FAAST Inventory System

Version 2.1

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 15/09/16 | 1.0 | Initial Creation and simple revisions | Jonathan Ebert |
| 17/09/16 | 1.1 | Content added to section 3 | Jonathan Ebert |
| 19/09/16 | 1.2 | Content added to section 1 & 2 | Grayson Scott |
| 22/09/16 | 1.2 | Content added to section 3 & 4 | Jonathan Ebert |
| 25/09/16 | 1.3 | 4.1 – 4.2 updated | Jonathan Ebert |
| 26/09/16 | 1.4 | 4.2 updated | Jonathan Ebert |
| 27/09/16 | 1.5 | 4.2 updated | Jonathan Ebert |
| 27/09/16 | 2.0 | Documents Merged | Grayson Scott |
| 29/09/16 | 2.1 | Section 4.3 updated | Grayson Scott & Jonathan Ebert |

Table of Contents

1. Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.3 Definitions, Acronyms, and Abbreviations 3

1.4 References 3

1.5 Overview 3

2. Project Overview 3

2.1 Project Purpose, Scope, and Objectives 3

2.2 Assumptions and Constraints 3

2.3 Project Deliverables 3

2.4 Evolution of the Software Development Plan 3

3. Project Organization 3

3.1 Organizational Structure 3

3.2 External Interfaces 3

3.3 Roles and Responsibilities 3

4. Management Process 3

4.1 Project Estimates 3

4.2 Project Plan 3

4.2.1 Phase Plan 3

4.2.2 Iteration Objectives 3

4.2.3 Releases 3

4.2.4 Project Schedule 3

4.2.5 Project Resourcing 3

4.2.6 Budget 3

4.3 Iteration Plans 3

4.4 Project Monitoring and Control 3

4.4.1 Requirements Management Plan 3

4.4.2 Schedule Control Plan 3

4.4.3 Budget Control Plan 3

4.4.4 Quality Control Plan 3

4.4.5 Reporting Plan 3

4.4.6 Measurement Plan 3

4.5 Risk Management Plan 3

4.6 Close-out Plan 3

5. Technical Process Plans 3

5.1 Development Case 3

5.2 Methods, Tools, and Techniques 3

5.3 Infrastructure Plan 3

5.4 Product Acceptance Plan 3

6. Supporting Process Plans 3

6.1 Configuration Management Plan 3

6.2 Evaluation Plan 3

6.3 Documentation Plan 3

6.4 Quality Assurance Plan 3

6.5 Problem Resolution Plan 3

6.6 Subcontractor Management Plan 3

6.7 Process Improvement Plan 3

7. Additional Plans 3

8. Annexes 3

9. Index 3

# 

# Introduction

The Software Development Plan for the FAAST Inventory System is the artifact that will be used to describe what we are doing, how we will accomplish it and our definition for when we have completed the project. Additionally, the document will define the scope of the project, external stakeholders, the number of iterations, and the processes we will use in order to complete the project.

## Purpose

The purpose of the Software Development Plan is to map out the development activities for the iterations and deliverables that will be needed to create an inventory system that will track assistive technology devices for FAAST.

## Scope

The project will be submitted by April, 2017, and will be considered done when all of the requirements for the project have been met.

## Definitions, Acronyms, and Abbreviations

Assistive Technology – Any devices that provides functionality or benefits to a client who has a disability. This can range from iPads with specialized applications to simply toys.

FAAST – Florida Alliance of Assistive Services and Technology

Inventory System – Software for tracking inventory levels. In our case, in order to track the number of assistive technology devices or items that FAAST currently has and whether the item has been checked out or not. **USE ADRIAN’S ABBREVIATIONS**

## References

* Group Contract
* Course Syllabus and Schedule

## Overview

**FILL IN**

# Project Overview

## Project Purpose, Scope, and Objectives

FAAST is a company that provides assistive technology available for loan to the Jacksonville area. Currently, their inventory system is lacking in functionality and efficiency and they are looking for a replacement. The purpose of our project is to provide an inventory system that may be used to replace certain functionalities of their current inventory system.

The deliverables that are expected to be completed is **FILL IN**

## Assumptions and Constraints

Assumptions:

* The project will be completed by May 2017.
* Access to the project will be granted to FAAST employees only (no client portal).
* The team responsible for completing the application will remain the same.
* We will be using Visual Studio 2015 and SQL Server 2012 to complete the project.

