Foodify - a food management web application
[SOFTWARE CONFIGURATION MANAGEMENT PLAN]
Version 1.0

Revision History

Version	Date	Author	Description of Changes	Approved by
1.1	22/08/2022	Chua Jim Sean	Initial Template	Anandarajan Sindini
1.2	23/09/2022	Chua Jim Sean	Add Identification, Organization	Anandarajan Sindini
1.3	25/10/2022	Chua Jim Sean	Add Configuration Identification, Control, Configuration Support Activity	Anandarajan Sindini

TABLE OF CONTENTS

1 Identification

This document amplifies the Configuration management Plan of the Foodify Application of Runtime Terror.

1.1 Document overview

This document contains the software configuration management plan of software Foodify.

1.2 Abbreviations and Glossary

1.2.1 Abbreviations

● SCM: Software Configuration Manager

● SVN: Apache Subversion, an open source versioning and revision control system.

● SRS: Software requirement specification

● CI: Configuration Item

● SOUP: Software of Unknown Provenance

● VDD: version description document

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.

1.2.3 Standard and regulatory References

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

#	Document Identifier	Document Title
2	SSCM1	IEEE Standard for Software Configuration Management Plans

1.3   Conventions

#	Purpose	Example
Bold	Emphasis, headers, Titles	**Software Configuration Management**
Italics	Minor emphasis, file names.	Source File: *Home.js*

2   Organization

- SCM: Jim Sean

- Project Manager: Nandini Verma

- Technical Manager: Yen Duong

- Lead Developer: Anamika Kolady

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
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

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

2.1.1 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	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:

 XXX_Vm.n

where: "Vm.n a" is the version of the

configuration item.

3.1.1.1 *Identification of a configuration item*

- Code
- o Source code
- Documentation
- o Project proposal
- o System requirement Specification
- o Quality plan
- o Project plan
- o Risk management
- o Test plan
- System designs

- o Use case models

- o Conceptual models

- o Software architecture

- o Software prototype

- o Unit test

- o Coverage test

- Specification

- o Project requirements specification

o   Unit test

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:

•   The version number is of the form MAJOR.MINOR.PATCH. The version must be incremented when there's any changes.

•   Major releases:

o   Major change to business logic

▪   Affect how personal data is stored

▪   Removal of a major function

o   Adding new API which involves changes to user interface

•   Minor releases:

o   Changes to existing functions which does not change the user interface

•   PATCH:

o   Compatibility issues

o   Grammatical issues

o   Improvement of performance

3.1.2 Identification rules of SOUPs

3.1.2.1   Identification of a SOUP

Use you own ID like above or take ID of SOUP manufacturer

3.1.2.2   Version number of a SOUP

Use you own ID like above or take ID of SOUP manufacturer

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:

● "document type" is the type of document (quality plan, srs, …).

● "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.

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.

3.1.4 Identification rules of a media

A media is for example a tape, a CDROM.

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.

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

Describe what baselines are to be established. Explain when and how they will be defined and controlled.

Examples of baselines :

• 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

Describe your process for managing configuration changes and variances in configurations.

4.1 Change Management

The process for controlling changes to the baselines and for tracking the implementation of those changes are shown below.

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 underscore (-), prefixed by problem

• Branch content contains the applied 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 underscore (-), prefixed by config

• Branch content contains the applied changes.

• Branch content is …

4.2 Interface Management

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

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

4.3 Evolutions 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 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 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 are sorted chronologically.

5.2   Configuration audits

- baseline audit

- functional configuration audit

- software configuration audit.

5.3   Reviews

Technical reviews during the project are related to the establishment of baselines and branches. The SCM plays an important role in these reviews as SCM needs to help the teams to maintain consistency and control over what is produced.

- After establishment of the baseline, changes to the SCI can only be made under a formal change control procedure.

- Configuration reviews will be carried out periodically to verify the correctness of the configuration status

- Purpose of the configuration review is to make sure all the changes are recorded and that all project components are correctly identified, and document change has been noted.

5.4   Configuration management plan maintenance

The QA team will be in charge of the configuration management plan maintenance and responsible to update the CM plan and verify components' correctness. The maintenance will be carried out every 2 weeks.