**Software Project Management Plan**

**Team 4**

March 2, 2016

**Team Members**

Hemin Qaradagi

Dan Stucky

Drew Shoemaker

Jeff Sterner

Mohamed Said

Document Control

**Change History**

|  |  |  |
| --- | --- | --- |
| **Revision** | **Change Date** | **Description of changes** |
| V1.0 | 03/03/2016 | Initial release |
|  |  |  |

**Document Storage**

This document is stored in the project’s github repository at: https://github.com/cs451/Spring2016Team4

**Document Owner**

Hemin Qaradagi is responsible for developing and maintaining this document.

**Table of Contents**

[1 Overview 4](#_Toc444704262)

[1.1 Purpose and Scope 4](#_Toc444704263)

[1.2 Goals and Objectives 4](#_Toc444704264)

[1.3 Project Deliverables 5](#_Toc444704265)

[1.4 Assumptions and Constraints 5](#_Toc444704266)

[1.5 Schedule and Budget Summary 6](#_Toc444704267)

[1.6 Success Criteria 6](#_Toc444704268)

[1.7 Definitions 6](#_Toc444704269)

[1.8 Evolution of the Project Plan 7](#_Toc444704270)

[2 Startup Plan 7](#_Toc444704271)

[2.1 Team Organization 7](#_Toc444704272)

[2.2 Project Communications 8](#_Toc444704273)

[2.3 Technical Process 8](#_Toc444704274)

[2.4 Tools 8](#_Toc444704275)

[3 Work Plan 8](#_Toc444704276)

[3.1 Activities and Tasks 8](#_Toc444704277)

[3.2 Release Plan 9](#_Toc444704278)

[3.3 Iteration Plans 9](#_Toc444704279)

[***3.3.1 First Iteration*** 9](#_Toc444704280)

[***3.3.2 Second Iteration*** 9](#_Toc444704281)

[3.4 Budget 9](#_Toc444704282)

[4 Control Plan 9](#_Toc444704283)

[4.1 Monitoring and Control 9](#_Toc444704284)

[4.2 Project Measurements 10](#_Toc444704285)

[5 Supporting Process Plans 11](#_Toc444704286)

[5.1 Risk Management Plan 11](#_Toc444704287)

[5.2 Configuration Management Plan 11](#_Toc444704288)

[5.3 Verification and Validation Plan 12](#_Toc444704289)

[5.4 Product Acceptance Plan 12](#_Toc444704290)

# Overview

## Purpose and Scope

The purpose of ACH Case Tracking Phase I will be to inhance visibility into various types of Government Reclamation requests received. This will allow the ACH Department to limit their risk of missing a request, resulting in a loss to the bank. This will also allow greater efficiency for the DNE process to be able to import the DNEs instead of logging each one individually in excel. The team would use this functionality to keep track of open requests, make appropriate comments in relation to action taken on requests, and be able to research past requests easily. This phase should also include reporting abilities, specifically month end type reporting, to allow us (at a minimum) to see how many cases were worked in a given month. These counts should be able to be broken down by sub-type.

The ACH Case Tracking application will feature a friendly UI with a main menu that allows users to navigate to a manual case creation screen, import cases from a NACHA file, or search for specific cases according to various parameters. Cases may be edited after their creation and their state can be updated as progress on the case advances.

In addition, cases may be reopened and modified anytime after their closure. In order to not overwhelm the user with excess form data, the inputs for manually creating a case will update as the user enters in information so that only relevant items are shown on screen at once. Individual case views will also feature a SLA so users can see when a deadline is approaching. All case creation and editing instances will be audited in an organized and structured view according to date.

The client would like to have an authentication and reporting system built into the app as well, but these features will likely be left out due to time constraints associated with their implementation and scale of the project.

## Goals and Objectives

There are three main goals associated with the production of the ACH Case Tracking project:

1. Standardize and automate the process of DNE case creation.
2. Allow users to track and quickly find DNE cases stored in the system.
3. Add accountability by providing a transparent auditing layer for case creation and editing.

## Project Deliverables

|  |  |
| --- | --- |
| **Date** | **Deliverable** |
| 2/8/2016 | Project Charter |
| 2/11/2016 | Product Backlog |
| 2/11/2016 | Iteration #1 Plan |
| 2/19/2016 | Requirements Specification |
| 2/29/2016 | Technical Prototype |
| 3/3/2016 | Project Plan |
| 3/6/2016 | Iteration #1 Complete |
| 3/9/2016 | Customer Approved UI Prototype |
| 3/16/2016 | Architecture Document |
| 3/20/2016 | Iteration #2 Complete |
| 3/21/2016 | Project Demo |
| 4/10/2016 | Iteration #3 Complete |
| 4/27/2016 | User Guide and System Administration Manual |
| 5/2/2016 | Iteration #5 Complete |
| 5/5/2016 | Product Released |
| 5/6/2016 | Present Product to Commerce |

## Assumptions and Constraints

Assumptions:

* 1. Commerce will provide some form of project feedback.
  2. Changes in requirements will be conveyed at least 2 weeks prior to their deadline.
  3. All requirements can be completed with respect to the financial budget.

