

**NANYANG  
TECHNOLOGICAL  
UNIVERSITY**  

---

**SINGAPORE**

**School of Computer Science and Engineering**

**Advanced Software Engineering**

**Software Configuration Management Plan**

Website Portal: HouseHunt

Supervisor: Althea Liang

14 October 2021

By Team 5:

Tang Kai Wen, Alvin

Yong Wen Shiuan

Bankata Mishra Spriha

Gupta Suhana

Heng Chor Chen, Sabrina

Heng Fuwei Esmond

*HouseHunt Configuration Management Plan*

## Version History

This table demonstrates information how the development and distribution of the Configuration Management Plan, up to the final point of approval, was controlled and tracked. The table below provides the version number, the author implementing the version, the date of the version, the name of the person approving the version, the date that version was approved, and a brief description of the reason for creating the revised version.

Version #	Implemented by	Revision Date	Approved by	Approval Date	Reason
1.0	Tang Kai Wen, Alvin	07/10/2021	Tang Kai Wen, Alvin	10/10/2021	First Version
1.1	Tang Kai Wen, Alvin	14/10/2021	Tang Kai Wen, Alvin	14/10/2021	Finalize Details that are added

# Table of Contents

Version History	2
1. Identification	4
1.1 Document Overview	4
1.2.2 Glossary	4
1.3 References	5
1.3.1 Project References	5
1.3.2 Standard & Regulatory References	5
1.4 Conventions	6
2. Organization	7
2.1 Activities & Responsibilities	7
2.1.1 Decisions Processes & Accountabilities	9
3. Configuration Identification	10
3.1 Identification Rules	10
3.1.1 Identification Rules of Configuration Items	10
3.1.1.1 Version Number of a Configuration Item	10
3.1.2 Identification Rules of Documents	11
3.1.3 Definition and Evolution of the Revision Index	11
3.1.3 Identification Rules of a Media	12
3.2 Reference Configuration Identification	12
3.3 Configuration Baseline Management	12
4. Configuration Control	14
4.1 Change Management	14
5. Configuration Support Activities	16
5.1 Configuration Status Accounting	16
5.1.1 Evolutions Traceability	16
5.1.2 Setting Up Configuration Status	16
5.1.3 Configuration Status Diffusion	17
5.1.4 Configuration Status Records Storage	17
5.2 Configuration Audits	17
5.3 Reviews	18
5.4 Configuration Management Plan Maintenance	19

# 1. Identification

This document amplifies the “Configuration Management” of the Project Management Plan of HouseHunt.

## 1.1 Document Overview

This document contains the Software Configuration Management Plan of HouseHunt. It seeks to define, reference, and provide the necessary steps and activities that describe how Configuration is performed in the development of HouseHunt.

## 1.2 Abbreviations & Glossary

### 1.2.1 Abbreviations

- **SVN:** Apache Subversion, an open-source versioning and revision control system
- **SCM:** Software Configuration Manager
- **VDD:** Version Description Document
- **SRS:** Software Requirements Specifications

### 1.2.2 Glossary

- **Branch:** A line of development that exists independently of another line, yet still shares a common history, and can be merged in the future.
- **Version:** State of a configuration item at a well-defined point in time
- **Variant:** Versions that coexist

## 1.3 References

### 1.3.1 Project References

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

### 1.3.2 Standard & Regulatory References

#	Document Identifier	Document Title
[STD1]	1	IEEE STD 730-2014, IEEE Standard for Software Quality Assurance Processes
[STD2]	2	ISO/IEC/IEEE 90003:2018 Software Engineering
[STD3]	3	ISO/IEC 25010:2011 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models
[STD4]	4	IEEE 828-2012 - IEEE Standard for Configuration Management in Systems and Software Engineering

## 1.4 Conventions

Typeface	Usage	Example
<b>Bold</b>	Emphasis, headers, titles	<b>Software Configuration Management</b>
<b>Red</b>	Comments and user requests for change to mark changes to the document	I believe this part should be changed with ....
<i>Italic</i>	Minor emphasis, file names.	The chatbot <i>GreetingIntent</i>

## 2. Organization

The software configuration is managed by members of the project, with specific tools.

