NANYANG
TECHNOLOGICAL
UNIVERSITY
SINGAPORE

Drawdiculous

Software Configuration Management Plan

Version 1.0

Prepared by Team Luck

Bian Hengwei	U1923732B
He Yinan	U1922693C
Jin Han	U1922733A
Lee Yu Jie Melvin	U1922533G
Renganathan Ramasamy	U1922494L
Sannabhadti Shipra Deepak	U1822459L

Table of Contents

1. Identification

1.1 Document overview

This document contains the software configuration management plan of software Drawdiculous.

1.2 Abbreviations and Glossary

1.2.1 Abbreviations

- SVN: Apache Subversion, SVN is used to manage and track changes to code and assets across projects.
- SCI: Software Configuration Item.
- SCM: Software Configuration Manager.
- CM: Configuration Management.
- CMP: Configuration Management Plan.
- VDD: Version Description Document.
- VCS: Version Control System.
- ASCII: American Standard Code for Information Interchange.
- SRS: Software Requirement Specification.
- CMMI: Capability Maturity Model Integration.
- SemVer: Semantic Versioning.
- PM: Project Manager.
- SOUP: Software of Unknown Pedigree

1.2.2 Glossary

- Baseline: A foundation that has been formally agreed upon on which product is built on.
- Branch: A copy of a codeline, managed in a VCS.
- Version: State of a configuration item at a well-defined point in time.
- Variant: Versions that coexist.
- Pascal notation: Names begin with an uppercase letter. Multiword names are written with no underscores and with the first letter of each word capitalized.
- Camel notation: Names begin with a lowercase letter. Multiword names are written with no underscores and with the first letter of each word capitalized.

1.3 References

1.3.1 Project References

#	Document Identifier	Document Title
[R1]	1	Project Proposal
[R2]	2	System Requirement Specifications
[R3]	3	Quality Plan
[R4]	4	Risk Management Plan
[R5]	5	Project Plan
[R6]	6	Release Plan
[R7]	7	Design Report on Software Maintainability
[R8]	8	Change Management Plan

1.3.2 Standard and regulatory References

#	Document Identifier	Document Title
[STD1]	1	IEEE SQA 730-2014, IEEE 730 Standard on Software Quality Assurance

1.4 Conventions

Identifier Type	Rules for Naming	Examples
Packages	Written in all-lowercase ASCII letters and should be of the top-level domain names.	com.teamluck.drawdiculous.app.handler
Classes	Pascal notation	DrawActivity
Interfaces	Pascal notation	Protocol
Methods	Camel notation	handleGuessUpdate()
Variables	Camel notation	painterIndex
Constants	All letters capitalized with words separated by underscores.	SERVER_ID

2. Organization

The software configuration is managed by members of the project, with specific tools. Responsibilities are shared between

- SCM: Lee Yu Jie Melvin

- Project manager: Bian Hengwei
- Technical manager: Renganathan Ramasamy

2.1 Activities and responsibilities

The functions required to manage the configuration of the software 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 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	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

Code: Version number and details are written in the JavaDoc comment of the class.

- Source code

Documentation:
"DocumentName_versionNumber.x" will be used as the naming convention of the documentations. Where versionNumber is the version number and x is the file extension.

- Project proposal
- SRS
- Quality plan
- Project plan
- Risk management
- Design report on software maintainability
- Configuration Management Plan
- Change Management plan
- Release plan
- Test plan
- Test Cases and Requirements Test Coverage report
- CMMI

System designs:
"DocumentName_versionNumber.x" will be used as the naming convention of the documentations. Where versionNumber is the version number and x is the file extension.

- Use case model
- Software architecture diagram

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.

SemVer versioning system is used. The definition rules of a version number are the following:

- The MAJOR version is incremented when incompatible API changes are made.
- The MINOR version is incremented when functionalities are added in a backwards compatible manner.

- The PATCH version is incremented when backward compatible bug fixes are made.

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:

- Foo for FOO documents,
- BAR for bar documents

" document number " is a 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.

- MAJOR version and MINOR version will be used
- MAJOR version is incremented when there is changes to the overall design
- MINOR version is incremented when changes is made to some less important content

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 management baselines are:

- 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.

4. Configuration control

Configuration changes and variation management is described in the following subsections.

4.1 Change Management

The process for controlling changes to the baselines and for tracking the implementation of the following changes:

Problem resolution:

- Changes 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 branch is created in the SCM.
- The branch identification is the title of the parent branch and the problem title separated by an underscore.
- Branch content is the updates and changes to the code in an attempt to solve the problem.

Multiple configuration:

- Changes 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 the SCM.
- The branch identification is the title of the parent branch and the change name separated by an underscore.
- Branch content is the related changes.

4.2 Evolutions control of SOUP items

SOUPs are freezed at the beginning of the project. The QA team will keep track of any new releases in the SOUPs we use. Reviews will be conducted by the PM and Lead Developer when there are new releases in the SOUPs to analyse and decide if the system code needs to be upgraded to use the latest versions of the SOUPs.

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: 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,
- 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 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
- The configuration states are sorted chronologically

5.2 Configuration audits

The following formal audits are made to assess the compliance with the CMP:

- Baseline audit

- Functional configuration audit
- Software configuration audit

5.3 Reviews

Any experimental changes in the program will be done through branching. Successful branches that are accessed by the QA team will be merged into the main branch. SCI that has been agreed upon through walk-through and review will be set as the Baseline. Technical reviews and configuration reviews will be performed periodically to verify the correctness of the configuration status of the main branch and experimental branches.

Role of configuration manager:

- Execute the CMP throughout the project lifecycle
- Oversee the daily management of SCIs
- Review and update CMP
- Ensure that all team members follow the CMP

5.4 Configuration management plan maintenance

Activities and Responsibilities:

PM and Lead Developer:

The PM will review all assets and set new SCIs according to new technology or business needs with the help of the Lead Developer.

QA Manager, QA Engineer and Release Engineer/Manager:

The QA team, together with the Release team, will review and update the CM plan and verify component correctness before each release which is done every three months.

Front-end Developer and Back-end Developer:

The developers will follow the processes defined in the CMP and constantly maintain the code and documentation.