**Software Configuration Management Plan**

**Mall-E.v1**

**TABLE OF CONTENTS**

[**1 Identification**](#_heading=h.gjdgxs) **2**

[1.1 Document overview](#_heading=h.3dhdu0czs49) 3

[1.2 Abbreviations and Glossary](#_heading=h.467g28plocmq) 3

[1.2.1 Abbreviations](#_heading=h.3znysh7) 3

[1.2.2 Glossary](#_heading=h.h5iqgyg46gow) 3

[1.3 References](#_heading=h.po9dpyjjx2q4) 3

[1.3.1 Project References](#_heading=h.3dy6vkm) 3

[1.3.2 Standard and regulatory References](#_heading=h.hrfb8qjelk9x) 3

[1.4 Conventions](#_heading=h.trhc65lbi3ck) 3

[**2 Organization**](#_heading=h.z0vd0kreq1d2) **3**

[2.1 Activities and responsibilities](#_heading=h.5mi82xm921l8) 4

[2.1.1 Decisions process and responsibilities](#_heading=h.q1bv13o9bote) 4

[**3 Configuration identification**](#_heading=h.ro5trky2cdn1) **5**

[3.1 Identification rules](#_heading=h.35nkun2) 5

[3.1.1 Identification rules of configuration items](#_heading=h.1ksv4uv) 5

[3.1.1.1 Identification of a configuration item](#_heading=h.44sinio) 5

[3.1.1.2 Version number of a configuration item](#_heading=h.cjjh3tvg8rm0) 5

[3.1.2 Identification rules of SOUPs](#_heading=h.i78wrl70gnmp) 5

[3.1.2.1 Identification of a SOUP](#_heading=h.1n6kwswukrop) 5

[3.1.2.2 Version number of a SOUP](#_heading=h.tlwusiqpnohf) 5

[3.1.3 Identification rules of documents](#_heading=h.ktcpfbg9bnxp) 6

[3.1.3.1 Description of documents identifiers](#_heading=h.iqvqf1fqi6ii) 6

[3.1.3.2 Definition and evolution of the revision index](#_heading=h.adtfp99emnvx) 6

[3.1.4 Identification rules of a media](#_heading=h.t1wvtnsmevur) 6

[3.1.4.1 Internal identification](#_heading=h.q2rukch0ntaj) 6

[3.2 Reference configuration identification](#_heading=h.onraulo17uew) 6

# 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

This document contains the software configuration management plan of Mall-E.

Assumptions \*due to tight resources we do not have an existing Software Configuration Manager, Quality Assurance Manager will take on the role of the SCM\*

## 1.2 Abbreviations and Glossary

### *1.2.1 Abbreviations*

• PM: Project Manager

• FD: Front-End Developer

• BD: Back-End Developer

• QM: Quality Assurance Manager

• SCM: Software Configuration Manager

• VDD: Version Delivery Description

• SVN: Sub-Version Number

### *1.2.2 Glossary*

• Version: Indicate state of a configuration item at a well-defined point in time

• Baselines: a set of configuration items formally designated and fixed at a specific point in time

• Branch: A line of development that exists independently of another line, yet still share a common history, and can merged in the future

## 1.3 References

### *1.3.1 Project References*

|  |  |  |
| --- | --- | --- |
| # | Document Identifier | Document Title |
| 1 | Project Plan V2.1 | Project Plan |

### *1.3.2 Standard and regulatory References*

|  |  |  |
| --- | --- | --- |
| # | Document Identifier | Document Title |
| 1 | [STD1] | IEEE Standard for Software Configuration Management Plans -- IEEE Std 828 - 2012 |

## 1.4 Conventions

Typographical convention.

Any other convention.

# 2 Organization

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

* The Project Manager
* Front-End Developer
* Back-End Developer
* Quality Assurance Manager

## 2.1 Activities and responsibilities

Described here are the functions required to manage the configuration of the software and responsibilities.

|  |  |
| --- | --- |
| **Activities when setting up the project** | **Person responsible** |
| Identify the configuration items | QM |
| Install the bug repository tool and set up the database | PM |
| Install the software configuration repository tool and set up the database | PM |
| Manage and structure the reference space | PM |
| Define the configuration processes | QM |

|  |  |
| --- | --- |
| **Activities during the project lifecycle** | **Person responsible** |
| Export components for modification, test or delivery | FD |
| Set under control validated components | FD |
| Create version, write version delivery document | BD |
| Approve reference configurations | PM |
| Verify version to be delivered and authorise deliveries | PM |
| Backup spaces | BD |
| Do configuration audits | QM |
| Inspect configuration records | QM |
| Archive reference version | QM |

|  |  |
| --- | --- |
| **Management activities** | **Person responsible** |
| Manage versions and archives | PM, QM |
| Manage configuration records | PM, QM |
| Produce reports and statistics | QM |
| Manage reference space and its access control list | PM, QM |
| Manage spaces backup and archive media | QM |
| Manage quality reports | QM |