Constraints:

* Work will be suspended during the month of December because of winter break.
* The team members associated with this project are all full-time students, which causes time constraints.

## Project Deliverables

* **Deliverable 1 – 09/29**
  + Project Vision
  + Product Backlog & User Stories
  + Sprint Backlog
  + Software Development Plan
  + First Iteration Product Release
* **Product Release 1 Demo – 10/04**
* **Deliverable 2 – 10/27**
* **Deliverable 3 – 12/03**

## Evolution of the Software Development Plan

This document is expected to undergo revisions in the future. See Revision History at the top of the document.

# Project Organization

## Organizational Structure

Team Singularity consists of four student developers and one mentor. While there are specific tasks that only some team members carry out, no one team member is designated as the team leader. Each student team member is responsible for their parts of the project and their progress is monitored by their mentor and their professor.

## External Interfaces

The project will interact with the Hope Haven’s FAAST employees. More specifically, Edward Monagan is the main internal contact and he is the FAAST N.E. Regional Coordinator at Hope Haven. Operating out of Hope Haven, this FAAST Regional Demonstration center interfaces with the public and professional consumers. All consumers of the AT will interact with Edward Monagan or Eric Witherspoon.

## Roles and Responsibilities

The table below depicts the different project organizational units that are needed to carry out this project.

|  |  |
| --- | --- |
| **Role** | **Responsibility** |
| Developer | Without a dedicated project manager, the developers on the team are responsible for all client interaction, all of the work that is required to complete a software project, and managing work distribution between team members. All the developers share equal responsibility for maintaining client interaction, completing their share of work, and learning any necessary skills needed for completing their work.  Possible work to be completed by the developers includes, but is not limited to:   * Writing code * Testing Code * Designing UI * Database design and creation * Client Interaction |
| Mentor | Responsible for keeping track of the student developers’ progress and conducting code reviews. By simulating a real software team using the Software Development Life Cycle, the mentor uses the opportunity to teach the students valuable skills for real-world application. |

# Management Process

## Project Estimates

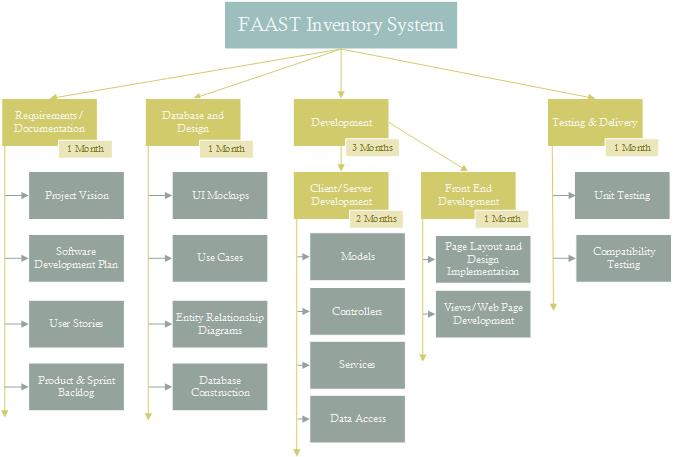
Project initiation and initial requirements phase will take one month and requirements for the project will be reformed or rewrote in following phases as needed. This is a zero cost project in which the client does not pay for the software they will receive nor will the team members be compensated for their time.

There are 7 deliverables within the lifetime of this project and each deliverable has a known due date. The final product will be delivered to the client on or before the last day of the Spring semester, April 28th. Re-estimation for project scheduling will occur at the end of each deliverable sprint.

