Configuration Management Plan for the Development of Super Rummy

[Table of Contents]

1. Introduction

1.1. Purpose

This configuration management plan will establish standard practices for guiding the development of the Super Rummy web app. It will also inform others about the development practices used to create the app.

1.2. Scope

The scope of this plan is limited to the development of Super Rummy. Code, documents, and other elements relevant to development will be included in this document.

1.3. Key Terms

Super Rummy: Name of project under development
Rummy: Popular card game where players try to form sets and sequences of cards called melds.

1.4. References

SRS Document
Sprint Backlog

2. SCM Management

2.1. Organization

Rummy Team is the organization developing Super Rummy. Rummy Team consists of four UMaine students enrolled in the course COS 420. Rummy Team is dedicated to ensuring the accuracy of this plan and making development of Super Rummy move along smoothly.

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.
        - This role rotates bi-weekly

- Scrum Master
    - Ensures 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.
        - This role rotates bi-weekly
- Development Team
    - Determine the work needs 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.
        - This role rotates bi-weekly

2.3. Applicable Policies, Directives, and Procedures

There are currently no policies, directives, or procedures that constrain the development of Super Rummy. This document will be updated if any appear in the future.

3. SCM Activities

3.1. Configuration Identification

- **3.1.1: Identify configuration items (events, items, procedures)**
    - Configuration items will be identified via system requirements, user stories, and user feedback.
        - These may reveal ideas that can improve Super Rummy (increased user engagement, app performance, etc)
- **3.1.2: Name configuration items (unique identifiers)**
    - Items should be named in a consistent and clear manner. Underscores should separate words and the deliverable/version number and team name should be clearly visible. For example, this document shall be named: RummyTeam_Deliverable_4_Configuration_Management_Plan.docx
- **3.1.3 Acquiring configuration items (physical procedures)**
    - Configuration items are stored on Github and in a shared Google Team Drive. Unauthorized users will have to request access to the Github repository.

3.2. Configuration Control

- **3.2.1 Requesting changes**
  - Changes to configuration items may be requested through GitHub via an issue or a pull request. Unauthorized users must first contact a team member to gain access to the repository.
- **3.2.2 Evaluating changes**
  - Changes are to be evaluated in Scrum meetings with as many group members present as possible. If a meeting is not possible, changes can also be evaluated over the team group chat in rare cases.
- **3.2.3 Approving or disapproving changes**
  - The team will decide if the change should be implemented, altered, or ignored and add it to the product backlog if necessary. If opinions are split, the change will be added to a log to be reviewed again later.
- **3.2.4 Implementing changes**
  - Changes will be added to the product backlog and be assigned a sprint number. They will also be marked as critical or non-critical. Critical changes should be implemented within the next sprint, if possible. Non-critical changes can be assigned to a later sprint.

3.3. Configuration Status Accounting

- **3.3.1 Metrics to be tracked and reported and type of report**
  - Configuration items are to be tracked on a Kanban board and on Github
- **3.3.2 Storage and access control of status data.**
  - The storage and accessing of status data is controlled by the members of Rummy Team on the project's Github repository

3.4. Configuration Evaluation and Reviews

Configuration evaluations and reviews are to be performed by Rummy Team members during scrum meetings. All CI's must be audited by at least one development team member that was not the main contributor to the CI before it is pushed to Github. This process ensures that CI's have been created according to their specifications in the Grading Guidelines. If the CI matches its specifications or passes its tests(if it is mostly code), it may be pushed to Github.

3.5. Interface Control

Because changes that affect items that interface outside of this scope will be widely varying, no explicit instructions will be laid out here. Despite this fact, care should be taken to ensure changes do not compromise the functionality of Super Rummy.

3.6. Subcontractor/Vendor Control

The development team will need to monitor any changes made to Flask to ensure the functionality of Super Rummy is not compromised by said changes.

3.7. Release Management and Delivery

The time of releases and deliveries will be determined by the COS 420 class schedule. Releases can only occur after necessary CI auditing.

4. SCM Schedules

4.1. Sequence and coordination of SCM activities

Coordination of SCM activities can be found in our Sprint backlog.

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

- **4.2.1 Establishment of configuration baseline**
  - Established at the beginning of a sprint, before development begins
- **4.2.2 Implementation of change control procedures**
  - Changes to code shall occur continuously throughout a sprint
- **4.2.3 Start and completion dates for a configuration audit**
  - Configuration audits begin shortly before a sprint is to be completed, day 9-10
  - Configuration audits should be completed on day 13-14

4.3. Schedule Relative to Sprint Dates

- Day 1:
  - User stories slated for implementation are identified
  - CI's to be created are identified
  - Team discusses goals for sprint
  - Team members take charge of certain CI's
- Day 2:
  - Concrete roles are assigned to team members and development begins
- Day 3-6:
  - Development continues
- Day 7:
  - Scrum meeting to discuss progress
- Day 8:
  - Development continues
- Day 9-10:
  - Initial development completes
  - Auditors assigned to CI's

- Day 11-12:
    - Revisions occur for all CI's
- Day 12:
    - Scrum meeting to discuss overall development and revision progress
    - Tasks that need to go into the next sprint will be identified
- Day 13-14:
    - Revisions complete
    - Audits complete
    - CI's pushed to github

5. SCM Resources

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

- Environment
    - Our programming environment uses Python, Javascript, HTML, and CSS. We use a WebSocket based API for communication between the server and the client, in addition to JSON.
- Infrastructure
    - The projects that Rummy Team works on each have their own infrastructure requirements.
- Software Tools
    - Git
    - Github
    - VScode
- Techniques
    - Git and Github should be used in a way that makes code organized and easy to navigate. While no strict coding styles are enforced, functions should be properly commented so that the team members that did not create them can understand them.
- Equipment
    - Computers that can host a programming environment and act as a server for the game are required for the development of Super Rummy.
- Personnel
    - Rummy Team consists of: Matthew Virgin, Kion Tupper, Gunnar Eastman, Gavin Palazzo
- Training
    - There are no training procedures because no new personnel are being hired.

5.2. Key factors for infrastructure:

- *Functionality, performance, safety, security, availability, space requirements, equipment, costs, and time constraints.*
- Functionality:
    - A server will be required to get the game running full-time. See the SRS document for more details
- Performance:
    - A WebSocket based API is used to keep the communications protocol fast with low latency for a persistent connection.
- Safety + Security:
    - Super Rummy will not collect user's personal data
- Availability:
    - As a website, Super Rummy will be available on most major browsers
- Space requirements:
    - None
- Equipment:
    - A server for full-time availability
- Costs:
    - No costs other than the time it takes to develop Super Rummy
- Time Constraints:
    - Development constrained to one semester for COS 420

5.3. Identify which tools are used in which activity.

- Git, Github:
    - version control
    - code repository
- VScode
    - implementing code changes
- Google Drive
    - document storage
- YouTrack
    - Kanban board to keep track of work status

6. SCM Plan Maintenance

6.1. Who is responsible for monitoring the plan?

Scrum Master

6.2. How frequently updates are to be performed?

Weekly

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

Changes to the plan will follow the same protocol outlined in section 3.2

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

` Changes to the plan will follow the same protocol outlined in section 3.2

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

Name	Date	Change
Matthew Virgin	4/2/2022	Creation of CMP