Software Project Management Plan

Idea Engine

Team: FDIC#

October 10, 2014

Team Members

Ryan Johnson
Caleb Iamkui
Robert Holland
Carol Du
Eric Gonzalez
Samir Ibrahimi

Document Version Control

Change History

Revision	Change Date	Description of changes	Author
Version 1.0	10/10/14	Initial release	Ryan Johnson
Version 1.1	10/15/2014	• Revised section 2.3 • Added tasks for Iteration 1 • Added tasks for Iteration 2	Ryan Johnson
Version 2.0	10/26/2014	• Updated Iteration Plan for Iteration 2 and 3	Ryan Johnson
Version 3.0	11/11/2014	• Updated Iteration 3 plan	Ryan Johnson
Version 4.0	11/23/2014	• Updated Iteration 4 plan, Iteration 5 plan • Updated Final Presentation Date	Ryan Johnson
Version 5.0	12/4/2014	• Final Document • Updated Iteration 5	Ryan Johnson

Document Storage

This document and other project documents are stored on GitHub at:
FDIC# - Project Documents

Document Owner

Ryan Johnson is the main author and contributor of this document.

Table of Contents

1 Overview

1.1 Purpose and Scope

Commerce bank development team often coordinates with local universities and colleges to create technologies that give students experience in the software life cycle. A great way to learn this technical process is to work through a project with real-life requirements.
In order to be effective, commerce employees need a focal point to collect project ideas, refine the project details, and then send the project into production.
The purpose of this project is to provide a central point for all Commerce employees that engage in the ideas for university projects. The goal is to create a web application that is easily accessible by all Commerce domain users, given their appropriate group permissions. At this web page, users can see already submitted project ideas, and submit their own ideas. The project information is detailed on sub-level pages, where the project's information can be managed by users with non-standard permissions.

A web-based application is necessary for this kind of collaborative work. Any commerce employee on the commerce domain will have access to this web site.

The user interface will be intuitive and simple to navigate, with at least 90% of new users using the features of the site within 20 seconds without any prior training. A user guide and system administration manual will be provided.

1.2 Goals and Objectives

The overall objective is to give Commerce employees a central point to submit a project idea.

Project Goals:

1. Create a web app that functionally meets the customer's requirements.
2. Use the technology mandated by the customer.
3. We, as the students developing the web app, gain experience working on a team and using ASP.NET, C#, Razor Engine, and MVC technologies.

Project Objectives:

1. Create a web app that meets functional and non-functional needs of users.
2. Provide intuitive interface for Commerce bank users for project ideas.

1.3 Project Deliverables

Date	Deliverable
09/10/2014	Project Charter
09/17/2014	Product Backlog
9/22/2014	Technical Prototype (to reduce technology risk)
9/24/2014	Requirements Document (Baseline)
10/10/2014	Project Plan
10/12/2014	Iteration 1 Post Report
10/15/2014	Customer Approved UI Prototype
10/26/2014	Iteration 2 Post Report
10/29/2014	Architecture Document (Baseline)
11/09/2014	Iteration 3 Post Report
11/12/2014	Testing Report
11/23/2014	Iteration 4 Post Report
12/03/2014	User Guide and System Documentation
12/08/2014	Project Due & Code Freeze
12/10/2014	Product Released & Presentations

1.4 Assumptions and Constraints

1.4.1 Assumptions

1. The customer will host a requirements meeting to gather system requirements.
2. The development team has not used these technologies, but will learn them to create the project.
3. Security framework will be provided by Commerce.

1.4.2 Constraints

Constraints:

1. The application will be web-based.
2. The technology used is ASP.NET, C#, and MVC.
3. The final solution should not reply on any third-party licensed software (beyond the operating system and browser).

1.5 Schedule and Budget Summary

1.5.1 Cost Estimate

There is not a fiscal budget for this project, so we will define cost in terms of hours spent on the project.

Title	**Hours/Week**	**Number of Weeks**	**Total Hours**
Project Manager	1x5 = 5	14	70
Software Engineer	2x5 = 10	14	140
Web Developer	2x5 = 10	14	140
Database Designer	1x5 = 5	14	70
		TOTAL HOURS:	**420**

Actual hours are tracked in a separate document.

