Table of Contents

1. Introduction

1.1. Purpose

The purpose of this document is to give the reader an understanding of the practices that the organization Wombat Workshop follows while developing the mobile application Dream Catcher.

1.2. Scope

This Configuration Management document is exclusively looking at the development of the mobile application Dream Catcher. This includes the code, other documentation, database systems, as well as any other recourse directly related to the development process.

1.3. Key Terms

Bucket: Key component of this application that has a descriptive name and contains items that users wish to complete.
Dream Catcher: The name of the application that is being developed
Item: A goal/event that a user wishes to complete

1.4. References

N/A

2. SCM Management

2.1. Organization

This Configuration Management document is intended to be used by the organization Wombat Workshop. Wombat Workshop is composed of 5 UMaine students that are developing the application for the software engineering course COS 420. The organization has and will continue to develop the application according to this document to the best of their ability.

2.2. Responsibilities

- Product Owner
    - Manages the product backlog.
    - Orders the items in the product backlog.
    - Makes the product backlog visible to all.
    - Determines what is “done” and is acceptable in the sprint.
    - Creates the required documents.
    - May cancel a sprint if the sprint goal becomes obsolete.
- Scrum Master
    - Makes sure the development team is practicing in Scrum properly.
    - Facilitates the product owner’s and development team’s work.
    - Creates the required documents.
    - Interfaces with external entities.
- Development Team
    - Determine the work needed to be done in a sprint.

- Work with the product owner to determine what “done” is.
- Create the required documents.
- Implement the user stories in the sprint.
- Manage the sprint backlog.

2.3. Applicable Policies, Directives, and Procedures

Due to the nature of the application, the organization is not restrained by any external source. As the development of the application continues, there may be some policies that need to be adhered to. In that situation this document will be updated and those policies will be located here.

3. SCM Activities

3.1. Configuration Identification

3.1.1 Configuration items are identified through the following:

- User Stories
- User Requirements
- System Requirements
- Functional/Non-Functional Requirements
- Scrum Discussions

3.1.2 Name configuration items (unique identifiers)

Naming conventions of configuration items consists of a naming of the item appropriate for its purpose, followed by its version number. Words shall be separated with underscores rather than spaces, and the item name shall end with the file type.

3.1.3 Acquiring configuration items (physical procedures)

Configuration files can be stored and acquired via the team Wombat Workshop GitHub repository, with access limited to Wombat Workshop team members. The link for the GitHub repository is https://github.com/ElijahStory/COS420-App.

3.2. Configuration Control

3.2.1 Requesting changes

Any requests for changes can be established during scrum meetings, within the team discord channel, or via submitting an issue to the team GitHub repository

3.2.2 Evaluating changes

The evaluation of changes will occur during scrum meetings or within the team discord channel.

3.2.3 Approving or disapproving changes

The approval or disapproval of any requested changes may occur during scrum meetings or within the team discord channel. Any disagreements on the approval or disapproval of a change may be resolved by a vote among all Wombat Workshop team members.

3.2.4 Implementing changes

Changes that are to be implemented will be added to the backlog and be given a priority depending on factors such as time to implement and significance to meeting product requirements. Changes with the highest priority shall be developed and pushed to the GitHub repository by the end of each sprint.

3.3. Configuration Status Accounting

3.3.1 Metrics to be tracked and reported and type of report.

- Commits to the GitHub repository
- Implementation progress tracked via sprint backlog
- Estimated time to complete tracked via sprint backlog
- Actual time to complete tracked via sprint backlog

3.3.2 Storage and access control of status data.

Access is limited to Wombat Workshop team members.

3.4. Configuration Evaluation and Reviews

Evaluations and reviews shall be conducted by at least one team member of Wombat Workshop prior to the release of a CI. The evaluation process of a CI will include testing for bugs/functionality, and a brief code review to help maintain best practices. The purpose of this evaluation is to minimize oversight Dream Catcher releases.

3.5. Interface Control

Coordination of changes to CIs with changes to interfacing items outside of the scope of the plan will be assessed as they occur by developers. Any significant changes that affect product functionality will be marked as high priority and immediately addressed by the development team.

3.6. Subcontractor/Vendor Control

Incorporation of items developed outside the project environment into the project CIs will undergo the same auditing process defined in section 3.4.

3.7. Release Management and Delivery

Prior to a release, builds are to be tested for full functionality of added changes to the previously released build. Once adequate testing has been completed, a time of release may be established at the discretion of Wombat Workshop team members.

4. SCM Schedules

4.1. Sequence and coordination of SCM activities

Coordination of SCM activities are logged in our Sprint backlog which is updated weekly to track our progress with the activities.

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

Establishment of configuration baselines occurs at the beginning of each sprint. Implementation of any changes occurs continuously throughout each sprint once configuration baselines have been established. Configuration audits occur once implementation of an item is complete, and continue until the end of each sprint or until the CI has been adequately audited.

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

Each development cycle schedule consists of a sequence of events/tasks described in the following list:

- Establish features to be implemented
- Identify necessary changes/updates/additions to documentation
- Hold scrum meeting
- Allocate tasks amongst team members
- Implement changes/features
- Test all implemented changes/features
- Update/change/add necessary documentation
- Final revisions and/or discussions
- Push deliverable to GitHub

4.4. Graphical representations can be used here.

Not sure if required

5. SCM Resources

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

We are creating this application in Android Studio and we are using Java. We are using many different software techniques to create this application, our team has engaged in techniques such as scrums, sprints, kanban boards, and more. We have been using Github for our version control in this application.

5.2. Key factors for infrastructure:

- Android Studio was chosen for creating this application since it is the industry standard.
- Github is used for version control due to the ease of use and also being an industry standard.
- Java was used because it can be used efficiently to create the application that is desired.

5.3. Identify which tools are used in which activity.

Development Tools: Android Studio
Version Control Tools: GitHub,
Documentation Tools: Google Docs, Google Sheets
Communication Tools: Discord

6. SCM Plan Maintenance

6.1. Who is responsible for monitoring the plan?

The scrum master is responsible for monitoring the plan to ensure that the schedule is being followed.

6.2. How frequently updates are to be performed?

Updates should be performed weekly.

6.3. How changes to the Plan are to be evaluated and approved?

Changes to the plan should be discussed with all members of the development team and should be approved by the product owner.

6.4. How changes to the Plan are to be made and communicated?

Changes to the plan can be made by members of the development team as deemed necessary through the use of our kanban board. These can then be communicated to other members via the discord server that is being used by the team.

6.5. Also includes history of changes made to the plan.

Version History	Title	Date
1.0	1st Release	4/3/2022