Constraints:

1. No expenses are available for the project
2. App completed by May 5, 2016.
3. App built using Java MVC or .NET framework.
4. Database must be relational

## Schedule and Budget Summary

The project has no budget to work with.

Gantt Chart of Project Schedule

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Step | Iteration | 1 | | | | | | 2 | | 3 | | | 4 | | 5 | |
| Week | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Date Range | 1/25 - 3/6 | | | | | | 3/7 - 3/20 | | 3/21 - 4/10 | | | 4/11 - 4/24 | | 4/25 - 5/2 | |
| Design Architecture | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|
| Specify/Evaluate Requirements | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|
| Iteration 1 Dev | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|
| Iteration 2 Design | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|
| Iteration 2 Dev | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|
| Iteration 3 Design | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|
| Iteration 3 Dev | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|
| Iteration 4 Design | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|
| Iteration 4 Dev | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|
| Iteration 5 Design | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|
| Iteration 5 Dev | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|

## Success Criteria

The project will be considered a success if (1) the team delivers an operational prototype at the end of the semester with the features mentioned in the goal section above, and (2) 80% or more of the team members would be willing to work together on another software project in the future.

## Definitions

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **Actor** | user or other software system that receives value from a user case. |
| **Baselined** | the work product has undergone a formal review and can only be changed through the prescribed change control procedures |
| **Client or Customer** | the person or organization for which this Roo Balance application is being built. |
| **Developer** | the person or organization developing the system, also sometimes called the supplier. |
| **Project** | activities that will lead to the production of the Roo Balance application. |
| **ACH Case Tracking** | the product that is being described here; the software system specified in this document. |
| **Scenario** | one path through a user case |
| **Stakeholder** | anyone with an interest in the project and its outcomes. This includes clients, customers, users, developers, testers, managers and executives. |
| **User** | the person or persons who will actually interact with the Roo Balance application. |
| **Use case** | describes a goal-oriented interaction between the system and an actor. A use case may define several variants called scenarios that result in different paths through the use case and usually different outcomes. |

## Evolution of the Project Plan

Changes to the project plan must be approved by the project manager and will appear in the change history of this document.

Before the start of an iteration, the project plan will be updated to include a broad description of the tasks for the upcoming iteration. A record of the detailed tasks will be kept on our Trello Board (https://trello.com/b/cKEJLQlU/scheduling).

Risk mitigation efforts will be evaluated at the start of each iteration. Severe risks will be analyzed and added to the project plan as soon as they materialize.

# Startup Plan

## Team Organization

|  |  |  |
| --- | --- | --- |
| **Role** | **Actor(s)** | **Responsibility** |
| Project Manager | Hemin Qaradagi | Call team meetings, coordinate communications within group, coordinate communications outside group, break out tasks, assign them to teammates |
| Front End Developer | Dan Stucky | Develop user interface software based on requirements. Testing |
| Business Logic Developer | Drew Shoemaker | Develop models complete with logic for handling required operations. Testing |
| Back End  Developer | Jeff Sterner, Mohamed Said | Develop and integrate relational database. Aid other programming features as needed. |

## Project Communications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Event** | **Information** | **Audience** | **Format** | **Frequency** |
| Team Meeting | Task status: completed since last meeting & planned for next;  obstacles encountered; change requests in process | All team members | Informal meetings following class; Formal meetings as needed; Slack chat for status updates & problems as they occur | As needed |
| Project Status Report | Review finished items, status of prototype; review any problems, schedule slippage, programming issues | All team members, customer | E-mail with information or In-person as customer sees fit | Twice:  Project Demo & Final Presentation. (see milestones for dates) |

## Technical Process

An iterative and incremental development process is planned. Feedback will be used from each iteration to improve the next. The first iteration will focus on basic functionality of the application. Subsequent iterations will build upon that and incorporate more features as time allows.

## Tools

* Programming & Markup Languages – Java, Scala, JavaScript, HTML
* Framework - Java Play 2.4.x (Web), JUnit (Testing)
* Browser – IE9+
* Version Control – all work products will be stored in a github repository
* Development Tools – IntelliJ IDE with Scala and Play plugins.
* Build Tools - Travis
* Developer Libraries - Play-Bootstrap (includes Bootstrap & JQuery), play2-pdf (for exporting pdfs)
* Database technologies - PostgreSQL (production), H2 (development)

# Work Plan

## Activities and Tasks

Tasks are arranged by iteration. Details of individual tasks can be found on Trello: https://trello.com/b/cKEJLQlU/scheduling

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iteration #1 | | | | |
| Task ID | **Task Description** | **Est Effort** | **Actual Effort** | **Dependencies** |
| 1.1 | H2 datbase setup for dev environment | 2 | 2 | 1.2 |
| 1.2 | Incorporate project dependencies | 2 | 2 |  |
| 1.3 | Manual creation of a case | 9 | TBD |  |
| 1.4 | Design DB for storing case data | 8 | TBD |  |

