NANYANG
TECHNOLOGICAL
UNIVERSITY
SINGAPORE

Planit Software Configuration Management Plan

Team Members	Role
Ryan Tan Jinn En	Project Manager
Neo Yong Tai	Development Lead
Lee Yu Sheng Daniel	Back-End Developer
Mamuduri Paulani	Front-End Developer
Frankie Ye Htet Myat	Release Engineer / Manager
Chen Xueyao	QA Manager
Kundu Koushani	QA Engineer

Name of Design Team: **Team Syan**
School of Computer Science and Engineering, Nanyang Technological University

Version History

Version #	Implemented By	Revision Date	Reason for Revision
1.0	Ryan Tan	22/10/2021	First Version

Table of Contents

1 Identification

1.1 Document Overview

This document is the Configuration Management Plan of Planit. This document will define and provide the necessary steps for configuration management for the project.

1.2 Abbreviations and Glossary

1.2.1 Abbreviations

- SCM: Software Configuration Manager
- SVN: Apache Subversion
- SRS: Software Requirements Specifications

1.2.2 Glossary

- Branch: Line of development that exists independently of another line, but still sharing a common history, and can be merged in the future
- Version: State of configuration item at a well-defined point in time
- Variant: Versions that coexist

1.3 Standard and Regulatory References

#	Document Identifier	Document Title
[STD1]	1	IEEE 828-2012 - IEEE Standard for Configuration Management in Systems and Software Engineering

1.4 Conventions

Typeface	Usage	Example
Bold	Emphasis, headers, and titles	**Software Configuration Management**
Highlighted	Emphasis	Key points

Red	Comments and user requests for change to mark changes to the document	I believe this part should be changed with ….
Italic	Minor emphasis, file names.	The file *Notifications* contains...

2 Organization

The software configuration is managed by the members in the team of this project, whereby the responsibilities are shared between:

- The Software Configuration Manager (SCM)
- The Project Manager
- The Quality Manager
- The Lead Developer

2.1 Activities and Responsibilities

Activities When Setting Up the Project	**Person Responsible**
Identify the configuration items	SCM
Install the bug repository tool and set up the database	SCM
Install the software configuration repository tool and set up the database	SCM
Manage and structure the reference space	SCM
Define the configuration processes	SCM

Activities During the Project Lifecycle	**Person Responsible**
Export components for modification, test or delivery	SCM
Set under control validated components	SCM
Create version, write version delivery document	SCM
Approve reference configurations	Project Manager

Verify version to be delivered and authorise deliveries	Project Manager
Backup spaces	SCM
Do configuration audits	Quality Manager
Inspect configuration records	Quality Manager
Archive reference version	SCM

Management Activities	Person Responsible
Manage versions and archives	SCM
Manage configuration records	SCM
Produce reports and statistics	SCM
Manage reference space and its access control list	SCM
Manage spaces backup and archive media	SCM
Manage quality reports	Quality Manager

2.1.1 Decision Process and Responsibilities

At the end of an activity of the project:

Activities	Person Responsible
Do a configuration freeze	SCM
Present a configuration state of the components impacted by the activity	SCM

Present a documentation state of the components impacted by the activity	SCM

During a configuration management process audit:

Activities	**Person Responsible**
Do the configuration management process audit	Project Manager
Present the records of the configuration management process	SCM
Present the quality records of the configuration management process	Quality Manager
Present the records of the documentation management process	SCM

3 Configuration Identification

3.1 Identification Rules

3.1.1 Identification Rules of Configuration Items

3.1.1.1 Identification of a Configuration Item

The identification of the configuration item is **XXX_Va.b**, where "Va.b" is the version of the configuration item.

3.1.1.2 Version Number of a Configuration Item

The attribution of a version number is a prerequisite to any delivery of any configuration item. This number shall be incremented before a new delivery, if the product or its documentation was modified.

The definition rules of a version number are the following:

- Major edits call for a new major version (a)
  - Adding or removing significant sections, functions, or features
  - Redevelopment of document
- Minor edits call for a new subversion (b)
  - Editing significant sections, functions, or features

3.1.2 Identification Rules of Documents

3.1.2.1 Description of Documents Identifiers

The identification of documents is described below: **XXX_<document type>_<document number>_<revision index>**

Where the following refers to:

- "document type" is Foo for FOO documents and BAR for bar documents
- "document number" is an incremental number, with a separate list for each document type
- "revision index" designates the approved iteration of the document, where the revision index is V1 for the first iteration and so on

3.1.2.2 Definition and Evolution of the Revision Index

The attribution of a revision index is a prerequisite to any delivery of a document or file. This index shall be incremented before the diffusion of a modified document. The definition rules of a revision index are similar to the ones in section 3.1.1.2.

3.1.3 Identification Rules of a Media

3.1.3.1 Internal Identification

A media can be identified as either a thumbdrive, hard disk or CD ROM for submission. It will be identified as: **<configuration item identification>/<media>/<volume>**

Where:

- “media” is the media number
- “volume” is an incremental number to distinguish the media if the delivery contains more than one media

3.2 Reference Configuration Identification

Each reference configuration is defined by:

- An identifier
- Its content listed in the corresponding Version Delivery Description document
- The acceptance or validation reviews associated with the building of the reference configuration

A reference configuration is established for each design review and each test review of the project.

