DVCS Management Plan

Jiao Mingyang, Simon Budker, Aeshaan Wahlang, Wei Zhang

**Overview**

In order to create our DVCS, we found it was extremely important to understand the scope of our technical tasks and create a development roadmap following those constraints. In the first section of this document we estimate the development cost of each module in our system design.

**Module Implementation Estimation**

In order to get a better sense of the task we are taking on, as a group we estimated the develop time for a single person to complete each module. Combining all our estimates we arrived at the table seen below. These times include explicit documentation, complete implementation, and extensive quality assurance through various testing strategies.

|  |  |
| --- | --- |
| **Module Name** | **Implementation Estimate (person-days)** |
| User Interface | 2 |
| Repository | 7 |
| File Diff | 5 |
| ChangeSet Log | 5 |
| Manifest Log | 3 |
| File Log | 3 |
| Working Directory | 4 |

**Development Strategy**

Based on the relatively expensive development time constraints, we think we should begin our project by creating a minimum viable product and later expanding the functionality of each module. More specifically, we will focus on basic, essential functionality that compose the simplest DVCS system. Our first tool should do things like commit, checkout, and add, however, it shouldn't need to deal with outside repositories through functionality like push, pull, and clone. To actually make the basic functionality possible we will need to build the core of almost every module. We will approach this by starting with the most essential modules like the Revlog and Repository so we get both a sense of how the modules are integrated, but also how the system is supported from the base data structure. Once we can ensure the quality and functionality of our base system we will move on to the creation of the entire system.