NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

QMePls

Software Configuration Management Plan

Date: 10 October 2021

Seow Jing Hng Aloysius
Jacob Law Zhen
Jolene Tan
Soh Qian Yi
Samuel Lee Si En
Zeta Chua Hui Shi

Team Titans
School of Computer Science and Engineering

Revision History

Name	Date	Changes	Version
Qian Yi	10 October 2021	Initial document draft, References and Organization section	1.0
Qian Yi	11 October 2021	Document overview and abbreviations	1.1
Jolene	12 October 2021	Identification rules, reviews, and configuration management plan maintenance	1.2
Qian Yi	13 October 2021	Configuration identification, configuration control and configuration support activities.	1.3
Jolene	15 October 2021	Conventions and Interface Management	1.4
Qian Yi	20 October 2021	Edited relevant sections and formatted document	1.5

Table of Contents

1. Identification

This document amplifies the “Configuration Management” of the Project Management Plan.

1.1 Document Overview

This document provides a standard outline and format of the Configuration Management for ‘QMePls’.

1.2 Abbreviations and Glossary

This section summarises the abbreviations and glossary used in this document.

1.2.1 Abbreviations

Abbreviation	Definition
SCM	Software Configuration Manager
SCMP	Software Configuration Management Plan
SCI	Software Configuration Items
PM	Project Manager
LD	Lead Developer
QAM	Quality Assurance Manager
QA	Quality Assurance
RM	Release Manager
API	Application Programming Interface
VDD	Version Delivery Description

1.2.2 Glossary

Word	Description
Baseline	A baseline establishes an approved standard where changes to the SCI can only be made under a formal change control procedure
Branch	A branch is a copy of codeline, managed in a version control system (VCS) and can be merged in the future
Version	State of a configuration item at a well-defined point in time

1.3 References

This section summarises the project, standard and regulatory references made, and the convention used in this document.

1.3.1 Project References

#	Document Identifier	Document Title
1	D01	Use Case Description
2	D02	System Requirement Specification
3	D03	Quality Plan
4	D04	Project Plan
5	D05	Risk Management
6	D06	Change Management Plan
7	D07	Release Plan
8	D08	Design Report for Software Maintainability
9	D09	Software Configuration Management Plan
10	D10	Test Plan

11	D11	Test Cases and Requirements Test Coverage Report
12	D12	CMMI Level 2 Definition

1.4 Conventions

Conventions	Descriptions
Font size 20, Bold	Document Section header
Font size 16, Bold	Document Subsection header

2. Organization

Specific tools are utilised by members of Team Titans to manage software configuration. Responsibilities are shared between:

- Project Manager (PM) - Jolene Tan
- Release Manager (RM) - Soh Qian Yi
- Quality Assurance Manager (QAM) - Aloysius Seow
- Lead Developer (LD) - Jacob Law

2.1 Activities and responsibilities

The following section describes the functions required to manage the configuration of the software and responsibilities.

Activities when setting up the project	**Person responsible**
Identify the configuration items	PM
Install the bug repository tool and set up the database	LD
Install the software configuration repository tool and set up the database	LD
Manage and structure the reference space	RM
Define the configuration processes	QAM

Activities during the project lifecycle	**Person responsible**
Export components for modification, test or delivery	QAM
Set under control validated components	QAM
Create version, write version delivery document	RM
Approve reference configurations	PM
Verify version to be delivered and authorize deliveries	PM
Backup spaces	LD
Do configuration audits	QAM
Inspect configuration records	QAM
Archive reference version	RM

Management activities	Person responsible
Manage versions and archives	RM
Manage configuration records	RM
Produce reports and statistics	PM
Manage reference space and its access control list	LD
Manage spaces backup and archive media	LD
Manage quality reports	QAM

2.2 Decision Process & Responsibilities

At the end of an activity of the project:

Activities	Person Responsible
Do a configuration freeze	PM
Present configuration state of the components impacted by activity	RM
Present documentation state of the components impacted by activity	QAM

During a configuration management process audit:

Activities	Person Responsible
Do the configuration management process audit	PM
Present the records of the configuration management process	RM
Present the quality records of the configuration management process	QAM
Present the records of the documentation management process	RM

3. Configuration Identification

3.1 Identification Rules

Stated below are the identification rules which must be adhered to minimize confusion during the development of the QMePls project.

3.1.1 Identification Rules of Configuration Items

Identification of a Configuration Item

The naming convention format for identifying configuration items will be used:

Format: <Project_name>_<Version_number>.<File_type>

Example: modules_V1.0.go

Version Number of a Configuration Item

Version numbers must be attributed prior to the delivery of configuration items. If the product or documentation is/are modified, this number should be incremented before a new delivery. The defining rules for version numbering are as follows:

The version number will use 2 numbers separated by a period (.), which represents 2 types of software iterations.

Format: V<Major_number>.<Minor_number>
Example: V1.2

Where:
<Major_number> is incremented when a major update is performed on products or documents e.g., adding of new functions/features.
<Minor_number> is incremented when a minor modification or bug fix is performed on products or documented e.g., editing a subsection of a function.

