Software Configuration Management Plan for Illo

# Introduction

# Purpose

The purpose of this Software Configuration Management Plan is to outline the structure of the change management process which is followed throughout the development of our mobile app, Illo. An Android application that is focused on the improvement of health and fitness through the use of a Pomodoro Timer.

By following this plan, we hope to ensure the efficient management of our applications development, reduce delays, and improve the overall quality of Illo. This plan will address and define the associated activities, the coordination between activity development, the required tools and resources for this plan, as well as how it will be maintained moving forward.

# Scope

This plan covers the development of the Illo application. The scope of it includes the organization and management of team members, the listing of all activities to be performed, the coordination of multiple activities, identification of tools and human resources required, and the process for maintaining this plan while it is in effect.

# Key Terms

Pomodoro Technique – A time management method which breaks work into intervals that are separated by short breaks.

Configuration (Version) Control – The process of ensuring that versions of systems and components are recorded and maintained, so that changes are managed, and all versions of components are identified and stored for the lifetime of the system.

Configuration Item – Anything associated with the software project that has been placed under configuration control. Items have a unique name and often have different versions.

Version – An instance of a configuration item that differs from other instances of that item and has a unique identifier.

# References

The Wikipedia page for the Pomodoro Technique provides a detailed description of this time management method and its correct execution - <https://en.wikipedia.org/wiki/Pomodoro_Technique>

# SCM Management

# Organization

The development of Illo consists of a product owner, a scrum master, and three members of the software development team. The product owner manages the applications’ requirements and priorities. The scrum master oversees and facilitates the Scrum development process. The development team implements, tests, and delivers the requirements.

# Responsibilities

* Product Owner
  + The purpose of the product owner is to decide the goal of each sprint and to determine what is acceptable as “done” in the sprint. They have the authority to cancel a sprint should they decide that the goal of it has become obsolete, and they are responsible for the following tasks:
  + Updating the status and order of items in the product backlog.
  + Updating items in the sprint backlog with “done” or “not done”.
  + Creating required documents.
* Scrum Master
  + The purpose of the scrum master is to facilitate the product owner’s and development team’s work. They ensure the development team is practicing Scum properly, and are responsible for the following tasks:
  + Interfacing with external entities
  + Creating required documents.
* Development Team
  + The purpose of the development team is to communicate with the product owner to determine what “done” is, and to determine the work that needs to be completed in a sprint. They manage the development of the software, and are responsible for the following tasks:
  + Implementing user stories in the sprint.
  + Managing the sprint backlog.
  + Creating required documents.

# 

# Applicable Policies, Directives, and Procedures

This project is subject to the following external constraint(s):

* To be submitted for COS-420: Intro to Software Engineering, the project’s development is limited to the timeframe of one semester.

# SCM Activities (Ideally you will have subsections 3.1.1, 3.1.2, …)

# Configuration Identification

* + 1. GitHub will be Illo’s primary library of configuration documents and artifacts.
    2. Configuration documents include the SRS document, the Backlog, the Kanban board, Use Case document, and the Illo software.
    3. Changes to configuration documents will be recorded via GitHub history.

# Configuration Control

All changes to configuration will be handled through GitHib. A developer may create a branch and implement what they want. Final products will be reviewed by the group before merging to the main branch.

# Configuration Status Accounting

Versions of Configuration documents will be tracked via version numbers and GitHub history

# Configuration Evaluation and Reviews

* + 1. Audits will be conducted when milestones are hit.
    2. Configuration Management plan will be reviewed during audits.
    3. Mistakes and discrepancies will be documented.
    4. Corrective measures will be taken to fix discrepancies.

# Interface Control

* + 1. Interfaces will be defined and documented.
    2. Requirements for the interfaces and interactions between them will be discussed and agree upon by group.

# Subcontractor/Vendor Control

Items developed outside the scope of Illo will be discussed and reviewed before being implemented.

# Release Management and Delivery

* + 1. New releases will be made at the discretion of the developers.
    2. Post-release reviews will be conducted to identify mistakes or opportunities.

# SCM Schedules

# Sequence and coordination of SCM activities

4.1.1 Same as deliverable submission schedule

# Relationship of key SCM activities to project milestones or events, such as:

* + 1. Audits are had after milestones are hit.

# Schedule either as absolute dates, relative to SCM or project milestones or as sequence of events.

* + 1. Same as deliverable submission schedule .

# Graphical representations can be used here.

# SCM Resources

# Identifies environment, infrastructure, software tools, techniques, equipment, personnel, and training.

* + 1. GitHub, Android Studio, Design Patterns listed in Design patterns document, Gradle, Java, Computer Science Students.

# Key factors for infrastructure:

5.2.1 Everything at discretion of the developers.

# Identify which tools are used in which activity.

5.3.1 GitHub for version management. Android Studio, Gradle, Java, and more for development.

# SCM Plan Maintenance

# Who is responsible for monitoring the plan?

The Scrum Master will be responsible for monitoring this plan as a part of their duty in ensuring that the development team practices Scrum. This will ensure that the plan is being followed correctly.

# How frequently updates are to be performed?

Every two weeks this plan will be reviewed and updated by the Scrum Master, if necessary.

# How changes to the Plan are to be evaluated and approved?

The Product Owner will be responsible for evaluating and approving any changes to the plan as a part of their duty. This will ensure that any proposed changes align with set goals.

# How changes to the Plan are to be made and communicated?

Any suggested changes to the plan will be made during group meetings. Following the Product Owner evaluating all suggested changes, should any be approved, the Scrum Master will update the plan and be responsible for communicating the changes to the development team via Discord.

# History of changes made to the plan.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Changes** | **Version** |
| Riley Mills | 23 Apr | Formatted document & filled in sections: 1, 2, and 6. | 1.0 |
| Cole Adams | 4/24/23 | Filled in sections 4, 5, and 6. | 2.0 |
|  |  |  |  |