Configuration Management Plan for TimeWise

Team 0
Lab group : SSP7

Group members	Roles
Mantri Raghav	Project Manager
Dwivedee Lakshyajeet	Development Lead
Harding James	Back-End Developer
Alex Bernini	Front-End Developer
Mittal Madhav	Release Manager
Xue Xueting	QA Engineer
Koh Hui Ling	QA Engineer
Lek Jie Ling	QA Manager

Version History

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Xue Xueting	7 April 2020	Lakshyajeet	8 April 2020	Completed Identification
1.1	Koh Hui Ling	7 April 2020	Lakshyajeet	8 April 2020	Completed till Configuration Control
1.2	Harding James Sebastian	7 April 2020	Lakshyajeet	8 April 2020	Completed Document

Table of Contents

Identification

Document Overview

This document is the Configuration Management Plan of TimeWise. This document will define and provide the necessary steps for configuration management for the project.

Abbreviations and Glossary

1. **Abbreviations**
   - **SCM**: Software Configuration Manager
   - **SVN**: Apache Subversion
   - **SRS**: Software Requirements Specifications
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

Standard and Regulatory References

#	Document Identifier	Document Title
[STD1]	1	IEEE 828-2012 - IEEE Standard for Configuration Management in Systems and Software Engineering

Conventions

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

Organization

The software configuration is managed by the members in the team of this project, whereby the responsibilities are shared between
- The Software Configuration Manager (SCM)
- The Project Manager
- The Quality Manager
- The Lead Developer

Activities 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

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

Decision Process and 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

Configuration Identification

Identification Rules

1. **Identification rules of configuration items**
    a. **Identification of a configuration item**
       The identification of configuration item is
       XXX_Va.b
       Where:
          - "Va.b" is the version of the configuration item
    b. **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 (a)
              - Adding or removing significant sections, functions or features
              - Redevelopment of document
          - Minor edits call for a new subversion (b)
              - Editing significant sections, functions or features
2. **Identification rules of documents**
    a. **Description of documents identifiers**
       The identification of documents is described below:
       XXX_<document type>_<document number>_<revision index>
       Where:
          - "document type" is Foo for FOO documents and BAR for bar 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 and so on
    b. **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 similar to the ones in Version number of a configuration item.

3. **Identification rules of a media**
    a. **Internal identification**
       A media can be identified as either a thumbdrive, hard disk or CDROM for submission. It will be identified as:
       <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

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.

Configuration Baseline Management

The baselines to be established are:
- Functional baseline (FBL) describes the system functional characteristics
- Allocated baseline (ABL) describes the design of the functional and interface characteristics
- Product baseline (PBL) consists of completed and accepted system components and documentation that identifies these products

The baselines will be defined and controlled by:
- Having gained approval from the Project Manager of the team
- The quality assurance team has approved the standards
- The project has been completed or else has at least been tested once

Configuration Control

Most features will not have to be changed unless a bug is found or performance is not optimal. However, the web scraper in particular may require heavy changes based on the changing environment, both business and technological changes. Our process will involve ensuring the proper functioning of the application after any change is made.

Change Management

Changes are bound to occur in complex software projects. 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. The process for controlling the changes to the baselines and for tracking the implementation of these changes is as follows:

Problem resolution for software errors:
- Change Request Form containing details of Software Errors are submitted by users.
- The development team reviews the Change Request Form and evaluates the impact and extent of the modification.
- When the Change Request Form is deemed to be valid, the developers will assess the cost, feasibility and priority of the change.
- The CCB then reviews the change and decides whether to implement it or not.
- If the change is accepted then the change is implemented in a new branch of the system.
- Note that the change form is updated at each step documenting comments and proceedings from all parties involved.

Example for multiple configuration:
- Change Request Form containing details of Software Errors are submitted by users.
- The development team reviews the Change Request Form and evaluates the impact and extent of the modification.
- When the Change Request Form is deemed to be valid, the developers will assess the cost, feasibility and priority of the change.
- The CCB then reviews the change and decides whether to implement it or not.
- If the change is accepted then the change is implemented in a new branch of the system.
- Note that the change form is updated at each step documenting comments and proceedings from all parties involved.

Interface Management

This section Identifies the interfaces to be managed and describes the procedures for identification of interface requirements.
The web scraper needs to scrap a users timetable from the university website and the only way that it can do that is by knowing the format of both the website and the timetable. This means that if the university updates their website or the format of the timetable the scraper may also have to be updated accordingly. The scraper requires the application to send a post request containing the username and password to it and will respond with all the timetable events for that particular user, assuming the username and password is correct.

Configuration Support Activities

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.

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.

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.

3. ## Configuration status diffusion
   The SCM and the Quality Assurance Manager write the VDD

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.

Configuration Audits

SCM audits help to ensure that all the SCM processes are followed, it also helps in ensuring the integrity of the configuration baselines.

A Peer Review audit is a formal review where a person or a group of people examined the product (document, code, or other) in detail, so as to help in evaluating its technical content and product quality.

Formal audits are conducted at predetermined points. This audit is to ensure that the design, development and integration of Timewise meet the technical requirement, and make sure that everything is documented accurately with no unauthorized changes.

Reviews

The technical reviews during the project helps to ensure that a qualified product will be delivered. The Configuration Manager will verify and make sure all project components are identified correctly, and changes made to the documents are also noted down. These reviews will be performed periodically to check on the correctness of configuration status and to baselining the codes for audits.
The Configuration Manager will give constructive feedback on the working product.

Configuration Management Plan Maintenance

The life cycle of the Timewise will follow the Agile Software Development Cycle.
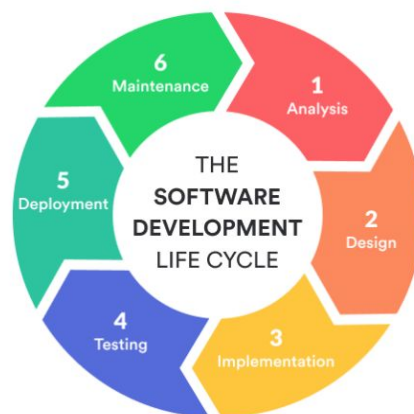
A total of 6 steps in the development cycle, which Timewise will follow an incremental and iterative approach, repeated when needed.

The QA team will be responsible in handling all the updates and maintenance on the CM plan throughout this development lifecycle. The updates to the CM plan will take place every 3 months or whenever 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)

The Configuration Manager will be responsible for the periodic updates to this plan and on all of the above activities.

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