

Software Configuration Management Plan of RoomieLah software		
Doc #1	Version: 01	Page 1 / 12



## RoomieLah: Roommate Matching Platform

# *SOFTWARE CONFIGURATION MANAGEMENT PLAN*

### **Submitted to:**

Shi Hanyu  
School of Computer Science and Engineering  
Nanyang Technological University

### **Delivered By:**

StrawHats

Arora Srishti  
Pandey Pratyush Kumar  
Surawar Sanath  
Iyer Rajagopal Mahadevan Acharya Atul  
Tayal Aks  
Agarwal Gopal  
Acharya Atul

Software Configuration Management Plan of RoomieLah software		
Doc #1	Version: 01	Page 2 / 12

## Table of Contents

<b><i>SOFTWARE CONFIGURATION MANAGEMENT PLAN.....</i></b>	<b><i>1</i></b>
<b><i>1 Identification.....</i></b>	<b><i>3</i></b>
1.1 Purpose .....	3
1.2 Objectives.....	3
1.3 Document Overview .....	3
1.4 Abbreviations and Glossary .....	4
1.4.1 Abbreviations .....	4
1.4.2 Glossary .....	4
1.5 References .....	5
1.5.1 Standard and regulatory References.....	5
1.5.2 Conventions.....	5
<b><i>2 Organization.....</i></b>	<b><i>5</i></b>
2.1 Activities and responsibilities.....	5
2.2 Decisions, processes, and responsibilities .....	6
<b><i>3 Configuration identification.....</i></b>	<b><i>6</i></b>
3.1 Identification rules.....	6
3.1.1 Identification rules of configuration items.....	6
3.1.2 Version number of a configuration item .....	7
3.1.3 Identification rules of documents.....	7
3.1.4 Identification rules of a media.....	7
3.2 Reference configuration identification.....	8
3.3 Configuration Baseline Management.....	8
<b><i>4 Configuration control.....</i></b>	<b><i>9</i></b>
4.1 Change Management.....	9
4.2 Interface Management .....	9
4.3 Evolution's control of SOUP items.....	10
<b><i>5 Configuration support activities.....</i></b>	<b><i>10</i></b>
5.1 Configuration Status Accounting.....	10
5.1.1 Evolution's traceability .....	10
5.1.2 Setting up Configuration status .....	10
5.1.3 Configuration status diffusion .....	11
5.1.4 Configuration status records storage.....	11
5.2 Configuration audits .....	11
5.3 Reviews .....	11
5.4 Configuration management plan maintenance .....	11

Software Configuration Management Plan of RoomieLah software		
Doc #1	Version: 01	Page 3 / 12

## Identification

### 1.1 Purpose

This document contains the Software Configuration Management Plan (SCMP) of the software application RoomieLah, developed by the team StrawHats. As a facet of the quality control process of the project, configuration management aims to establish, maintain, and ensure the integrity of the project throughout the Software Development Life Cycle. Identifying and controlling configuration items and changes for the project is also part of what configuration management encompasses. This is done by recording the statuses of these configuration items and reporting their change activity.

### 1.2 Objectives

The objectives of the Software Configuration Management are:

- **Delivery:** Ensures that the correct version of the software is delivered.
- **Traceability:** Ensures that the software being produced, StrawHats, meets the stakeholders' requirements.
- **System Integrity:** Ensures that the software and its supporting materials are consistent throughout the Software Development Lifecycle.

### 1.3 Document Overview

The table below lists the sections that will be documented in this document as well as the contents within each section:

S.No	Section Name	Section Description
1	Identification	Discusses the abbreviations and technical terms used in the project. Descriptions of the terms are provided in detail.
2	Organization	Describes how the team will be organized to perform configuration management. The roles and responsibilities of each team member are mentioned as well.
3	Configuration Identification	Describes how the Software Configuration items will be identified.
4	Configuration Control	The configuration change management process in the Software Configuration items of RoomieLah will be discussed here.
5	Configuration Support Activities.	The activities that evaluate the degree of compliance towards the procedures and standards in implementing changes to the Software Configuration Items (established within the Software Configuration Management Plan) demonstrated by the development team will be described here.

Software Configuration Management Plan of RoomieLah software		
Doc #1	Version: 01	Page 4 / 12

## 1.4 Abbreviations and Glossary

### 1.4.1 Abbreviations

Abbreviation	Description
SCMP	Software Configuration Management Plan
SCI	Software Configuration Item
SCR	Software Change Request
SRS	System Requirements Specification

### 1.4.2 Glossary

Glossary Term	Description
Software Configuration Management (SCM)	The discipline of identifying the configuration of a system for controlling changes to this configuration as well as maintaining the integrity and traceability throughout the Software Development Life Cycle.
Software Configuration Management Plan (SCMP)	A reference document for the SCM process that includes the work breakdown structure and description of each section of the process.