Responsibilities are shared between:

- The Software Configuration Manager (SCM),
- The Project Manager,
- The Quality Assurance Manager,
- The Lead Developer,
- Release Manager

### 2.1 Activities & Responsibilities

Described here 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
<b>Management activities</b>	<b>Person responsible</b>
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 Processes & Accountabilities

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

In this section, we discuss the rules established to identify items that should be controlled, identification schemes, tools and techniques. We will refer to Software Configuration Items (SCIs) as a component or part in the project. SCIs may be decomposed into further SCIs and can also be modified to create newer versions of the original SCIs.

#### 3.1.1 Identification Rules of Configuration Items

The identification of a SCI is given as:

<b>XXX_Va.b.c</b>
-------------------

- Where XXX represents the Document Name
- V stands for Version
- a.b.c is the version number given with at most one decimal place for major edits (a), minor sub-edits (b) and even more minor fixes (c).

##### 3.1.1.1 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:

- **Major edits call for a new (a) major-version**, for example and not exhaustive to;
  - Adding many new significant sections, functions or features.
  - Removing more than one significant section, function or features
  - Redevelopment of the entire document in a different manner
  - Submitted Document or Revised Document that requires approval from the Project Manager or person of respective authority

- **Minor edits call for a new (b) sub-version** E.g.:
  - Editing of more than one new section, function of features
  - Addition of sentences (1-40 sentences) that do not make a up multiple paragraphs in the same section.
- **Minor fixes call for a new (c) stub-version** E.g.:
  - Minor fixes to bugs
  - Grammatical corrections
  - Formatting numbering or table of contents numbering
  - Spell-check or spelling editing

### 3.1.2 Identification Rules of Documents

The identification of documents is described below:

**XXX\_<document type>\_<document number>\_<revision index>**

Where:

- **Document Type** is:
  - Foo for document type 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, as followed by the identification of the version number in section 3.1.1. Identification of a Configuration Item.

### 3.1.3 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 as similar as section 3.1.1.1 Version Number of a Configuration Item.

### 3.1.3 Identification Rules of a Media

A media is identified as a CD-ROM for submission, a hard disk or thumb drive. In which case it will be identified as per:

<b>&lt;configuration item identification &gt;/&lt;media&gt;/&lt;volume&gt;</b>
--

Where:

- "media" is the media number,
- "volume" is an incremental number to distinguish the media if the delivery contains more than one media.

## 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

The formal definition of “Baseline” is defined as follows: “A Specification or Product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed.” - IEEE

The baselines to be established and that will be used in the project 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.

These baselines are formally designated and fixed upon:

- ✓ Approval from the Project Manager
- ✓ Component/Part has been completed or has been tested at least once
- ✓ QA Team has approved the standards

They can only be changed under a formal change control procedure of which will be documented in the HouseHunt Change Management Plan (See References for location of [RC6] Change Management Plan).

## 4. Configuration Control

The following describes HouseHunt's process for managing configuration changes and variances in configurations.

### 4.1 Change Management

Software change is inevitable for any project. Throughout the software development life cycle phases, change happens whenever there are errors that must be repaired, or the performance or reliability of the system needs to be improved based on the customer's feedback.

Business environment changes, or when new computers and equipment are added to the system also results in software change. Despite these changes, none of the software systems established in our project should break.

The process for controlling changes to the baselines and for tracking the implementation of those changes are listed below:

#### **Error / Bugs Problem Resolution:**

- Change Request Form of Software Errors / Bugs are submitted by any of the users of the application.
- The development team evaluates the Change Request Form and impact analysis will be performed to determine extent of modification.
- When the Change Request Form is valid, the developers will produce and assess the feasibility, cost, and the urgency of the change.
- The Change Request Form is then submitted to the CCB which considers the Change Request Form.
- When the change request is accepted by the CCB, a new branch on the SCM will be created.

