository Structure 6](#_Toc463366585)

[2.1.5 Non CIs 7](#_Toc463366586)

[2.2 Configuration and Change Control 7](#_Toc463366587)

[2.2.1 Change Request Processing and Approval 7](#_Toc463366588)

[2.2.2 Change Control Board (CCB) 8](#_Toc463366589)

[2.3 Configuration Status Accounting 8](#_Toc463366590)

[2.3.1 Management of Configuration Status 8](#_Toc463366591)

[2.3.2 Configuration Audits 9](#_Toc463366592)

[2.4 Configuration Management Process Flow 9](#_Toc463366593)

[2.4.1 New Document Creation Process Flow 9](#_Toc463366594)

[2.4.2 Document Update Process Flow 10](#_Toc463366595)

[2.4.3 Document Review and Release Baseline Process Flow 11](#_Toc463366596)

SCM bLOG SCM Plan

# Introduction

This document details the Change and Configuration Management for SCM Blog project. The focus of this document is to clearly communicate and explain the criteria for implementing effective and efficient change and configuration management in an environment that supports shared documentation, code and executable components across the project.

## Organization and Responsibilities

### Configuration Manager

The Configuration Manager is the team on of the project. Since this is a collaborative works, the team itself will be responsible as Configuration Manager and support the overall CM environment so that the team has an appropriate environment to build and test their work.

Configuration Manager identifies, monitors, and controls CIs in this project. This includes review documents, releasing baseline, manage configuration status, and more importantly promote to change any CIs if something is changed in the project. Configuration Manager must review the document before releasing it to baseline.

### Change Control Board

The Change Control Board consists of representatives from all interested parties. The CCB provides a forum to review Change Requests on a regular basis. The responsibilities of CCB are:

* Assess and prioritize Change Requests
* Approve or secure approvals for Change Requests

This role is assigned to the team member.

### Change Approver

The Change Approver reviews and approves Change Requests with the following responsibilities:

* Assess the benefit of the change in relation to cost
* Assess the business risk and impact of change
* Ensure that the technical feasibility, risk and effect of the change have been adequately assessed
* Approve or reject Change Requests

This role is assigned to the team member.

## Tools, Environment, and Infrastructure

### Tool

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Tool | Version | Function |
| 1 | Git Bash | 2.10.1 | Git Client for Version control system for project development |
| 2 | Apache Web Server | 2.4 | Web Server |
| 3 | MySQL | 5.6 | Relational database management system |
| 4 | Git Hub |  | Git Server for Version control system for project development |

### Environment

* Windows 10 in Mitrais machine/host
* GitHub for repository for archiving of document and project component
* cpanel.idhostinger.com as Application Server

Both Mitrais machine and GitHub are connected through internet common connection.

### Infrastructure

A Configuration Item should be archived if there is new version of it. Move the item from Baseline directory to Archive directory. See [2.1.4 Repository Structure](#_Repository_Structure) for directory structure.

Everyone in this project should be able to get and see the repository. However, changing one should be discussed within Configuration Manager first.



The repository for developer A is shown in a development state. Within Developer A’s repository there are many components, source code, etc., and for example there is one component, shown as Component 1, that has been identified for project level reuse. The notion of project level reuse for Component 1 is that Component 1 can be used across the developers.

Component 1 is going to be copied into the Shared Repository, and thus the directional flow is from the Developer A repository to the Shared Repository. An important qualification for copying Component 1 into the Shared Repository is that Component 1 is in a state that precludes change.

If Component 1 is fully, then the component would be ready for use by other developers as well.

# The Configuration Management System

## Configuration Identification

### Naming and Versioning Standard

Naming and versioning standard for project documents is following this rule:

[ProjectName]\_[DocumentName]\_v[MajorVersion].[MinorVersion](\_review)

The “*\_review”* part is given if the document is being or has been reviewed. For example:

SCMBlog\_SCM Plan\_v0.1.docx  
SCMBlog \_SCM Plan\_v0.1\_review.docx

Major version is increased when large changes is applied to a document. Usually it happens when releasing First Draft documents to Baseline, changing document templates, or something else that has large impact.

Minor version on other hand is increased when small changes is applied to a document.

### CIs Items under CM

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Configuration Item | Storage Path | Approval |
| 1 | Software Configuration Management Plan | [01 Inception](https://github.com/edibwi/scmtraining/tree/master/01%20Inception/01%20Baseline) | Configuration Manager |
| 2 | Software Requirement Specification | [02 Elaboration](https://github.com/edibwi/scmtraining/tree/master/02%20Elaboration/01%20Baseline) | Configuration Manager |
| 3 | Source Code | [03 Construction](https://github.com/edibwi/scmtraining/tree/master/03%20Construction/01%20Baseline) | Configuration Manager |
| 4 | Change Request List | [03 Construction](https://github.com/edibwi/scmtraining/tree/master/03%20Construction/01%20Baseline) | Change Control Board |

### Access Control

All team members have privilege to read every CIs. They actually have privilege to edit/create a CI as well. However, they must get an approval from CM Manager first before they edit/create the CI.