Software Configuration Management Plan of RoomieLah software		
Doc #1	Version: 01	Page 5 / 12

## 1.5 References

### 1.5.1 Standard and regulatory References

#	Document Identifier	Document Title
1	SCM1	IEEE 828-2012 – IEEE Standard for Configuration Management in Systems and Software Engineering
2	SSCM1	IEEE Standard for Software Configuration Management Plans

### 1.5.2 Conventions

Typeface	Usage	Example
Bold	Emphasis	Delivery: Ensures that the correct version of the software is delivered
Highlighted	Special Emphasis	Chris
Red	Comments and user requests for change to mark changes to their document.	This section is unnecessary and should be replaced with...

## 2 Organization

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

- The Project Manager
- Release Engineer
- QA Manager who will assume the role of Software Configuration Manager (SCM)
- QA Engineer

### 2.1 Activities and responsibilities

Describe 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

Activities during the project lifecycle	Person responsible
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

## 2.2 Decisions, processes, 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

The identification of configuration item is:

Give you identification scheme, eg:

Software Configuration Management Plan of RoomieLah software		
Doc #1	Version: 01	Page 7 / 12

- XXX\_Vm.n

where: "Vm.n a" 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:

- Major edits call for a new major version (m)
- Adding or removing significant sections, functions, or features.
- Minor edits call for a new subversion (n)
- Formatting or optimization of code.

### 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:

- Mobile for MOBILE documents,
- Web for WEB documents

" document number " is an 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.

#### 3.1.3.2 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:

- Major edits call for a new major version (m)
- Significant changes made to the documentation. Redevelopment of the content.
- Revision index increments by 1.0.
- Minor edits call for a new subversion (n)
- Formatting the document or correcting spelling/grammatical mistakes.
- Revision index increments by 0.1. It resets to 0 in case of a major edit.

### 3.1.4 Identification rules of a media

#### 3.1.4.1 Internal identification

The identification of a media is described below:

<configuration item identification >/<media>/<volume>

where:

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

Software Configuration Management Plan of RoomieLah software		
Doc #1	Version: 01	Page 8 / 12

### 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

This section describes what baselines are to be established. It also explains when and how they will be defined and controlled.

3.3.1 In configuration management, a baseline is an agreed description of the attributes of a product, at a point in time, which serves as a basis for defining change. A change is a movement from this baseline state to a next state. The central purpose of baseline identification is to help identify significant changes from the baseline state.

3.3.2 Generally, a baseline may be a single work product, or set of work products that can be used as a logical basis for comparison. A baseline may also be established as the basis for subsequent select activities when the work products meet certain criteria. Such activities may be attributed with formal approval.

3.3.3 Conversely, the configuration of a project often includes one or more baselines, the status of the configuration, and any metrics collected. The current configuration refers to the status, current audit and/or current metrics. Similarly, but less frequently, a baseline may refer to all items associated with a specific project. This may include all revisions of all items, or only the latest revision of all items in the project, depending upon context.

3.3.4 While marking approval status covers the majority of uses for a baseline, baselines may also be established to signify the progress of work through the passage of time. In this case, a baseline is a visible measure through an endured collective effort, e.g. an operational baseline.

3.3.5 Baselines themselves are valued not only to use to identify the notable state of work product(s) but also provide historical views of how work product elements have proceeded together over time. When a historical baseline is retrieved, the state of the work product(s) in that subset share the same significance in their history of changes; that allows project leaders to compare the relative progress of single parts of a project to the project as whole, which allows project leaders to identify individual items that lag or lead in progress toward better functionality or performance.

3.3.6 For this reason, baseline identification, monitoring, and retrieval are critical to the success of configuration management. Once retrieved, the baseline may be compared to a particular configuration or another baseline.

3.3.7 Most baselines are established at a fixed point in time and serve to continue to reference that point (identification of state). However, some baselines are established to carry forward as a reference to the item itself regardless of any changes to the item. These



Software Configuration Management Plan of RoomieLah software		
Doc #1	Version: 01	Page 9 / 12

latter baselines evolve with the progression of the work effort but continue to identify notable work products in the project.

