



CONFIGURATION MANAGEMENT PLAN

Online Trail Making Test (TMT)

Version 1.0

Date:3/4/2020

By: Ang Zhan Phung, Ang Yong Xin, Brenda Ng Xin En, Emmanuelle Vania, Sam Jian Shen, How Mo Xuan

Software Configuration Management Plan of LarkDetect			
Version	Date	Author	Comments
1.0	April, 3,2020	Ang Yong Xin	First version

Table of Contents

1. Identification	4
1.1 Document Overview	4
1.2 Abbreviations and Glossary	4
1.2.1 Abbreviations	4
1.2.2 Glossary	4
1.3 References	4
1.3.1 Project References	4
2. Organization	5
2.1 Activities and responsibilities	5
3. Configuration identification	6
3.1 Identification rules	6
3.1.1 Identification rules of configuration items	6
3.1.2 Version number of a configuration item	6
3.1.2 Identification rules of SOUPs	7
3.1.3 Identification rules of documents	7
3.2 Reference configuration identification	8
3.3 Configuration Baseline Management	8
4 Configuration control	8
4.1 Change Management	8
4.2 Evolutions control of SOUP items	9
5 Configuration support activities	10
5.1 Configuration Status Accounting	10
5.1.1 Evolutions traceability	10
5.1.2 Setting up Configuration status	10
5.1.3 Configuration status diffusion	10
5.1.4 Configuration status records storage	10
5.2 Configuration audits	11
5.3 Reviews	11
5.3.1 System Requirements Review (SRR)	11
5.3.2 Design Review (DR)	12
5.4 Configuration management plan maintenance	12

1. Identification

This document amplifies the “§4 Configuration management” of the Project Management Plan. If you instantiate this document, leave empty the §4 in the Project Management Plan and add a reference to this doc.

1.1 Document Overview

Configuration management (CM) is the ongoing process of identifying and manage changes to deliverables and other work products. It will help to ensure consistency, traceability and correct version will be delivered to the client.

This document contains the software configuration management plan of LarkDetect.

1.2 Abbreviations and Glossary

1.2.1 Abbreviations

- SVN: Apache Subversion (Open source versioning and revision control system)
- SRS: Software Requirement Specification
- SCM: Software Configuration Manager
- SCI: Software Configuration Item
- VDD: Version Description Document
- CM: Configuration Management

1.2.2 Glossary

- Version: the state of configuration item at a well-defined point of time. Each version of a file has a unique version number.
- Branch: A line of development that exists independently of another line, yet still shares a common history, and can be merged in the future.

1.3 References

1.3.1 Project References

#	Document Identifier	Document title
1	It consists of the test result average timing. The following link is the website link: http://apps.usd.edu/coglab/schieber/psyc423/pdf/IowaTrailMaking.pdf	Trail Making Test(TMT) Parts A & B

2. Organization

- Project Manager: Ang Zhan Phung Edmund
- Quality Manager: Emmanuelle Vania
- Requirement Specification Analyst: Brenda Ng Xin En
- System Architecture: Ang Yong Xin
- Usability Analyst: Sam Jian Shen
- Software Configuration Manager: Ang Yong Xin, How Mo Xuan
- Developers: Sam Jian Shen, How Mo Xuan

2.1 Activities and responsibilities

The activities required to be managed for 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 and Developers
Install the software configuration repository tool and set up the database	SCM and Developers
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 authorize deliveries	Project manager
Backup spaces	SCM
Do configuration audits	Quality Manager
Inspect configuration records	Quality Manager
Archive reference version	SCM

Activities at the end of each configuration change	Person responsible
Present a summary of the configuration state of the component being impacted by the changes	SCM
Document the changes	SCM
Conduct review and testing for the changes	SCM, Project Manager, Quality Manager

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

3.Configuration identification

3.1 Identification rules

3.1.1 Identification rules of configuration items

Software Configuration item will consist the following items:

- Source code
- Executable code
- Documentation
 - Project Proposal
 - Quality Plan
 - Risk Management
 - Test Plan
- System designs:
 - Use case models
 - Software architecture
 - Class diagram
 - Software prototype
- Specifications:
 - Project requirements specification

The identification of configuration item will be in the format of itemName_version where version is the version of the configuration item

3.1.2 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:

- It will be using semantic versioning number system with the following format: <Major Update>. <Minor Update><Bug Fix>
- Major update numbering will be incremented when there is changes that is due to the operating system of the internet version. For example: Our product is unable to use on Chrome version 55.0.2883 but can use on Chrome version 54.0.2840. Thus, we will have an increment on this numbering after we fixed the problem.
- Minor update numbering will be incremented when there is an update on our product functionality that will increase the user friendliness performance or there is changes in our interface design.
- Bug fix will be incremented when there is an amount of bug fixed on the product.

