VOTAFUN

SOFTWARE CONFIGURATION MANAGEMENT PLAN

Version 1.1
25/10/2023

VERSION HISTORY

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Ng Yue Jie Alphaeus	12/10/2023	Ng Yue Jie Alphaeus	12/10/2023	Initial Software configuration management plan draft
1.1	Roy Lau Run-Xuan	13/10/2023	Ng Yue Jie Alphaeus	25/10/2023	Identification added Organisation added Configuration identification added Configuration control added Configuration support activities added

TABLE OF CONTENTS

1. Identification

1.1. Document overview

This document contains the software configuration management plan of VotaFun. It describes the configuration management process for the project and the activities that support the configuration management process.

1.2. Abbreviations and Glossary

1.2.1. Abbreviations

Abbreviations	Meaning
SCM	Software configuration manager
VDD	Version description document

1.2.2. Glossary

Word	Meaning
Branch	A separate line of development where developers can independently add or remove features without affecting others' work. The developer can later merge this line to incorporate features into the main codebase.
Version	A specific iteration of the software.
Baseline	A specific reference point that serves as a reference for future development.

2. Organisation

The software configuration is managed by members of the project team.

Responsibilities of the SCM are shared between:

- Software configuration manager – Ng Yue Jie Alphaeus
- Quality manager – Tran Trung Dung (Charles)

2.1. Activities and responsibilities

The different activities required to manage the configuration of the software and responsibilities are listed in the following tables.

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

Activities during the project lifecycle	Person responsible
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. Decisions process and responsibilities

The responsibilities of different project members 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	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 following are the various configuration items in VotaFun.

- Source code
- Use case model
- Entity relational diagram
- Software architecture
- Unit test
- Test coverage report

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 were modified.

The definition rules of a version number are the following:

- The identification scheme of a configuration item follows the format of <name of configuration item>_V<major version.minor version.patch version>
- The major version number is incremented when there is a major change in the configuration item. The changes refer to
    - Significant architectural changes in the software where multiple features or components are added or removed.
    - Change in software architecture.
    - Change in business logic.
- The minor version number is incremented when there is a minor change in the configuration item. The changes refer to
    - Changes to enhance the performance or sustainability of VotaFun.
    - Bug fixes for VotaFun.
- The patch version is incremented when there are very small changes to the configuration item.
    - Minor bug fixes that do not affect the functionality.
    - Small changes to the software.

3.1.2. Identification rules of documents

3.1.2.1. Description of documents identifiers

The identification scheme of documents is as follows:

- CZ3002_VotaFun_<document type>
- “document type” refers to the type of document and the following lists all possible document types.
    - Project proposal
    - Project plan
    - System requirements specification
    - Design report on software maintainability
    - Risk management plan
    - Release plan
    - Quality assurance plan
    - Configuration management plan
    - Change management plan
    - Test plan

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 the following:

- The revision index is in the form of V<major version.minor version>.
- The major version is incremented when there is a significant change in the documentation affecting multiple sections or requires the rewriting of the document.
- The minor release is incremented when there are minor changes to the document or changes to fix errors or grammar.

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 test review of the project.

3.3. Configuration Baseline Management

The following describes the different baselines that will be established for VotaFun.

- Functional baseline (FBL). The FBL describes the overall high-level functional requirements of VotaFun and describes what features VotaFun should have.
- Allocated baseline (ABL). The ABL describes the design of VotaFun and the functional and interface characteristics. It shows how VotaFun should be built, based on the requirements.
- Product baseline (PBL). The PBL represents the final completed version of VotaFun and contains all source code, design specifications and documentation used to build VotaFun.

4. Configuration control

4.1. Change Management

The following describes the process for controlling changes to the baselines and for tracking the implementation of those changes.

Process for problem resolution:

- Change requests are emitted from by the project manager according to the problem resolution process.
- When a change request is accepted by the project manager/product manager, a new branch is created in GitHub.
- The branch identification will be in the format “feature/<name>”. The name will be a title that describes the change.
- Branch contains the changes made to the software configuration item.

Process for multiple configurations:

- Change requests of configuration files are emitted by the product manager according to the production procedure.
- When a change request is accepted by the project manager/product manager, a branch is created in GitHub.
- The branch identification will be in the format “release/<name>”. The name is the version number that identifies the version of VotaFun.
- Branch contains the changes made to the software configuration item.

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. CSA tacks all software and related documentation throughout the software lifecycle, and it provides a record of the changes of every configuration item.

5.1.1. Evolutions traceability

The modification of different software configuration items is traced according to their type.

- Documents: The modification sheet number identifies the origin of the modification. The modified paragraphs in the document are identified, if possible, by revision marks.
- Source code: The software configuration management tool records, for each source file or group of source files, a comment to describe 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 and the team member who did the modification. The modification sheet allows the team to have a history of changes of different software configuration items.

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 VDD

The SCM writes the VDD.

5.1.3. Configuration status diffusion

The SCM and the quality manager will be responsible for writing the VDD and disseminating the VDD to the project team and client. This ensures that all stakeholders have the latest information about VotaFun software configuration items.

5.1.4. Configuration status records storage

The configuration status records are stored in a configuration folder located in the project team's Google Drive. The configuration folder contains:

- The requests sorted by record number,
- The software documents,
- The VDD's,
- The configuration states are sorted chronologically.

5.2. Configuration audits

To ensure compliance with the configuration management plan, the following audits are performed:

- Functional Configuration Audit.
- Baseline Audit.

A peer review audit will be done to ensure compliance with the configuration management plan. The following shows an overview of the audit process.

- Before the audit begins, the auditor will determine a clear objective and scope of the audit. The audit scope will mention the configuration item to be reviewed and the criteria for compliance.
- During the audit, the auditors will review the configuration item to ensure that it follows established specifications and documentation as stated in the baselines. Any deviation will be highlighted and marked for corrective actions.

- Once corrective actions are identified and assigned, the project team will monitor the process to ensure that actions are made and changes documented.
- The entire audit process will be documented and archived in the project team's Google Drive.

5.3. Reviews

Technical reviews during the project serve as critical checkpoints that contribute to the establishment of different baselines (FBL, ABL, PBL) or branches in version control. The SCM is responsible for ensuring that these reviews are done before the establishment of any baseline or branches.

During the project lifecycle, technical reviews contribute to the establishment of different baselines. Before the SCM establishes the FBL, a technical review ensures that all requirements are captured and comprehensive. It also ensures that all software components designed meet all functional requirements before establishing the ABL. Finally, it also ensures that the product created has met all requirements and functionalities which establishes the product baseline.

Having a technical review helps determine when branches should be created. Depending on the change, the technical review can trigger the creation of a feature branch, or a release branch. Feature branches are created when developers need to add new features or fix a bug, while a release branch represents a new stable baseline version of VotaFun.

5.4. Configuration management plan maintenance

The quality assurance team is responsible for the maintenance of the configuration management plan. A review of the configuration management plan is done every year, with the plan updated by the QA team as needed.