## Release Plan

**01/29/2016 - 1st half of Project Charter Complete**

**02/05/2016 - Gather requirements**

**02/08/2016 - 2nd half of Project Charter Complete**

**02/10/2016 - Product Backlog & Use Cases for Iter #1 Complete**

**02/17/2016 - Requirements Document Complete**

**02/29/2016 - Technical Prototype Complete**

**03/02/2016 - Project Plan Complete**

**03/06/2016 - Iteration #1 & Use Cases for Iter #2 Complete**

**03/09/2016 - Customer Approved UI Prototype Complete**

**03/16/2016 - Architecture Document Complete**

**03/20/2016 - Iteration #2 & Use Cases for Iter #3 Complete**

**03/21/2016 - Project Demo**

**03/25/2016 - Sprint Review with Commerce Bank**

**04/10/2016 - Iteration #3 & Use Cases for Iter #4 Complete**

**04/24/2016 - Iteration #4 & Use Cases for Iter #5 Complete**

**04/27/2016 - User Guide and System Documentation Complete**

**05/02/2016 - Iteration #5 Complete**

**05/05/2016 - Product Released**

**05/06/2016 - Present Product to Commerce Bank**

## Iteration Plans

Detailed iteration plans are provided in our Trello board at https://trello.com/b/cKEJLQlU/scheduling.

In general, each iteration will be planned a few days before it begins.

***3.3.1 First Iteration***

Form for manually entering ACH cases into the system. Integration with H2 database for mocking permanent storage during development.

***3.3.2 Second Iteration***

## Budget

Project has no monetary budget as all labor and technology is unpaid or free.

However, team members will have to devote about 5 hours/week to complete project assignments in a timely manner. Time is a valuable expense for students and time management skills will be key for project completion.

# Control Plan

## Monitoring and Control

Weekly – Team meeting. Project participants report status, progress and potential problems.

2/29/2016 – Technical Prototype complete

3/07/2016 – 5 minute status report

3/06/2016 - iteration 1 ends

3/09/2016 - customer approved UI due

3/20/2016 - iteration 2 ends

3/25/2016 - sprint review with Commerce

4/10/2016 - iteration 4 ends

4/27/2016 - user guide and documentation due

5/2/2016 - project due

# Supporting Process Plans

## Risk Management Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Rank** | **Risk** | **Probability of Loss** | **Size of Loss** | **Risk Exposure** | **Response** |
| 1 | Schedule / time line delivery | Likely | Major | High | Mitigate: Stick to the schedule. |
| 2 | Unforeseen fields needed in database | Likely | Moderate | Moderate | Design database for flexibility |
| 3 | Learning curve for new tools and technologies longer than expected | Unlikely | Moderate | Moderate | Buy Information: Begin working on a basic prototype early to test out fundamental programming concepts & knowledge |
| 5 | Underestimated effort for UI design | likely | moderate | moderate | Invest time in learning tools early. Build prototypes for testing. |
| 4 | Breakdown in team communication | likely | moderate | moderate | Commit to set amount of time for team communication at least once a week |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Configuration Management Plan

The following procedure is to be used when making changes to all baselined work products:

1. All project work products will be stored in a centralized github repository.
2. All baselined documents will have a Document Control section with a change history to track initialization and subsequent changes.
3. All project work products (documents, source code, test cases, program data, test data, etc) will be stored in the git repository but not all will be under change control (subject to formal change control procedures.) Only the system requirements, project plan and source code will be baselined and under configuration control.
4. Items that are subject to change control will be considered baselined after a group review at the end of the initial document creation.
5. The change control procedure once a product is baselined is:

(1) anyone wanting to make a change to a baselined item sends a message to the team members describing the change, reason for the change, expected schedule impact, and time line for integrating the change.

(2) if no one responds to the group within 2 days with a reason for why the change request shouldn't be permitted, it will be considered accepted and the person proposing the change may proceed with the change.

1. if anyone does object to the change, the reason for objecting will be discussed at a meeting where everyone is invited to attend and voice their opinion (may be online). At the end of the meeting a democratic vote will be held to decide whether or not the change should be allowed.
2. if a change takes place, the initiator must collaborate with the project manager to update the schedule.

## Verification and Validation Plan

In order to uphold the integrity and quality of the written code and final product, a couple of actions are to be taken.

1. All code must pass a review before being merged into the master branch. To pass a code review, the changes should be approved by the project manager and at least one other engineer.

a) In the event that nobody responds to a code review request within 2 days, the initiator may proceed with the merge in order to keep progress moving.

1. Code reviews entail checking code quality as well as feature testing in IE browser.
2. Business logic code should be associated with a written suite of JUnit tests to reduce the chances of missed errors and regression.

## Product Acceptance Plan

At the conclusion of each iteration, the prototype created will be tested to ensure it meets the requirements of that iteration. Because Commerce Bank is most interested in running the application through an IE browser, testing should be incorporated into this browser in addition to Chrome and FireFox.

For the final iteration, product acceptance testing will ensure that the prototype functions as expected with a user's data. All planned use cases may not be incorporated into the final product due to time constraints, but features that are present should function as outlined in the requirements. The exact features included in the final product is intentionally left vague here. However, at a minimum, the product should include manual user case creation, automatic case creation, case searching, and viewing and editing cases.