Though at the end, it is CM Manager responsibility to release the CI to baseline.

### Repository Structure

The repository is stored in this below hyperlink.

|  |
| --- |
| https://github.com/edibwi/scmtraining |

The structure inside is like this below

1. 01 Inception

Contains all things related with initial planning for the project including this document.

1. 02 Elaboration

Contains Software Requirement Specification and project estimation. It also contains configuration item list.

1. 03 Construction

Contains Source Code and Change Request List during the project.

1. 04 Transition

Contains any documentation related with product release.

Every directory above has 01 Baseline, 02 Working, and 03 Archive directory.

* 01 Baseline contains things that currently being used throughout the project
* 02 Working contains things that currently being worked on
* 03 Archive contains archive things that do not valid anymore

### Non CIs

Non CIs are but not limited to

* CI list document
* Tools and installers
* Configuration files

## Configuration and Change Control

Change control is the process by which changes to system components are managed. It begins after CIs are baselined and refers to the evaluation, coordination, approval or disapproval, and implementation of changes of those CIs. The goal of change control is to establish a mechanism that will help ensure the production and maintenance of quality system components. The list of all pending requests as recorded in the CM tool is processed and action is taken.

### Change Request Processing and Approval

Project change control practices will be established in the end of the Construction Phase. The purpose of having a standard, documented change control process is to ensure that changes are made within a project in a consistent manner, and the appropriate stakeholders are informed of the state of the product, changes to it and the cost and schedule impact of these changes.

The following state-chart diagram illustrates the change control process and depicts the state of a change request.



A change request is created by user representative and reviewed by Change Control Board. After technical analysis was completed, the change request will be submitted to the Change Control Board (CCB) for review. There are four basic courses of actions that CCB can take. The CCB can reject the change request, in which case the change request is closed. The CCB can postpone the change request to later in the development cycle, and the CCB can accept the CR. If a CR is postponed, it can be resubmitted at a later date for CCB review. The CCB may also request more information to be added to the CR. In this situation, the CR is appended with the additional information and resubmitted.

When a CR is accepted, the CR is assigned to a developer to make changes. When the changes are complete, the changes are verified. If the verification tests pass, the CR is closed. Otherwise, the process of development will be continued until the verification tests pass.

### Change Control Board (CCB)

Threshold is applied to each different cases and size of change request as follow:

|  |  |  |
| --- | --- | --- |
| Business Rule / Logic | Modification to an approved business rule or business flow | CCB |
| Database Change | A change to schema, tables, columns, indexes, constraints, or security | CCB |
| Hardware Change | Modification to existing hardware configuration in the production environment | CCB |
| New Feature | Development of new application functionality | CCB |
| Plan content / Definition | A change to static plan content displayed to user | CCB |
| Requirements Change | Modification to an existing approved requirement | CCB |
| Design Update | Indicates the need for cleanup of dead code or performance enhancements | CCB |
| Small Technical Change | Effort estimation to implement the change is less than 1 days | CCB |

## Configuration Status Accounting

### Management of Configuration Status

Status of CIs are maintained in CI list document.

|  |
| --- |
| 01 Inception\01 Baseline\SCMBlog\_CI\_List.xlsx |

The CM status

|  |  |
| --- | --- |
| CM Status | Description |
| OBSOLETE | The document version is obsolete. Updated by anyone. Once the status is updated to this, store the document version in Archive directory. |
| CURRENT | The document version is being used throughout the project. Updated by CM Manager. Once the status is updated to this, store the document version in Baseline directory. |
| UNDERCHANGED | The document version is being worked on by someone. Updated by anyone. Once the status is updated to this, stored the document version in Working directory. |
| UNDERREVIEW | The document version is being reviewed by CM Manager. Updated by CM Manager. Once the status is updated to this, stored the document version in Working directory. |
| REVIEWED | The document version has been reviewed by CM Manager. Updated by CM Manager. Once the status is updated to this, stored in Working directory. |

CI list document must always be updated before starting to work on a new document, modifying old document, to review a Working document.

A new version of a document is only created when these status change happened.

* No status to UNDERCHANGED
* CURRENT to UNDERCHANGED
* REVIEWED to UNDERCHANGED
* REVIEWED to CURRENT

Mainly if a document is new or modified, it is given a new version to it.

The following diagram outlines the procedure to baseline a new CI. Every CI record as explained in section is stored in the baselined repository by the Configuration Manager. The Configuration Manager needs the required modification privileges to perform this operation.