### *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 | FD, BD |
| Present a configuration state of the components impacted by the activity | FD |
| Present a documentation state of the components impacted by the activity | BD |

During a configuration management process audit:

|  |  |
| --- | --- |
| **Activities** | **Person Responsible** |
| Do the configuration management process audit | QM |
| Present the records of the configuration management process | QM |
| Present the quality records of the configuration management process | QM |
| Present the records of the documentation management process | QM |

# 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 identification of configuration item is:

* Mall-E\_Vm.n
* where: “Mall-E” is the name of the project
* where: “V” represents version
* where: “m.n.o” represents integers and the version no. of the project. The higher the numbers m.n.o the later the version

#### *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:

* Version numbers m.n.o represents the format: Major.Minor.Patch
* Major version is where the major changes that are incompatible with previous versions are made
* Minor version is where new backwards-compatible functionalities are made
* Patch version is where the backwards-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:

Mall-E\_<document type>\_<document number>\_<revision index>

where:

* <document type> is the type of documents e.g.(Risk Management Plan, Project Plan, Release Plan)
* <document number> is an incremental number, with a separate list for each document type if needed
* <revision index> designates the approval iteration of the document as well as their version number. 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 the following:

* <revision index>: m.n are the representations of the revision index and are integers / versions no.
* m.n refers to MAJOR.MINOR changes
* Major changes would refer to new changes added that at least takes up an entire heading / sub-heading of its own
* Minor changes would refer to the small changes made to improve the readability of the documents

## 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 baselines established are as follows:

* 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

Below are the managing configuration changes and variances in configurations.

## 4.1 Change Management

Below are the process for controlling changes to the baselines and for tracking the implementation of those 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 SVN
* The branch identification is the title of the change where each word is separated by a dash (-), prefixed by the problem
* 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 SVN
* The branch identification is the title of the change where each word is separated by a dash (-), prefixed by the configuration
* Branch content is the related changes

## 4.2 Interface Management

Identify the interfaces to be managed and describe the procedures for identification of interface requirements.

|  |  |
| --- | --- |
| APIs Used |  |
| Best Time API | * To obtain crowd density level of the malls in Singapore |
| Popular Times API | * To obtain the ‘busyness’ of businesses and other informations (etc. name, opening hours, address) |
| Google Maps / Places API | * To obtain functionalities of google map. * To obtain device current location |
| NewsAPI | * To obtain news related to Covid-19 in malls by limiting our queries |
| CCTV API | * To get the counter / number of people, floor by floor crowd density in a mall |

# 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 QM 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 QM writes the VDD.

### *5.1.3 Configuration status diffusion*

The QM and PM 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

Below are peer review audits and formal audits are made to assess the compliance with the Configuration Management Plan.

Audits:

* Baseline Audit: a preliminary assessment to identify and rectify improper procedures and coding practices
* Functional Configuration Audit: to ensure that functional characteristics / requirements of the product are met
* Software Configuration Audit: to ensure that software used meets the baseline needs and configuration control standards

## 5.3 Reviews

1. Weekly configuration reviews will be performed to check on the configuration status and verify if it is up to our standards.
2. Goals of these reviews are to verify all areas of the software creation are being met with the requirements set in place and if all changes are being tracked.

|  |  |
| --- | --- |
| Member Responsible | Roles |
| Project Manager | Come out with the objectives and Software Configuration Items for review |
| Quality Assurance Manager | Planning and deciding a schedule for the review based on procedures |
| Quality Assurance Manager | Come up with procedures to document the deficiencies within the project as well as the solutions to these deficiencies |
| Front-End Developer | Ensure any changes made to the code is reflected in the SVC |
| Back-End Developer | Ensure any changes made to the code is reflected in the SVC |

## 5.4 Configuration management plan maintenance

In the project, the Configuration Management plan acts as a reference and documentation. Throughout the development of the product, the Software Development Life Cycle will be constantly modified. This plan can be used as a ‘log tracker’ for changes applied to identified requirements.

The Configuration Management Plan should be developed by someone who:

* Understands the company’s product, processes, customer and business requirements
* Able to collaborate and communicate across organisational boundaries (etc. talk to Project Management Team and Project Development Team)
* Has experience crafting Configuration Management Plan before

The maintenance of this plan will be led by the Quality Assurance Manager with help from the Front-End Developers and Back-End Developers. The Quality Assurance Manager will then report or seek approval from the Project Manager. The Quality Assurance Manager must also track the processing of pending tasks actions and make sure that the CM Plan is updated to reflect the current status of the project and pending project status.

The latest plan is saved with a newer version number, once approval for the plans have been granted. The configurations folder will store all the old plans, which can be accessed to help facilitate the project schedule.