Go Where GaiGai
Software Configuration Management Plan

Version: 1.0
Last Modified: 23/10/2022

TABLE OF CONTENTS

1   Identification

This document is the Configuration Management (CM) Plan of Team ONE which will define the team's configuration management plans for their Go Where GaiGai project.

1.1   Document overview

This document contains the software configuration management plan of software Go Where GaiGai.

1.2   Abbreviations and Glossary

1.2.1   Abbreviations

Abbreviation	Full Form
CI	Configuration Item
CM	Configuration Management
CMS	Configuration Management System
SCM	Software Configuration Manager
SCI	Software Configuration Item
VDD	Version Description Document
QA	Quality Assurance

1.2.2   Glossary

Word/Phrase	Explanation
Branch	A particular form of document/application that differs in certain aspects from an earlier form or other forms of the same thing.
Merge	To develop a document/application independently in parallel, with the possibility of merging in the future.
Version	To combine different branches of work into a main branch.

1.3   References

1.3.1   Project References

#	Document Identifier	Document Title
1	PP	Project Plan
2	QP	Quality Plan
3	RMP	Risk Management Plan
4	SRS	Project Plan

1.3.2   Standard and regulatory References

#	Document Identifier	Document Title
1	SCMSSE	IEEE 828-2012 – IEEE Standard for Configuration Management in Systems and Software Engineering
2	SSCMP	IEEE Standard for Software Configuration Management Plans

1.4   Conventions

Typeface	Usage	Example
Bold	To assert emphasis on the text. Used in headers and titles as well.	**1.4 Conventions**
Highlighted	To assert special emphasis on important information in a text.	Web Application: Go Where GaiGai
Italics	To assert minor emphasis on the text. Also used to refer to file names.	This document is the *Software Configuration Management Plan* for team ONE.
Red	To indicate comments and mark out text that requires changes.	Go Where GaiGai helps to … (Change this to "aims to" instead)
Underline	To help draw attention to the text. Also used to represent hyperlink.	Refer to the Release Plan for the release strategy of Go Where GaiGai.

2   Organization

The software configuration management is managed by relevant members of the project, which includes the Project Manager/Release Manager, Developers, QA Manager and Engineer. The responsibilities of the project are shown below in the following structure:
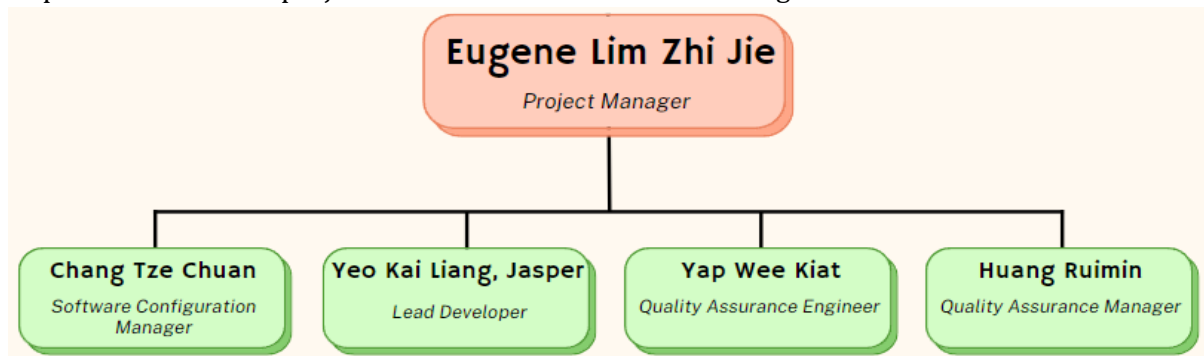
2.1   Activities and responsibilities

The functions required to manage the configuration of the software and responsibilities are listed below.

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	QA Manager
Inspect configuration records	QA 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	QA Manager

2.1.1 Decisions process and responsibilities

Responsibilities during reviews, audits and approvals are listed below:

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	QA Manager
Present the records of the documentation management process	SCM

3 Configuration identification

The purpose of configuration identification is to maintain control of our project by uniquely identifying revisions of our project and the component parts of each revision. Through configuration identification, we will be able to understand the status of configuration items as

they progress through the development process. Listed below are additional advantages from a proper configuration identification maintained in a project:

- Determines the structure (hierarchy) of a product and the organization and relationships of its configuration documentation and other product information
- Documents the performance, interface, and other attributes of a product
- Determines the appropriate level of identification marking of product and documentation
- Provides unique identity to a product or to a component part of a product
- Provides unique identity to the technical documents describing a product
- Modifies identification of product and documents to reflect incorporation of major changes
- Maintains release control of documents for baseline management
- Enables correlation of document revision level to product version/configuration
- Provides a reference point for defining changes and corrective actions.

3.1 Identification rules

3.1.1 Identification rules of configuration items

3.1.1.1 Identification of a configuration item

The format for identification of configuration item is:
XXX_Vm.n

- where: "XXX" is the configuration item name
- "_V" is the delimiter for version number
- "m" is a digit indicating the major version category
- "n" is a digit indicating the minor version category

3.1.1.2 Version number of a configuration item

Software versioning is a way to differentiate the different states of computer software and documentation as it is developed and released. The version number consist of 2 numbers separated by a full stop. It is required for any delivery of any configuration item.
The definition rules of a version number are as follows:

- Major edits will increment the major version digit
  - ➢ Adding or removing significant features
  - ➢ Redevelopment of document
- Minor edits will increment the minor version digit
  - ➢ Bug fixes
  - ➢ Small changes to code

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:

- "document type" is the type of document
- "document number" is an incremental number, with a separate list for each document type.
- "revision index" designates the approved iteration of the document. The revision index is V1 for the first iteration, V2 for the second 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 rules of version number of a configuration item in 3.1.1.2.