### Configuration Audits

CM audit is part of Mitrais Project SQA Audit.

## Configuration Management Process Flow

### New Document Creation Process Flow

These are the step on how to create a new document.

1. Pick a document template to use

Pick one at *home mitrais --> Forms & Stationery Templates*. Usually we will use *Stationary templates --> Mitrais Standard SWD Documentation Template*.

Please be aware that those templates has extension *\*.dotx*. So please save it to a *\*.docx* document first before continue to the next step.

1. Place the document into respective project folders/directories

See [2.1.4 section](#_Repository_Structure) about the list of project directories. Put the document to the right place, and since the document is being worked on, place it in Working folder. Always working a document on its respective Working folder, see Section [2.1.4 section](#_Repository_Structure).

1. Give name to the document

See [2.1.1 section](#_Naming_and_Versioning) about document naming convention.

1. Update CI List

See [2.3.1 section](#_Management_of_Configuration) about CI List. Create a new record there about the document.

*Note: please do not forget to update the hyperlink of the Path document. When people clicking the link, it is expected to open the document.*

*Another Note: please do not forget to update the* ***Last updated*** *time and* ***by*** *at the top.*

1. Open the document, and then turn Track Changes on

Go to *Ribbon --> Review --> Track Changes*. This is to ease the reviewer to review the document later.

*Note: Track Changes settings is as per document, not per Office Word application. So please make sure it is turned on before making any changes.*

1. Start working on the document
2. Once done, notify CM Manager for review

### Document Update Process Flow

These are the step on how to update a document.

1. Find out the latest version of the document in CI List

Please find **the latest version**, not the CURRENT one. Someone might already has started updating the document. See [2.3.1 section](#_Management_of_Configuration) about CI List.

*For example: the latest version is SCMBlog\_SCM Plan\_v1.0.docx in Baseline, status CURRENT*

1. Create a copy of the latest document in the Working folder, and then increase the version

Always create a copy of a document if the version is increased.

*For example: so now there are two documents, SCMBlog\_SCM Plan\_v1.0.docx in Baseline and SCMBlog\_SCM Plan\_v1.1.docx in Working*

1. If the old document is not needed anymore move it to Archive folder

A document is still needed if it is still in Baseline. Only CM Manager release a document to Baseline. So if the old document is a Baseline document, **do not** **move** it to Archive.

It is different, however, if the old document is from Working folder: **Move** the old document to Archive.

*For example: SCMBlog\_SCM Plan\_v1.0.docx is in Baseline, so do not move it to Archive.*

*Another example: SCMBlog\_XXX\_v0.1.docx is in Working folder. Somehow the previous author cannot continue the documenting work, so they passed it to someone else. Thus the new version now is SCMBlog\_XXX\_v0.2.docx and so the old document SCMBlog\_XXX\_v0.1.docx is moved to Archive folder.*

1. Update CI List according to current status

Change status or add a new record depends on what has been done. See [2.3.1 section](#_Management_of_Configuration) about CI List. In short, adding new record is done if there is new version of the document.

*Note: please do not forget to update the hyperlink of the Path document. When people clicking the link, it is expected to open the document.*

*Another Note: please do not forget to update the* ***Last updated*** *time and* ***by*** *at the top.*

1. Open the document, and check Track Changes turned on

Please make sure Track Changes turned on. Go to *Ribbon --> Review --> Track Changes*. This is to ease the reviewer to review the document later.

*Note: Track Changes settings is as per document, not per Office Word application. So please make sure it is turned on before making any changes.*

1. Start working on the document

Better to fix Review defect first if any, then adding another things.

1. Once done, notify CM Manager for review

### Document Review and Release Baseline Process Flow

These are the steps on how to review a document.

1. Create a copy of the document in Working folder, add the document name with “\_review”

See [2.1.1 section](#_Naming_and_Versioning).

1. Move the old document to Archive folder
2. Update CI List according to current status

Change status or add a new record depends on what has been done. See [2.3.1 section](#_Management_of_Configuration) about CI List. In short, adding new record is done if there is new version of the document. Reviewing process always adding a new record in CI List document.

*Note: please do not forget to update the hyperlink of the Path document. When people clicking the link, it is expected to open the document.*

*Another Note: please do not forget to update the* ***Last updated*** *time and* ***by*** *at the top.*

1. Start reviewing the document

If using Office 2013, use Comment feature to review the document. Add comments to the part that does not seems right. When that part has fixed, mark the comment as done.

To comment: simply right click the part --> New Comment

To mark done: simply right click the comment --> Mark Comment Done

1. Finished Review or Releasing Baseline

It depends on the result of the document review:

* Release Baseline if it is all good
* Finished Review and schedule to Document Update if defect found.

1. Update CI List again

Whether the document is released to Baseline or scheduled to be updated due to defect found.