#### **Multiple Changes Configuration:**

- Changes requests of configuration files are submitted by any of the users of the application.

- The development team evaluates the Change Request Form and impact analysis will be performed to determine extent of modification.
- When the Change Request Form is valid, the developers will produce and assess the feasibility, cost, and the urgency of the change.
- The Change Request Form is then submitted to the CCB which considers the Change Request Form.
- When the change request is accepted by the CCB, a new branch on the SCM will be created.

## **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 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 also writes the VDD.



### **5.1.3 Configuration Status Diffusion**

The SCM and the Quality Assurance 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**

Software Configuration Management (SCM) audits provides an objective assurance and ensures that the SCM processes are followed. In addition, the purpose of the SCM audits is to maintain integrity of configuration baselines.

Baseline Audit examines and reviews the relevant documents and source code taken at the specific point in time as a basis using configuration management tools. It ensures compliance and that both the documents and code conform to the technical documentation that defines them.

Functional Configuration Audit examines the functional characteristics of HouseHunt and verifies that it has met the requirements stated in the Functional Baseline Documentation approved at the Preliminary Design Review (PDR) and Critical Design Review (CDR).

Software Configuration Audit are ongoing evaluations conducted throughout the software development life cycle to provide management with the information about compliance to SCM policies, plans, processes, and systems, and about the conformance of software product to their requirements and workshop standards.

## 5.3 Reviews

The Software Technical Review has a purpose of delivering a superior version of the product based on qualified personnel (in this case, the Configuration Manager) formal assessment to verify all project components are correctly identified, and documents change have been noted. The Configuration Manager can then provide constructive feedback and possibly better alternatives to the working product.

Configuration reviews will be performed periodically to verify the correctness of configuration status as well as baselining the codes for audits.

According to the **IEEE 1028**, it recommends that the participants of a review may include the following:

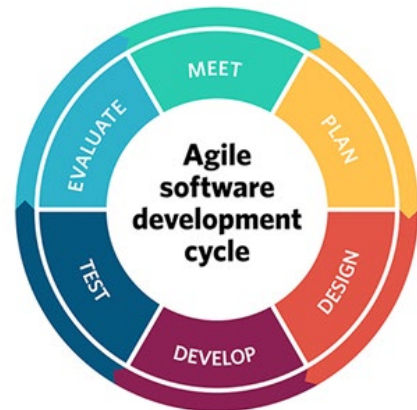
- Decision Maker - Determines if the review objectives have been met.
- Review Leader - Performing administrative tasks relative to the review.
- Recorder - Documents anomalies, decisions, recommendations, and action items.
- Technical Staff - Review and evaluation of the software product.
- Management Staff - Identifying issues that require management resolution.
- User representations - Fill any roles as determined by the Review Leader.

Some participants involved in the review may overlap.

## 5.4 Configuration Management Plan Maintenance

The life cycle of the HouseHunt Project follows the Agile Software Development Cycle as shown in the diagram on the right, with a total of 6 steps officially that will be repeated accordingly when needed.

For the maintenance of the Configuration Management Plan throughout this development life cycle, the **QA team** will be handling the updates to the CM plan. The periodical updates to the plan will take place after **every quarter of the year (when HDB resale data updates) and when necessary.**



Possible CM Plan Maintenance activities that may take place include:

- Revising CM Organizational Structure (Personnel, Resources)
- Revising SCM Activities like Identification Methods, Control, Auditing and Reporting methods.
- Changing the SCM schedules
- Adjusting SCM Resources (Increasing or Decreasing)

All of which will follow the **IEEE 828 Standards** for Software Configuration Management Plans.