3.1.3 Identification rules of a media

There is no form of media used in the Go Where Gai Gai website.

3.1.3.1 Internal identification

The identification of a media is described below:
<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 acceptation or validation reviews associated to 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

Indicated below are the baselines to be established and when and how they will be defined and controlled.

Examples of baselines :
- functional baseline (FBL), which describes the system functional characteristics;
- allocated baseline (ABL), which describes the design of the functional and interface characteristics,
- product baseline (PBL), which consists of completed and accepted system components and documentation that identifies these products.
- Development baseline, which indicates the state of work products amid development.

All baselines will be defined and controlled after:
- Project Manager's approval
- Meeting QA team approval standards
- The project has been completed and tested at least once

4 Configuration control

Configuration control is a method for controlling, approving and tracking changes to a product's deliverables and processes.

4.1   Change Management

It is inevitable to make changes in software projects due to changing target users' demands or bugs and errors. The process for controlling the changes to the baselines and for tracking the implementation of those changes are listed as follows:

Problem resolution:
- When a change request is accepted by the project manager/product manager, a branch is created in the SCM
- The branch identification is done as per the configuration item identification protocol
- Branch content is where the required changes will go to, before going through the rest of the change management workflow

Multiple configuration:
- When a change request is accepted by the project manager/product manager, a branch is created in the SCM
- The branch identification is also done as per the configuration item identification protocol
- Branch content is where the required changes will go to, before going through the rest of the change management workflow

4.2   Interface Management

Go Where GaiGai will interact with the following 3rd parties:
- Yelp
- Google

It requires respective interfaces for each 3rd party.

The following is the procedure to identify the interface requirements for each interface:
- Identify the interface
- Identify the characteristics of the interface
- Define and determine the interface responsibilities

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. It ensures that all configuration data and documentation are recorded as each CI progresses through its life cycle from test to production to retirement.

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: The software configuration management tool records, for each source file or group of source files, a comment where is described the modification.
- 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,
- The version number,
- The creation date of the Version Description Document (VDD),

The SCM writes the VDD, which serves the following purposes:
- Primary configuration control document used to track and control versions of software being released to testing, to implementation, or to the final operational environment.
- Provides a summary of the features and contents for a specific software build or release, and facilitates product implementation, testing, operations and maintenance.
- Every unique release of the software, starting from the initial release, shall be described by a VDD. Even if there are multiple forms of the same software being released at approximately the same time (e.g., different forms to different clienteles), each must have a unique version number and VDD.
- The VDD is part of the SCI product baseline. When distributed, the version description document should be sent with a cover memo that contains an executive summary, on a single page, the significant changes included in the release.
- The label and version will assist in uniquely identifying each VDD and trace them accordingly.

5.1.3 Configuration status diffusion

The SCM and the quality manager will write the VDD as described above.

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,
- The configuration states sorted chronologically.

5.2 Configuration audits

Configuration audits are conducted to determine that a system or item meets it functional requirements and has been built in accordance with its blueprints, source code, or other technical documents. It must be performed on a configuration item before it is released and define the following:
- The objective and schedule of the audit
- The configuration items under audit or review
- The participants involved and the procedures for conducting the audit or review
- Documents required and the procedure for recording and deficiencies and reporting corrective actions as well as the approval criteria

Compliance with the CM Plan assessed as follows:
- **Functional Configuration Audit** - The objective of the functional audit is to provide an independent evaluation of a software product, verifying that its configuration items' actual functionality and performance is consistent with the relevant requirement specification.
- **Physical Configuration Audit** - A Physical Configuration Audit is used to examine the actual configuration of the Configuration Item (CI) that is representative of the product configuration in order to verify that the related design documentation matches the design of the deliverable CI. It is also used to validate many of the supporting processes

that the contractor uses in the production of the CI. This is also used to verify that any elements of the CI that were redesigned after the completion of the Functional Configuration Audit also meet the requirements of the CI's performance specification.

5.3 Reviews

Technical reviews help to examine the suitability of the software product for its intended use and identifies discrepancies from specifications and standards. It ensures that an early stage the technical concepts are used correctly. This allows the access to the value of technical concepts and alternatives in the product. In addition, there will be consistency in the use and representation of technical concepts.

The primary task of a configuration manager is the preparation of complete configuration documentation and overseeing the management of configuration items. Configuration managers not only coordinate the harmonisation of the individual components of a product, but are also responsible for its overall configuration description. This means that they plan and execute configuration management throughout the lifecycle of a project.

5.4 Configuration management plan maintenance

The following is the Software CM Plan Maintenance for the Go Where GaiGai project:
- SCM is in charge of monitoring the configuration plan
- A review will be conducted weekly to ensure that all configurations are adhered to and any changes made to the CM plan will result in an update of the configuration management plan.
- The Project Manager must approve the changes before proceeding with the update. The changes will be notified through email to the project team by the SCM

The SCM will maintain the CM Plan as follows:
- Revising CM organization structure, CM activities and CM schedules
- Adjusting CM resources

Any changes made will adhere to the IEEE 828 SSCMP.