Examples of baselines:

- **Functional baseline (FBL)** - Baseline to describe the system functional characteristics of the system or the system specifications. It should document the system's functionality and overall minimum performance.
- **Allocated baseline (ABL)** - Baseline to describe the design of the functional and interface characteristics. The performance of each configuration item in this baseline is described in its preliminary design specification.
- **Product baseline (PBL)** - Baseline which consists of completed and accepted system components and documentation that identifies these products.
- **Operational Baseline (OBL)** - Baseline for the state of work products during and after deployment.

Any other baseline based on proprietary business practices may be used. To define or control these baselines, the respective team member will have to take approval from the Project Manager of the team. It should also align with the standards set by the Quality Assurance team.

## 4 Configuration control

Most features will not undergo changes unless a performance bottleneck or a bug is discovered, However RoomieLah's system can undergo changes with respect to integration with tertiary institutions based on their changing rules of dormitories (for example, not allowing in-dorm to stay for certain groups of individuals viz. final year students). The standard operating procedure will then involve ensuring the proper functioning of the application after any change is made.

### 4.1 Change Management

Software products inevitably go through changes. This could either be due to changing user demands or due to errors and bugs. No matter the change, none of the software components must break. Changes requests are emitted from by the project manager according to the problem resolution process.

If there were to be a change, the steps to be undertaken for the same for RoomieLah would be:

1. After a change request has been created and logged by the change manager, the change request's status is updated.
2. When a change request is accepted by the project manager/product manager, a branch is created in the SCM.
3. The branch identification is done as per the configuration item identification protocol.
4. The branch content will then have the required changes which will then go through the rest of the change management workflow.
5. After implementation of the change, the CR will be reviewed and closed accordingly.

### 4.2 Interface Management

Optional, use it only if you have interfaces with 3rd parties

Software Configuration Management Plan of RoomieLah software		
Doc #1	Version: 01	Page 10 / 12

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

#### 4.3 Evolution's control of SOUP items

Explain how you manage evolutions of SOUP.

A simple solution is to freeze SOUPS at the beginning of the project, not always feasible.

## 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 Evolution's 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 Version Description Document (VDD), which is described as follows:

- It is the primary configuration control document used to track and control versions of software to be released to the operational environment.
- The VDD is a summary of the features and contents for the software build, and it identifies and describes the version of the software being delivered to the State, including all changes to the software since the last VDD was issued.
- Every unique release of the software (including the initial release) shall be described by a VDD. If multiple forms of the software are released at approximately the same time (such as, to different sites) each must have a unique version number and a VDD.

Software Configuration Management Plan of RoomieLah software		
Doc #1	Version: 01	Page 11 / 12

The VDD is part of the software CI product baseline. When distributed, the version description document should be sent with a cover memo that summarizes, on a single page, the significant changes that are included in the release.

The label and the version help uniquely identify each VDD and trace it accordingly. The attached memo with the version description document will serve as an executive summary for the details found in the attached VDD. The treatment should be titled, on the cover memo, as Summary of Changes.

### 5.1.3 Configuration status diffusion

The SCM and the quality manager write the VDD (Version Description Document)...

### 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 this project, the baseline audits form the largest part of the Configuration Audits. It entails establishing the baseline that decides the attributes and the description of the software application.

The latest status of the SCIs is examined in order to ensure that their performance aligns to the specifications mentioned in the SRS. The formal examination of the functional requirements will be conducted by the QA Manager and the QA Engineer to ensure it meets the requirements specified.

A standardized checklist can be used during the functional configuration audit to keep a history and promote learning.

## 5.3 Reviews

Describe how the technical reviews during the project relate to the establishment of baselines, branches and so on, and explain the role of configuration managers in these reviews.

An independent review of the existing configuration plan is required which will be covered by the Configuration Management Plan Review (CMPR). It will evaluate the configuration management plan on five core functions, which are planning, identification, status accounting, change control and verification and audits.

The reviews will ensure that the product that is delivered meets all the baseline requirements.

## 5.4 Configuration management plan maintenance

The software configuration serves the purpose of being the reference and documentation for the process of Software Configuration Management. The document will follow the IEEE 828 Standards for Software Configuration Management Plans.

Software Configuration Management Plan of RoomieLah software		
Doc #1	Version: 01	Page 12 / 12

This document shall be regularly updated every week by the QA Engineer and the QA Manager to successfully define how changes applied to the identified requirements will be performed.

The SCM will oversee assessing and approving all changes to the plan. The latest plan must be saved with a newer version number, once the approval has been granted by the SCM. Past versions will be stored to help that the project is going according to the schedule.