Configuration Management Plan (CMP)

for

Acies

Created by Sam Waggoner, Dylan Haughton, Will Cunningham,
Callen Shaeffer, and Michael Wilkinson

[Table of Contents]

1. Introduction

This document describes the configuration management that will occur during the creation and maintenance of the Acies desktop music application. Acies is a sound game in which users draw lines on a grid and adjust pitch, volume, and compression in order to create music.
The Institute of Electrical and Electronics Engineers (IEEE) Standard 828-2005, "IEEE Standard for Software Configuration Management Plans", was used as a guide for the document's organization and content.

1.1. Purpose

The purpose of this document is to define the activities that will take place for changes surrounding the Acies application. This document will lay out the version control standards for the application, as well as the process for change requests from both users and developers. It will also discuss system building and release management procedures.

1.2. Scope

This Configuration Management Plan (CMP) applies to our codebase and documentation repository on Github. Thus, this document encompasses the software, documentation, presentations, and all other Acies-related material created or used by the Acies developer team.

1.3. Key Terms

a. Artifacts

Source code
Software Requirements Specification Document
Configuration Management Plan Document
Sprint Backlogs
Product Backlog
User Stories Document
Project Description Document
Team Member Report Document
Sprint Review Documents
Use Case Model Document
Sequence Diagram Document
Domain Model Document
Software Architecture Document
Design Model and Design Pattern Document

1.4. References

Visual Studio Code - IDE, code development application
Github - Version Control System, local and cloud-based
PyGame - Event handler and graphics library
PyGame_GUI - UI/UX tooling library
Google Drive - Team Documentation Storage, website

Discord - Team Communication, application
No ambiguous external references are mentioned outside of the References or Artifacts above.

2. SCM Management

2.1. Organization

Everyone involved in the project is a member of the development team. There is no single person who is the organizer for SCM activities.

2.2. Responsibilities

a. Roles:

Developers:

Sam Waggoner, William Cunningham, Michael Wilkinson, Callen Shaeffer, Dylan Haughton

Scrum Masters:

The Scrum Master varies by deliverable. It is decided every two weeks, and can be any of the developers as listed above.

b. Purposes:

Developers:

The role of the developer is to help build the application by contributing to comprehensive source code, and assist with associated documentation. Associated documentation includes all of the **Artifacts** in **Key Terms** section 1.3.

Scrum Masters:

The role of the scrum master is to encourage that scrum practices are in use. Namely, the Scrum Master may facilitate communication between members of the team, organize meetings, assist in distributing workload, and contribute to solving blockers.

c. Memberships and Affiliations:

The developer team is the only structure within our team, so there are no other affiliations or memberships.

d. Period of Effectivity:

These roles, expectations, and memberships are effective for the duration of the development of Acies, which is Tuesday January 18 2022 through Friday May 06 2022.

e. Scope of Authority:

Developers:

Developers have equal say in matters of decisions about the formation of the application, whether it be features, implementations, or documentation. Disagreements are decided by majority vote.

Scrum Masters:

As developers have equal say, there are no authoritative roles in the team other than the Scrum Master. The Scrum Master has authority to suggest discussion direction within the development team, and has no authority outside of that scope.

f. Operational Procedures:

Developers:

Developers shall distribute responsibility before each deliverable in order to finish the scheduled work. Developers shall contribute their fair share. Tasks do not have to be distributed equally among all team members; for example Team Member 1 may add a feature while Team Member 2 may work on fixing feedback. Developers must communicate and show up to meetings.

Scrum Masters:

Scrum Masters will be arbitrarily assigned for each deliverable. Scrum Masters will attend meetings and enact duties as detailed in the Purposes section above.

2.3. Applicable Policies, Directives, and Procedures

This document follows IEEE Standard 828 – 2005.

3. SCM Activities

3.1. Configuration Identification

a. Acquiring Configuration Items

Will be obtained through:

- Analysis of User Stories
- Analysis of Deliverable tasks

b. Anatomy of Configuration Items

Will consist of :

- An assigned number
- A description of the CI
- An assigned name

i.e. **256, Glacial:** Glacial is a document that outlines the current status of the baseline.

3.2. Configuration Control

As the Acies team is small and each person is involved in several processes, the required steps for requesting, evaluating, approving/rejecting, and implementing changes will be minimal. To request changes, a team member should bring the idea up in a deliverable meeting. The other

members will then evaluate and discuss the idea. If the decision merits a longer evaluation time, then that time will be granted. If the decision can be made on-the-spot, then the plan to implement the changes can be discussed at that time. To implement the change, the idea can be added to the applicable documentation, then selected for a sprint.

3.3. Configuration Status Accounting

a. Baselines

The product submitted at the end of each sprint will be a functional product, though it may be unfinished. Thus, the submission for each deliverable contains a baseline that includes finished documentation and working code.

b. Metrics and Storage

The progress of work done on each deliverable is not tracked until the very end. This is the case because with such a small group, intermediate status can be attained by simply checking in with the other team members or viewing the work done and the work left to do. For reports on our development process deliverable-by-deliverable, one can refer to the Product Backlog, Sprint Backlog, and Sprint Review.