1.5.2 Schedule Summary

Step	Iteration	1					2		3		4		5		
	Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	Start Date	09-07	09-14	09-21	09-28	10-05	10-12	10-19	10-26	11-02	11-09	11-16	11-23	11-30	12-07
Evaluate Rqmts.															
Design Arch.															
Iteration 1 Dev															
Iteration 1 QA															
Iteration 2 Design															
Iteration 2 Dev															
Iteration 2 QA															
Iteration 3 Design															
Iteration 3 Dev															
Iteration 3 QA															
Iteration 4 Design															
Iteration 4 Dev															
Iteration 4 QA															
Iteration 5 Design															
Iteration 5 Dev															
Iteration 5 QA															
Product Release															

1.6 Success Criteria

The project will be deemed a success if all requirements are met on schedule (the project due date), and the customer has given positive feedback on the finished product.

1.7 Definitions

Term	Definition
Actor	System user or other software system that receives interacts with the system.
Baseline	Document that outlines formal requirements of a specified document. Changed via Change Control Procedures.
Customer/Commerce	The product owner
Developer	The organization responsible for developing the application. (FDIC#)
Project	The Idea Engine web application
Scenario	one path through a user case
Stakeholder	Entities with an interest in the project and its outcomes. This includes clients, customers, users, developers, testers, managers and executives.
User	A person that uses the application.
Use case	Describes a unique situation within the application and the possible user scenarios within that situation.

2 Startup Plan

2.1 Team Organization

Role	Actor(s)	Responsibility
Project Manager	Ryan Johnson	Determines priorities, assigns tasks, hold team meetings, set expectations, resolve personnel issues, coordinate needs among group members.
Developers	Caleb Iamkui (L) Robert Holland	Develop software using ASP.NET, setup Models, create Controllers, setup Views
Web Developers	Eric Gonzalez Carol Du	Program CSS, JavaScript, CSHTML to optimize UI
Database Designer	Samir Ibrahimi	Create database scripts per E-R diagram
Architect	Robert Holland	Specify overall internal workings of application
Tester/QA	Samir Ibrahimi, Ryan Johnson, Carol Du	Write test cases, perform unit testing of test cases against incremental release of code, report issues and bugs
Requirements Engineer	Ryan Johnson	Outline and document project requirements and note external dependencies.

2.2 Project Communications

For formalized Expectations see FDIC# - Ground Rules.pdf at FDIC# - Project Documents

Event	Information	Audience	Format	Frequency
Monday meeting	Weekly overview, plan next Team Meeting	All team members	~5-10 minutes standing meeting	Weekly
Team Meeting	Task status: Current iteration issues, planning, discussing next iteration	All team members	Informal meetings following Wednesday class; Formal meetings as needed; E-mail status updates & problems as they occur	As needed
Informal Communication	Bugs, Ideas, concerns	All team members	Use Trello cards to communicate in a chat-like format or email team members	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	Iteration Closeout

2.3 Technical Process

- We will use the Scrum development methodology to develop iteratively and incrementally.
- Each iteration has 2 major development components: MVC and User Interface Each iteration spawns two new branches from Master: UI and MVC.
- Once iteration tasks are complete, the MVC Branch merges with UI Branch to a temporary Test Branch. Once determined the merge is complete, the Test Branch merges with Master Branch.

- This approach allows us to divide and conquer tasks, then integrate often. While functionality of the application is the main priority, it is important that non-functional goals are met.
- We are using a database first approach, given that we are comfortable with that process. This will reduce technical risk.
- Feedback will be used from each iteration to improve the next. Subsequent iterations will build upon previous iterations and implement additional features.

2.4 Tools

- Development Environment – Visual Studio 2013
- Version Control – GitHub repository for FDIC#
- Technology and Languages – ASP.NET, MVC, C#, Razor, CSS, JavaScript, jQuery
- Browser is Internet Explorer version 9 and above

3 Work Plan

3.1 Resource Estimate

Detailed resource estimates are available in the linked file < FDIC# - Iterations- Sprint Backlog.pdf >. In this document, tasks, roles, owners, and effort estimates & actuals are listed.

