Tool Co-op

# Project Overview

This project aims to build a system for renting out tools for a community of users.

This tool co-op system will automatically handle the management of the tool warehouse who is managed by Joe. The software will handle everything from keeping track of tools in a database to renting out those tools to customers, keeping track of checkout and due dates. The overall project will automate the repetitive tasks that Joe does, but still giving Joe functionality such as checking tools and viewing reports of tools and user accounts. The primary priorities for the tool co-op is to provide tools for its community and maintain this business over a long period over time.

# Team Organization

*(The team description should be complete and accurate, yet concise. You may refer to the text book or other authors for standard team organizations. Be sure to describe any team philosophies that you intend to adapt (e.g. egoless programming). You may use a figure to describe your team organization. Also, you may anticipate shifts in responsibilities as the project progresses)*

# Software Development Process

The development will be broken up into five phases. Each phase will be a little like a Sprint in an Agile method and a little like an iteration in a Spiral process. Specifically, each phase will be like a Sprint, in that work to be done will be organized into small tasks, placed into a “backlog”, and prioritized. Then, using on time-box scheduling, the team will decide which tasks the phase (Sprint) will address. The team will use a Scrum Board to keep track of tasks in the backlog, those that will be part of the current Sprint, those in progress, and those that are done.

Each phase will also be a little like an iteration in a Spiral process, in that each phase will include some risk analysis and that any development activity (requirements capture, analysis, design, implementation, etc.) can be done during any phase. Early phases will focus on understanding (requirements capture and analysis) and subsequent phases will focus on design and implementation. Each phase will include a retrospective.

|  |  |
| --- | --- |
| **Phase** | **Iteration** |
| 1. | Phase 1 - Requirements Capture |
| 2. | Phase 2 - Analysis, Architectural, UI, and DB Design |
| 3 | Phase 3 - Implementation, and Unit Testing |
| 4 | Phase 4 - More Implementation and Testing |

We will use Unified Modeling Language (UML) to document user goals, structural concepts, component interactions, and behaviors.

# Communication policies, procedures, and tools

**Customer Communication**

To reasonably build what the customer desires, there is often (but not constant) communication between the development team and the customer in the agile methodology. Since our project is not being shipped to anyone, we will roleplay as the customers and test the product this way. Switching between customer and developer mindsets is necessary in giving insight to the usability and intuitiveness of our product.

**Team Communication**

Communication will be necessary when each member takes time to work on the project. While we already know what each team member will do organizationally, it is necessary to know when and what each person will be doing so that everyone else on the team is aware. When our development team is not available to meet, we will communicate through GroupMe, a group messaging app.

# Configuration Management

See the README.md in the Git repository.