Software Development Process

Table of Contents

Introduction

This document provides a comprehensive guide to the development methodology, outlining activities, goals, roles, tools, testing procedures, timelines, risk management, documentation practices, and continuous improvement strategies.

The overall goal is to create a seemless, organized, workflow process that allows contributors to collaborate efficiently.

The software development process begins with ideation, where creative concepts are generated as issues, followed by discussion, implementation, iterative refinement, and testing.

Development Phases

1. Planning

The issues serve as a space where ideas are generated, refined, and broken down into their sub-tasks. Team members contribute to discussions, prioritize features, and outline the scope of the task.

An optional backlog board will be available to refine concepts, set milestones, and create a transparent and organized planning process.

2. Communication and Collaboration

The issues and optional backlog board will serve as a hub for project communication.

3. Development

The project will benefit from members iterating through several versions of code before reaching a solution. Through this series of incremental improvements, members can build upon the lessons learned previously, enhancing the project in a meaningful way, while remaining adaptible to new insights discovered.

This iterative development approach (agile in a way), will lead to a codebase that aims to be secure, resilient, and organized solution, ensuring that the final product aligns with the user expectations.

4. Testing

Through the discussions and iterations, members will write unit test that adhere to industry standards.

5. Documentation

Code comments will be written to enhance the understanding when necessary. If a discussion would better explain the code reason, provide a link as a comment instead.

6. Adjustment and Feedback

The discussions about project direction and feature implementation will occur as issues or the optional backlog board. In this way, maintainers to the project can have a full view of the discussion history. Feedback regarding ideas is encouraged as this is the only way to have a discussion and reach an optimal solution.

Roles and Responsibilities

Due to the team size, all members are encourage to collabarate in every section of the product. Repository admins are put in place to create stable development and feature-testing branches.

As of now, there is only one team member. As the project grows, the roles and responsibilities will be updated.

Tools and Technologies

The project will be written in Rust. Additions to the project are not required to be written in rust if it makes sense based on requirements.

Members should be comfortable moving between languages as requirements demand.

Quality Assurance and Testing

The release cycle will follow **Major.Minor.Patch.** - Major: Breaking changes - Minor: New features - Patch: Bug fixes

With the iterative approach taken to the project, quality assurance and testing will be incorporated into the software development process. The ensure that the software meets the required specifications, the following items should be considered during discussion and implementation:

1. Analyze the requirements
2. Plan appropriate tests (security, performance, regression, user)
3. Execute those tests
4. Use the results of the test for further discussion
5. Repeat

Project Timeline

A stable version of the product will take 9 months.

Risk Management

Risks, hurdles, obstacles are to be expected. Through the discussions and iterations, a solution-first approach can mitigate issues as they arise.

Documentation

The documentation will be included in this repository as a README.md file.