3.2 Release Plan

3.2.1 Plan by Feature

Iteration #1 **9/29/2014 – 10/12/2014**

Summary: Create initial project, Models, Controllers, Views, UI Prototype

Features / Deliverables	**Estimated Effort**	**Actual Effort**
Risk Management – Tech Prototype	5	2
Architecture – View Models, Controllers	5	6
UI Prototype	5	4.5
Database Design	5	5.5
MVC & UI Integration	3	2.5

Iteration #2 **10/13/2014 – 10/26/2014**

Summary: Integrate Database to controllers, save info, recover, edit, save history, refining UI to match the Views

Features / Deliverables	**Estimated Effort**	**Actual Effort**
Database Integration	5	3
Database Manipulation	5	3
Controllers for Data Manipulation	5	3
User Interface	5	3
UI Integration	2	1

Iteration #3 **10/27/2014 – 11/09/2014**

Summary: Create user-specific views, assign users their system abilities

Features / Deliverables	**Estimated Effort**	**Actual Effort**
Changing status of a project	3	3
Assign schools, emissaries to project	3	3
Set constraints by permission level	3	3

Iteration #4 **11/10/2014 – 11/23/2014**

Summary: Restrict features to non-emissary users. Increase usability/filters

Features / Deliverables	**Estimated Effort**	**Actual Effort**
Overflow of project lists	10	12
Render views/partial views to users	10	10
Custom filters for users	10	10

Iteration #5 **12/01/2014 – 12/07/2014**

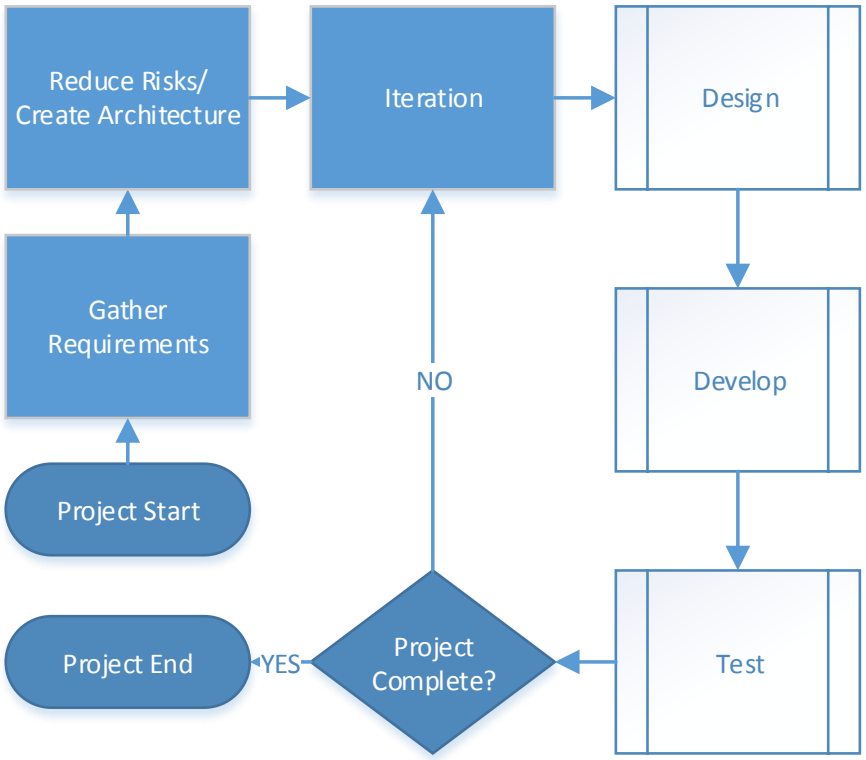
Summary: Addition of 'nice to have' features

Features / Deliverables	**Estimated Effort**	**Actual Effort**
Adding constraints to users.	10	10
Refine Usability	6	6

Features not scheduled, but under consideration

Features	**Estimated Effort**
Mobile usability	
Run ad hoc reporting with export to CSV	

3.2.2 Flow Chart of Project



3.3 Iteration Plans

A detailed iteration plan will be provided for Iteration 1. Further task details for all iterations are available in the <FDIC# - Iterations- Sprint Backlog.pdf>

3.3.1 First Iteration

Initial project is created and uploaded to GitHub. Technology risks have been reduced. Main landing page is designed, and controllers allow information to be saved to a database.