3.3 Configuration Baseline Management

The baselines to be established are:

- Functional baseline (FBL) describes the system functional characteristics
- Allocated baseline (ABL) describes the design of the functional and interface characteristics
- Product baseline (PBL) consists of completed and accepted system components and documentation that identifies these products

The baselines will be defined and controlled by:

- Having gained approval from the Project Manager of the team
- The quality assurance team has approved the standards
- The project has been completed or else has at least been tested once

4 Configuration Control

Most features will not have to be changed unless a bug is found or performance is not optimal. Our process will involve ensuring the proper functioning of the application after any change is made.

4.1 Change Management

Changes are bound to occur in complex software projects. This could either be due to changing user demands or due to errors and bugs. Regardless of the change, none of the software components must break. The process for controlling the changes to the baselines and for tracking the implementation of these changes is as follows:

Problem resolution for software errors:

- Change Request Form containing details of Software Errors are submitted by users
- The development team reviews the Change Request Form and evaluates the impact and extent of the modification
- When the Change Request Form is deemed to be valid, the developers will assess the cost, feasibility and priority of the change
- The CCB then reviews the change and decides whether to implement it or not
- If the change is accepted then the change is implemented in a new branch of the system
- Note that the change form is updated at each step documenting comments and proceedings from all parties involved

Example for multiple configuration:

- Change Request Form containing details of Software Errors are submitted by users
- The development team reviews the Change Request Form and evaluates the impact and extent of the modification
- When the Change Request Form is deemed to be valid, the developers will assess the cost, feasibility and priority of the change
- The CCB then reviews the change and decides whether to implement it or not
- If the change is accepted then the change is implemented in a new branch of the system
- Note that the change form is updated at each step documenting comments and proceedings from all parties involved

4.2 Interface Management

This section identifies the interfaces to be managed and describes the procedures for identification of interface requirements.

We will require a third party API in order to get the current and historical temperature and humidity information to measure plant yields. When the plants' growth is below the expected threshold, we will send warning notifications to the users. Every growth measurement is predicted based on the temperature. Thus, if there are major updates to the API, our application has to be updated accordingly, if applicable.

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: Modification sheet number identifies the origin of modification. The modified paragraphs in the document are identified, if possible, by revision marks
- Source file: The software configuration management tool records, for each source file or group of source files, a comment where the modification is described
- Configuration item: The Version Delivery Description of the article identifies the modification sheet included in the current version

The modification sheet describes the modifications done to the components with enough precision to identify the modified parts.

5.1.2 Setting Up Configuration Status

The SCM sets up the state of all versions and of each configuration article with the label, version number, and creation date of VDD.

5.1.3 Configuration Status Diffusion

The SCM and the Quality Assurance Manager write the VDD.

5.1.4 Configuration Status Records Storage

The records are stored in a configuration folder, which contains the requests sorted by record number, the software documents, the VDD's, and configuration states sorted chronologically.

5.2 Configuration Audits

SCM audits help to ensure that all the SCM processes are followed. It also helps in ensuring the integrity of the configuration baselines.

A Peer Review audit is a formal review where a person or group of people examines the product (document, code, or other) in detail, so as to help in evaluating its technical content and product quality.

Formal audits are conducted at predetermined points. This audit is to ensure that the design, development and integration of Plantit meet the technical requirement, and ensure that everything is documented accurately with no unauthorized changes.

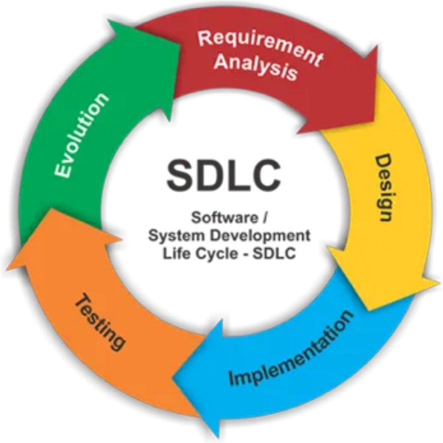
5.3 Reviews

Technical reviews during the project help to ensure that a qualified product will be delivered. The Configuration Manager will verify and make sure all project components are identified correctly, and changes made to the documents are also noted down. These reviews will be performed periodically to check on the correctness of configuration status and to baselining the codes for audits.

Also, the Configuration Manager will give constructive feedback on the working product.

5.4 Configuration Management Plan Maintenance

The life cycle of Planit will follow the Agile Software Development Cycle.



There are a total of 6 steps in the development cycle, which Plantit will follow an incremental and iterative approach, and repeated when needed.

The QA team will be responsible for handling all updates and maintenance on the CM plan throughout the development lifecycle. Updates to the CM plan will take place every 3 months or whenever necessary. Possible CM Plan Maintenance activities that may take place include:

- Revising CM organisational structure (personnel, resources)
- Revising SCM activities like identification methods, control, auditing and reporting methods.

- Changing the SCM schedules
- Adjusting SCM resources

The Configuration Manager will be responsible for the periodic updates to this plan and on all of the above activities. All of which will follow the **IEEE 828 Standards** for Software Configuration Management Plans.