3.4. Configuration Evaluation and Reviews

Evaluations and reviews will be done during scrum meetings, where all developers will be participating. The Scrum Master will lead the evaluation and reviews. These evaluations will uncover any bugs or problems, which will then be addressed during the meeting and the teams responsible for the assignment will be assigned to deal with these issues. Next meeting, the developers will explain how they handled the issue or the failure to do so and request help if needed.

3.5. Interface Control

The circumstances of interfacing outside of the scope could be endless and therefore no documentation could account for every case. It is however worth noting that any changes should be met with extreme caution to ensure the continued function of any tools/resources of the project.

3.6. Subcontractor/Vendor Control

There are no subcontractors in the Acies project, as all of the work is being done by the Acies team. The dependencies for Acies, such as PyGame, may update their libraries and therefore change our product. However, they have no specific control or say in the application.

3.7. Release Management and Delivery

Time and release will be decided by the development team but will only occur after the Scrum Master has carefully reviewed the progress of the team and approves the release.

4. SCM Schedules

4.1. Sequence and coordination of SCM activities

The guidelines as described in the SCM document will be enacted for each deliverable. Updates of activities will be found on the weekly iterations of our sprint backlog. Team members will work separately or concurrently on tasks in order to finish them.

4.2. Relationship of key SCM activities to project milestones:

Each deliverable represents a milestone in the project. With deployment of new content, we output a new version of our application and output more documentation. Existing documentation is also updated if required. This process allows our development team to make small appropriate revisions to past deliverable releases that were otherwise an oversight during the initial development.

4.3. Schedule of Events

Each deliverable process should resemble the following guidelines:

- Schedule Meeting: The team will communicate on Discord what times would be best for them to meet. A time and location is decided.
- First Meeting: The first meeting will occur preferably at least four days prior to the due date of the deliverable. The Scrum Master is arbitrarily assigned.
    - The team will look over the deliverable and claim tasks that they would like to complete or contribute towards.
    - The team will spend time completing the tasks together. If members have questions, they should ask the others for help.
    - The Scrum Master and the other team members will allow others, once they have finished their tasks, to assist with their own tasks.
    - This is a great time for in-person discussion of the app features or implementation.
    - The next meeting time and location will be decided, if needed.
- Between Meetings: If decided upon, members should work on respective tasks before meeting again.
- Following Meeting(s): The next meeting or meetings will occur if required. If only a few people are finishing up their tasks, then not all members need to be present. The meeting structure should follow that of the first meeting as well as the work in between meetings.
- Review: Material is reviewed, and edited until it is satisfactory.
- Conclusion: Given no objections, the material will all be pushed to GitHub and Google Drive, and submitted to Brightspace.

5. SCM Resources

5.1. Team Resources

a. Tools

Visual Studio Code - IDE
Github - Version Control System
PyGame - Event handler and graphics
PyGame_GUI - UI/UX tooling
Google Drive - Team Documentation Storage
Discord - Team Communication
Physical equipment (hardware) is on a person-by-person basis, primarily simply laptops

b. Nonphysical Tools

The Agile Scrum methodology is used for developing Acies.
Familiarity with the Acies project can be acquired through reading our documentation and reading PyGame documentation (https://www.pygame.org/docs/).
Familiarity of using the application will be accessible with a FAQ section that is scheduled to be created.
As our project is small-scale currently, familiarity can also be gained by contacting one of our members. As developers, we learn from each other, since our combined knowledge is much more developed than our individual knowledge.

5.2. Key Infrastructure Factors:

We chose Pygame as our main library because of its vast range of sound and graphic support to make our application.
GitHub was chosen as our VCS tool due to its great security and because it is industry standard.

5.3. Activities and Tools

Visual Studio Code is used to edit and update the application.
Github is used to share the project with one another and easily push and pull any updates to the software.
Google Drive is used between team members to store documentation.
Discord is used to communicate meeting times, application features, and anything else Acies-related.

6. SCM Plan Maintenance

6.1. Plan Monitor Roles

The Scrum Master is responsible for monitoring the SCM plan and updating it if desired or required. Since the Scrum Master varies, this makes the entire Acies team accountable for maintaining the SCM plan.

6.2. Update Frequency

Updates to the SCM plan will be performed as-needed. The issue of updating the plan will be addressed in the first deliverable meeting. If changes are made, they will be included in the next deliverable submission.

6.3. Change Process

The person who deems that changes are required (once more, the Scrum Master is responsible) will make the changes, then propose them to the team in a deliverable meeting. If the team agrees with the changes, then they will be included in the next deliverable baseline.

6.4. Change Communication

Changes will be made when necessary and all team members will be notified of any changes in-person at a deliverable meeting, as it affects every team member.

6.5. Revision History

Version	**Label**	**Date Enacted**
1.0	First Release	4/3/2022