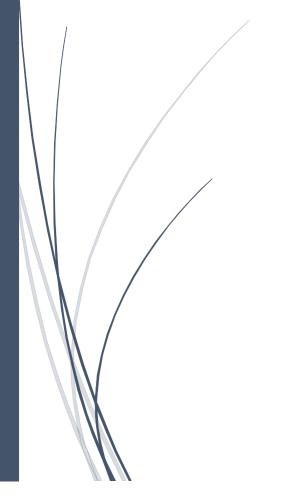# Configuration Management Plan

EXIT6 – Trip Planning Application for Singapore

Delivered by,
EXIT6 Team
Sim Long Siang(U1522053H)
Seshadri Madhavan (U1322790J)
Huang Jian Wei(U1521567A)
Lim Zi Yang(U1522218E)
Lim Hao Zhe(U1521589B)
Tan Jun Qiu(U1321638C)

Document Change record

Revision	Description of Change	Approved by	Date
0.1.0	Initial Template	Sim Long Siang	13/03/2017
1.0.0	Release	Seshadri Madhavan	21/03/17

Contents

Identification

This is the configuration management document for EXIT6 – Trip Planning Application for Singapore. It contains the software configuration management plan (SCMP) for this project. The plan ensures that a system is in place to control, maintain, and trace any configurations made to the project. It involves identifying and controlling configuration items and the changes made to the project. In addition, recording and reporting activities of configuration items and relating changes are also part of the management.

Document overview

The SCMP in this document is divided into sections and is illustrated in the table below:

Section #	Section Type	Section Description
1	Identification	Highlights the objective and purpose of SCMP, and the abbreviations and technical terms used in this document.
2	Organization	Details the organization of the EXIT6 team, and the roles and responsibilities of its members with respect to configuration management
3	Configuration Identification	States the identification rules of SCIs
4	Configuration Control	Details the process for managing configuration changes in the SCIs of *EXIT6 Trip Planning Application*
5	Configuration Support Activities	Highlights the assessment activities that assess the level of compliance of EXIT6 development team to the procedures and standards established in the SCMP, in implementing changes to the SCIs

Abbreviations and Glossary

Abbreviations

The list of abbreviations in the table below are used in this document:

Abbreviation	Description
SCMP	Software Configuration Management Plan
SCM	Software configuration Management
SCI	Software Configuration Item
SCR	Software Change Request
SRS	System Requirement Specification
QA	Quality Assurance
VDD	Version Delivery Description

Glossary

The glossary list introduces several technical terms used in this document:

Term	Description
Software Configuration Management (SCM)	The discipline of identifying the configuration of a system for controlling changes to this configuration and maintaining the integrity and traceability throughout the system lifecycle.
Software Configuration Management Plan (SCMP)	A reference document for the SCM process including responsibility distribution and details for each part of the process.
Bug Repository Tool	A bug tracking software that keep records of reported software bugs during software development project.
Configuration Repository Tool	A database software that keeps configuration Items of the software project.
Configuration Freeze	A preventive process to avoid SCIs from getting changed or edited after release.
Software Configuration Item (SCI)	A single, structural unit/component in the software that needs to be managed.
Version Delivery Description (VDD)	A reference document that track versions of SCIs to be released to the operational environment.

References

Standard and regulatory References

The list of Standard and regulatory references for this document are illustrated below:

#	Document Identifier	Document Title
[STD1]	1	IEEE Standard for Software Configuration Management Plans – IEEE Std 828 – 1998

Organization

EXIT6 Trip Planning Application will be managed by members in the EXIT6 team. The responsibilities of configuration management are shared among the Project Manager, Release Manager and QA Manager.

Activities and responsibilities

Illustrated below are the activities required to manage the configuration of the software and responsibilities:

Task	Person-In-Charge
Pre-Development	
Identify the configuration items	QA Manager
Install bug repository tool and database	QA Manager
Install the software configuration repository tool	Project Manager
Manage and structure the reference space	Release Manager
Define the configuration processes	QA Manager
Development	
Export components for modification, test or delivery	QA Manager
Create version, write version delivery document	Release Manager
Approve reference configurations	Project manager
Verify version to be delivered and authorize deliveries	Project manager
Backup spaces	Release Manager
Do configuration audits	QA Manager
Inspect configuration records	QA Manager
Archive reference version	Release Manager
Management	
Manage versions and archives	Release Manager
Manage configuration records	Release Manager
Produce reports and statistics	Release Manager
Manage reference space and its access control list	Release Manager
Manage spaces backup and archive media	Release Manager
Manage quality reports	QA Manager