3.3.2 Second Iteration

This focuses on data manipulation for projects, emissaries, administrators, and schools. The ability to add, edit, and delete a project.

3.3.3 Third Iteration

Focus on user-specific views. Each view adds/removes constraints for users based on the user permission level.

3.3.4 Fourth Iteration

Usability increase by filtering, list overflow, etc. M-N relationships, file upload, database relationship management.

3.3.5 Fifth Iteration

Implementation of 'nice to have' features such as email notifications to subscribers and running reports. Usability Testing, refactoring code, fixing bugs, and refining the UI.

4 Control Plan

4.1 Monitoring and Control

The following list of dates includes formal reviews outside of the Release Plan. Milestones are included to reference where the project is scheduled to stand as these reviews occur:

Date	Review / Milestone
09/22/2014	*Milestone: Technical Prototype Complete*
10/13/2014	5-Minute Status Report
10/12/2014	*Milestone: Iteration #1 Complete*
10/22/2014	Manager's Briefing
10/26/2014	*Milestone: Iteration #2 Complete*
10/29/2014	Meeting with Commerce
11/09/2014	*Milestone: Iteration #3 Complete*
11/12/2014	*Milestone: Test Report Complete*
11/19/2014	Class Inspection
11/23/2014	*Milestone: Iteration #4 Complete*
12/03/2014	System Documentation Due
12/07/2014	*Milestone: Iteration #5 Complete*
12/08/2014	*Milestone: Product Released*
12/12/2014	Final Presentations

4.2 Configuration Management Plan

The following procedure is to be used when making changes to all baseline documents:

1. All project working versions are staged on a GitHub repository.
2. All baseline documents follow a Document Control section with a change history to track initialization and subsequent changes.
3. All project working and finished products will be staged on the FDIC# GitHub Repository.
4. Items that are subject to change control will be considered baseline after a group review at the end of the initial document creation.
5. The following documents are subject to Change Control:
   - System Requirements
   - Project Plan
   - Architecture
6. The Change Control Procedure on a baseline document:
   (1) Team member(s) wanting to make a change to a baseline item will send an email to the rest of the team describing the change.
   (2) A team meeting with all team members will be scheduled to discuss in person the reasons and details of the change.
   Factors: Expected Schedule Impact, necessity of the change, and timeline to integrate the change.
   (3) After a team discussion, a final vote will be held to move forward with the change.
   (4) If a change takes place, the initiator must collaborate with the project manager to update the schedule and corresponding documents.
      1. Old versions of documents are archived with a filename change (i.e. FDIC# - <document> - v1.0)
      2. New versions of documents are staged on GitHub with filename change (i.e. FDIC# - <document> - v2.0)

5 Supporting Process Plans

5.1 Risk Management Plan

Rank	Risk	Probability of Loss	Size of Loss	Risk Exposure	Response
1	Technology Learning Curve	Unlikely	Major	High	Create technical prototype using MVC. From View, save information to a database, and recover data
2	Not meeting project milestones	Likely	Moderate	Moderate	Plan, assign, and schedule tasks. Stick to the plan
3	Not meeting functional requirements	Unlikely	Moderate	High	Have clear idea of project requirements, and keep product backlog up to date, design use cases.

5.2 Test Plan

The test plan will be staged in a separate document on GitHub.
See FDIC# - Test Plan - v1.pdf located at FDIC# - Project Documents

5.3 Product Acceptance Plan

At the conclusion of each iteration, the product will be tested by teammates not directly tied to the development to ensure it meets the functional requirements of that iteration. Each tester will need to sync the Master branch to run the application from Visual Studio.

During the final iteration, the system functional requirements should be met and additional optional features will be implemented. A branch called 'Cleaning Up' will be used to make final code comments, code refactoring, and optimizing UI.