3.1.2 Identification rules of SOUPs

3.1.2.1 Identification of a numbering SOUP

SOUP is a software that has not been developed with a known software development process or methodology. Operating Systems, Hardware Drivers and open-source libraries are few examples of SOUP, and all these depends on which category it belongs to. Thus, we will be using the ID of the SOUP manufacturer for easy identification.

3.1.2.2 Version number of a SOUP

We will be using the version number of the SOUP manufacturer in our product.

3.1.3 Identification rules of documents

3.1.3.1 Description of documents identifiers

The identification of documents is described below:

XXX_<document type>_<document number>.<revision index>

where:

- “document type” is the name of the document.

For example:

- Change_management_plan is our change management plan document.
- Risk_management_plan is our risk management plan document.
- SRS is our software requirement specification document.
- “document number” is an increment number, with a separate list for each document type
- “revision index” designates the approve iteration of the document. The revision index is Version 1.0 for the first iteration, Version 1.1 for the second and so on.

3.1.3.2 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:

- It will be in decimal numbering format. For example: 1.1, 1.2, 1.3, ..., 2.0

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 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

A baseline is a set of configuration items formally designated and fixed at a specific point in time. A baseline with a SCI will be established once a delivered form of the SCI has been completed or after the formal review of a SCI.

LarkDetect CM establishes 3 types of baseline for easy management and documentation.

The managed baselines are:

- Functional baseline (FBL), which defines the system functional characteristics. It is simply documenting all our system's functionality and capability.
- Allocated baseline (ABL), which describes the design of the functional and interface characteristics. It will be described in the early stage of design specification.
- Product baseline (PBL), which contains the selected functional and physical documentation that identifies these products.

4 Configuration control

In this section, we will be describing the way of managing configuration changes and variances in configuration.

4.1 Change Management

The process for controlling changes to the baselines and for tracking the implementation of those changes are listed in this section.

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 changes where each word will be separated by an underscore (_).
- Branch content is the related changes

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 changes where each word will be separated by an underscore (_).
- Branch content is the related changes.

For all changes, the change request will be submitted to the SCM for review

4.2 Evolutions control of SOUP items

Due to the rapid evolutions of SOUPs, the version would also change frequently. Thus, a clear list of dependencies and their versions will be listed out for easy identification.

For example:

```
{
name: LarkDetect

Dependencies:
  {
    • JQuery: 2.3.1
    • JQuery-validation: "2.3.4"
  }
}
```

Before we change to a newer version of the SOUPs, we will implement the following steps:

- we would conduct unit testing and use it as a reference of the system quality.
- We would also perform risk analysis of the SOUPs with a list of known bugs. This is because without any testing of the version, it will be often difficult to identify which library version has the bug.
- The updated version must be able to fulfill all the functionalities that the older version has.
- If the updated version has lesser bugs and is more efficient than the older version of our product, we will implement it into our product.

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.

Below are some of the example of the configuration status accounting process:

- Monitor status of change request
- Keeps a record of all the changes made from the previous baseline.
- Allows to check previous releases/ versions to be extracted for testing.

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 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 sorted chronologically.

5.2 Configuration audits

In Software Configuration Management (SCM) audits, 2 type of audits will be performed in our project.

The 2 type of audits are:

- Functional Configuration Audit (FCA)
 - It is an evaluation of our completed software product in terms of completeness, performance and functional characteristics to the functional baseline requirements.
 - Step for performing this audit are:
 - Preparing a list of requirement test case and determine whether it pass the test case.
- Physical Configuration Audit (PCA)
 - It will be used to confirm consistency of design and user documentation with the code.
 - It usually identifies what are the components to be deployed to the Project Repository.
 - Step for performing this audit is:
 - Inspecting the system documentation against the item listed in the product release. Such item includes source code, release notes and system deliverables.

5.3 Reviews

As our system design evolves through the 3 different type of baselines (FBL, ABL, PBL), 2 different type of reviews must be performed by all the managers to ensure our product will not be compromised by the evolution.

The 2 different type of reviews are:

- System Requirements Review (SRR)
- Testing Review (TR)

Software Configuration manager are to ensure all the project configuration changes are properly documented and the configuration status are correct.

We'll have team meeting every week to keep track of the team progress on any new updates, or any problem that they encounter. The meeting record details will be updated in the meeting minutes document. These records will also be extremely helpful when comes to documentation.

5.3.1 System Requirements Review (SRR)

The purpose of SRR is to ensure the system can completely describe the desired system. It will be using the project SRS as a reference and is conducted by all the managers. Blackbox and white box testing will be conducted under this review.

5.3.2 Design Review (DR)

The purpose of DR is to ensure the system under review can meet complete the project within the specific timeline and proceed to demonstration and testing. Weekly meeting will be conducted to check the progress ensure it is able to be delivered within the specific timeline. Project chart chart and proposal will be used as a reference for this review.

5.4 Configuration management plan maintenance

Configuration management plan maintenance will be done by the SCM. Any updates by the developer regarding the configuration item must inform the SCM, so he will be able to perform update on the management plan and ensure all the documentation and configuration status are correct.