Decisions process and responsibilities

Illustrated below are the Activities and Responsibilities during reviews, audits and approvals:

Activity	Person-In-Charge
At the end of an activity of the project	
Do a configuration freeze	Release Manager
Present a configuration state of the components impacted by the activity	Release Manager
Present a documentation state of the components impacted by the activity	Release Manager
During a configuration management process audit	
Do the configuration management process audit	Project Manager
Present the records of the configuration management process	Release Manager

Activity	Person-In-Charge
At the end of an activity of the project	
Present the quality records of the configuration management process	QA Manager
Present the records of the documentation management process	Release Manager

Configuration identification

Identification rules

This section will highlight the identification of items to be controlled throughout the whole lifecycle of the project and the establishment of identification scheme.

Identification rules of configuration items

SCI will be identified based on its purpose and development phase. The SCIs that form EXIT6 Trip Planning Application consist of the following:

- Source codes (Java files, xml layout files, external libraries)

- Database information

Source code files will be named in the format " <code_function>_<version_number>. <file_format> ". In addition, source files will be separated into two different packages: one for front-end functions and one for back-end services. Documentation files will be named in the format "EXIT6 - <title>_<version_number>. <file_format>".

Version number of a configuration item

Assignment of version number to configuration item is essential in keeping track of the latest changes done to it. It also reflects the impact of changes toward the item by having a specific notation in the version number format.

The version number of a configuration item will be of the format "Item name (major). (Minor). (Revision)". The numbers will be assigned in increasing order of 0.1 and every increase reflects the latest developments in the project, starting from the initial draft version 1.0. Each notation is defined as follows:

- **Major:** SCIs has implemented changes in main features of the application

- **Minor:** SCIs has major bug fix or changed or added minor features to the application. *Minor* will restart from zero when *Major* increments by 1

- **Revision:** SCIs has changed or added features that are least significant to the project. *Revision* will restart from zero when *Minor* increments by 0.1.

Identification rules of documents

Like the identification rules of configuration items, documents will be identified based on its purpose and development phase. The documents that guides EXIT6 Trip Planning Application consist of the following:

- Project Proposal

- System Requirement Specification (SRS)

- Use Case Diagram and Descriptions

- Test Plan

Documents will be named in the format "EXIT6 - <title>_<version_number>. <file_format>".

Version number of documents

Like configuration item, assignment of version number to documents is essential in keeping track of the latest updates on content modification. The version number of a document will be of the format "EXIT6 - <Document Title>_<version number>.<revision index>", starting from the initial draft version 1.0. Each notation is defined as follows:

- **Document Title:** Name of the document. (eg. Project Plan, Quality Plan)

- **Version number:** Number will be incremented by 1 when there is a major update to the document (e.g. content modification, removal, addition). These updates must be approved by the project manager.

- **Revision Index:** Index will be incremented by 1 when there are minor changes to the document (e.g. correction after proofreading)

Reference configuration identification

Each reference configuration is defined by:

- An identifier,

- 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.

Configuration Baseline Management

To better monitor and keep control of the project, EXIT6 team established several baselines. Baselines are reference point in the whole development lifecycle that a group of configuration items are formally approved by respective in-charge. Once a baseline is established, any changes made to any SCIs must going through a formal change control procedure.
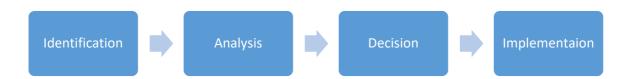
EXIT6 team has identified the following baselines: functional baseline, allocated baseline, developmental baseline and product baseline. The table below shows when each of the baselines identified is established and the group of items included in each baseline:

Baseline	Period of establishment	items
Functional	After requirements review from the first major version	Project Proposal, SRS, Project Plan
Allocated	After design review	Class Diagram, Lo-fi Prototype, Use Case Model
Developmental	After Hi-fi prototype review	Source Code and Manuals
Product	After user acceptance testing	Test Plan, Test cases and coverage report, Quality Plan

Configuration control

Overview

Configuration control deals with managing changes of the SCIs throughout the whole software development lifecycle. This is done to ensure that all changes to the project are approved and acknowledged by EXIT6, preventing any uncontrolled changes. The process of configuration control are as follows:

Identification	→	Analysis	→	Decision	→	Implementaion

- Identification: Identify the need for change

- Analysis: Evaluate change request

- Decision: Approve or disapprove change request

- Implementation: Execute and release change

Any Developer in EXIT6 can propose a change that must be recorded and contains:

- Justification of the proposed change

- Parts of the system which will be affected by the change

- Parts of the code that need to be updated and additional resources if required

All change request will be reviewed by the QA Manager and Lead Developer Manager. Given the approval of the change, respective members working on that part of affected code will handle the changes in a new branch. Test case will be executed to validate the new code. After Validation, the new branch will be combined to the main branch and will contain the latest version of the code. Shall the validation fails, the new branch will be deleted.

Change Management

Developers may make requests or SCRs if one feels the need for change to meet the requirements of the project. The requests can be divided into 2 levels:

Minor: When only parts of the code require a revision

The change should be implemented on a new branch and get tested before combining into the main branch. Formal comments containing details about the change, the developer that made the changes and the branch name should be provided to identify which part of the code is being modified.

Major: When the whole structure / architecture of the system is affected

The proposing developer must fill up a detailed SCR form and submit it to the Project Manager, QA Manager and Release Manager. The request form will go through a complete analysis and investigation on the feasibility of the changes proposed. If the change is approved, the proposing developer will carry out the alteration on a new branch. After the testing validated the changes, the new branch can be combined into the main branch, which will then have the latest version of the code. Formal comments, like the Minor changes, should be provided, clearly stating the change and the reason

Configuration support activities

Configuration Status Accounting

Configuration Status Accounting (CSA) is a set of activities done during the whole software development lifecycle on SCIs. These activities allow all software and relating documentation to be tracked throughout the whole period of the software's lifespan. The set of activities are as follows:

- Recording of information of SCIs

- Storage of recorded information on SCIs

- Maintaining and reporting of status of SCIs

Ultimately, Configuration Status Accounting decides on the following:

- The elements of SCI are to be tracked and reported.

- Types and format of status accounting reports to be generated and its frequency

- The collection, storage and reporting of information

- The access control to all configuration management data

Evolutions traceability

The traceability of modifications of items given their types are as follows:

- **Document**: The version number and revision history section in each document clearly identifies the period and content of modification.

- **Source file:** In-code comments and ReadMe version manual listing the revision history and the modified part of the code.

- **SCR:** Request ID

- **Design Diagrams:** Version Number

- **Test Cases:** Version number

- **VDD:** Version Number

Setting up Configuration status

The Lead Developer Manager sets up the state of all versions and of each configuration article with:

- The label,

- The version number

Configuration status diffusion

The Project Manager, QA Manager and Release Manager will write the VDD in liaison with the development team.

Configuration status records storage

The records are stored in a configuration folder, which contains:

- The requests sorted by record number,

- The software documents,

- The versions of all SCIs,

- The VDD's,

Configuration audits

Two configuration audits will be established to assess the compliance of the configuration management plan:

1. **Baseline Audit**

   Baseline audit ensures the SCI's performance are consistent with the specification mentioned in the SRS. This is done by first identifying the baseline of the latest EXIT6 Trip Planning Application configuration changes. Following which, the latest status of the SCIs will be investigated with the identified baseline.

2. **Functional Audit**

   Functional audit ensures that each SCI of the application has the required physical and functional characteristics, thus showing the consistency between the software and the written specifications. A checklist can be created about the specification to facilitate the functional audit, giving a more accurate conclusion. After which, functional configuration test on the SCIs are conducted and the test results are analyzed with the help of the checklist.

Reviews

Reviews are conducted as a management tool for establishing the baseline. The roles and responsibilities for reviews are as follow:

Role	Responsibility
Lead Developer	- Deciding the objectives and SCIs for review - Generate approval criteria
QA Manager	- Plan schedule for reviews, and decides upon procedures - Record deficiencies and correction method

Configuration management plan maintenance

The SCMP is a reference document for the process of software configuration management. This plan is maintained during the whole software lifecycle on how specific requirements will be carried out. The QA Manager will oversee the whole maintenance of the plan, which includes monitoring and updating the plan on a weekly basis. Any changes to the plan will be analyzed and approved by the QA Manager. After the plan's modification, the SCMP

document will be identified as the latest version through version number, and the history of modification will be specified in the revision history section of the document.