3.1.2 Identification Rules of Documents

Description of Documents Identifiers

The identification of documents is as follows:

Format: <Document_ID>_<Name>_<Version>.<File_Type>

Example: D01_Use_Case_Description_V1.1.docx

- Where the following defines <Document_ID> and <Name> is defined as follows:

<Document_ID>	<Name>
D01	Use Case Description
D02	System Requirement Specification
D03	Quality Plan
D04	Project Plan
D05	Risk Management
D06	Change Management Plan
D07	Release Plan
D08	Design Report for Software Maintainability
D09	Software Configuration Management Plan
D10	Test Plan
D11	Test Cases and Requirements Test Coverage
D12	CMMI Level 2 Definition

3.2 Configuration Baseline Management

The purpose of this section is to describe the baselines established and how they will be defined and controlled.

The following baselines are to be established:

Baseline	**Description**	**Person In-Charge**
Functional Baseline (FBL)	Approved technical documentations describing the functional performance characteristics and requirements for QMePls' SCI	PM
Allocated Baseline (ABL)	Approved specifications describing design of the functional and interface characteristics for QMePls's SCI	QAM
Product Baseline (PBL)	Approved documentation for defining a SCI during production, operations and maintenance phase of the system life cycle. describing completed and accepted system components, as well as documentations that server as these products' identifier	SCM

The initial baseline is reviewed, established and approved by the PM and SCM, while QAM approves the standard. Subsequent changes can only be made through a formal change request process. A prototype has to be tested at least twice and can only be established after a formal technical review of SCIs have been conducted.

4. Configuration Control

The PM must approve all SCIs before any modification. After approval, the version numbers of the modified items will be incremented according to the type of changes made which will be uploaded to the working folder and backup folder. The backup folder is used primarily for reversion for files with unfixable issues.

4.1 Change Management

Problem resolution and multiple configuration both utilize the same process of controlling changes to the baselines as well as tracking the implementation of the changes. Both processes will be separated using directories.

The following is the change management measures for problem resolution and multiple configuration:

- Changes requests will be submitted through a Change Request Form, which would be submitted for evaluation to PM
- Developers evaluates the cost, time and effect of the change on development
- QA team determines the effect on quality
- RM determines the significance of change
- PM verifies and approves or reject change
- RM creates branch (if necessary)
- Branch identification is identified via version numbers
- Branch content consists of source code, document and configuration items.
- Branch information must be documented.
- “PR” is the label for problem resolution, “MC” is the label for multiple configurations.

5. Configuration Support Activities

5.1 Configuration Status Accounting

Configuration Status Accounting (CSA) is the process to record, store, maintain and report the status of configuration items during the software lifecycle. All software and related documentation should be tracked throughout the software life.

5.1.1 Evolutions Traceability

The traceability of modifications of items given their types:

- **Document**: The modification sheet number identifies the origin of the modification. The modified paragraphs in the document are identified, if possible, by revision marks.
- **Source File**: A comment which describes the modification is recorded for each source file or group of source files by the software configuration management tool
- **Configuration item**: The VDD of the article identifies the modification sheet included in the current version.

5.1.2 Setting Up Configuration Status

The SCM sets up the state of all versions of each configuration article with the following 3 criteria:

- The label
- The version number
- The creation date of the VDD, where the VDD is written by the SCM.

5.1.3 Configuration Status Diffusion

The SCM and the QM write the VDD.

5.1.4 Configuration Status Record Storage

The records are stored in a configuration folder which contains the following:

- The requests sorted by record number
- The software documents
- The VDDs
- The configuration states sorted in chronological order

5.2 Configuration Audits

Configuration audits are performed to determine if the configuration item accurately reflects the physical and functional characteristics as defined.
The following are the detailed list of audits to conduct:

Configuration Audit	**Description**
Baseline Audit (BA)	Used to establish a reference point. By performing an audit in advance, we can identify improper coding practices before making corrective actions and checking if subsequent SCIs are on par with the baseline.
Functional Configuration Audit (FCA)	Used to verify if each SCI has adequately achieved the performance and functional characteristics as specified in the system documentation. It is conducted by the CM and the FCA team reviews the documentations listed in the FCA checklist to verify that all functional parameters were tested, and if the results were satisfactory. FCA is held at the end of the development lifecycle.
Software Configuration Audit (SCA)	Used to verify if all the software product satisfies the baseline needs. It checks if defined processes are followed and ensures that the SCM goals are satisfied, verifying compliance with configuration control standards. SCA ensures traceability is maintained and all changes made to the baseline complies with the configuration status reports, providing validation of completeness and consistency.

5.3 Reviews

Reviews are performed and created upon completion of testing, to ensure that SCIs have been correctly identified and produced. To guarantee that the system design is sufficient for further design and development, an evaluation of the completeness and sufficiency of the design must be done at important milestones throughout the system's life cycle.

Developers will conduct technical reviews on their assigned branch on a regular basis, and the SCM will compile and categorize them according to testing scenarios. SCM will verify that the reviews and configurations were carried out in accordance with the baseline criteria. If an existing baseline can be enhanced as a result of the review, the SCM and PM will discuss the new baseline or branches. The PM must approve the revised baseline, and the team will be notified of the changes.

5.4 Configuration Management Plan Maintenance

The PM creates and controls the configuration management plan (SCMP), and any changes to the SCMP must be approved by the PM. The QA team is in charge of approved updates and is also in charge of maintaining the SCMP throughout the project. The PM must inform the team of the change following the revision. The PM and QA teams are in charge of evaluating the SCMP on a weekly basis, and modifications must be made to keep the plan updated. The LD is intended to assist the PM in developing the strategy and technical words utilized in SCMP, as well as updating adaptation if necessary.