## Project Plan

### Phase Plan

|  |  |  |
| --- | --- | --- |
| Milestones | Date | |
| **Start\*** | **End/ Due Date** |
| Deliverable 1 | 30/08/16 | 29/09/16 |
| Product Release 1 Demo |  | 04/09/16 |
| Deliverable 2 | 01/10/16 | 27/10/16 |
| Product Release 2 Demo |  | 01/11/16 |
| Deliverable 3 | 02/11/16 | 01/12/16 |
| SOC Symposium |  | 02/12/16 |
| Product Release 3 Demo (Viable Product) |  | 06/12/16 |
| Deliverable 4 | TBD | TBD |
| Product Release 4 Demo |  | TBD |
| Deliverable 5 | TBD | TBD |
| Product Release 5 Demo |  | TBD |
| Deliverable 6 | TBD | TBD |
| Product Release 6 Demo |  | TBD |
| Deliverable 7 | TBD | TBD |
| SOC Symposium |  | TBD |
| Final Product Delivery |  | TBD |

 \*Not applicable to all fields

### Iteration Objectives

|  |  |
| --- | --- |
| Iteration | Objective |
| Sprint 1 | Project Vision, Software Dev. Plan, Product/Sprint Backlog, User Stories, First Iteration |
| Sprint 2 | Outline & Detail Use Cases, UI Mockups, Second Iteration Product Release |
| Sprint 3 | Activity/Analysis/Sequence/ERD diagrams, 3rd Product Release |
| Sprint 4 | n/a |
| Sprint 5 | n/a |
| Sprint 6 | n/a |
| Sprint 7 | n/a |

### Releases

|  |  |  |
| --- | --- | --- |
| Release | Status | Objective |
| Product Release 1 | Demo | Initial project setup and program layout |
| Product Release 2 | Demo | Data modeling, database setup, and second semi-functioning web app |
| Product Release 3 | Demo | Final Database design/build along with a working prototype |
| Product Release 4 | Demo | TBD |
| Product Release 5 | Demo | TBD |
| Product Release 6 | Beta | TBD |
| Product Release 7 | Production | TBD |

### Project Schedule

|  |  |  |
| --- | --- | --- |
| Project Schedule | Target Dates | |
| **Start** | **End** |
| Sprint 1 | 30/08/16 | 29/09/16 |
| Product Release 1 Demo |  | 04/09/16 |
| Sprint 2 | 01/10/16 | 27/10/16 |
| Product Release 2 Demo |  | 01/11/16 |
| Sprint 3 | 02/11/16 | 01/12/16 |
| Product Release 3 Demo (Viable Product) |  | 06/12/16 |
| Sprint 4 | TBD | TBD |
| Product Release 4 Demo |  | TBD |
| Sprint 5 | TBD | TBD |
| Product Release 5 Demo |  | TBD |
| Sprint 6 | TBD | TBD |
| Product Release 6 Demo |  | TBD |
| Sprint 7 | TBD | TBD |
| Final Product Delivery |  | TBD |

### Project Resourcing

#### Staffing Plan

N/A

#### Resource Acquisition Plan

N/A

#### Training Plan

The Singularity team members will learn the skills needed in order to take on certain project criteria that need to be completed. Learning will be done in anticipation of, and during, assigned work in order to successfully and correctly execute the job.

### Budget

There are zero budget costs associated with this project, therefore no costs need to be allocated for the different areas of work being done on this project. Team members will track hours for the work they complete.

## Iteration Plans

See Sprint Backlogs

## Project Monitoring and Control

### Requirements Management Plan

The team members will be assigned a section to gather requirements for based on the user stories they covered in Deliverable 1. If a team member needs additional help, then additional members of the team will help out.

### Schedule Control Plan

We will monitor our progress using the Scrum and Kanban boards found in Jira. If a team member is not adhering to the schedule, then we will bring it up via Slack or during our weekly scrum meetings to see if there are any obstacles blocking the team member from finishing their assigned work. This will prevent our team from derailing from our scheduled tasks.

### Budget Control Plan

N/A

### Quality Control Plan

The team will conduct quality assurance by having weekly reviews where team members may present the work that they have been focused on. During this the team will discuss any changes or adjustments that will increase the quality of our work.

### Reporting Plan

The team submits reports on a bi-weekly basis to our client, mentor and professor. Also, verbal reports are given during class on Thursdays.

### Measurement Plan

[Enclosed by reference.]

N/A

## Risk Management Plan

**Client Risks:**

* Miscommunication
* Lack of communication
* Change in expectations regarding the original scope of the project

Solution: In order to mitigate client risk, we will continue sending a bi-weekly report to the client to ensure that we are on the same page. Also, when we have client meet-ups we will have at least two team members present. In a lapse of communication, the team will take any necessary actions to regain contact with the client.

**Team Risks:**

* Mishandling of deliverable submissions
* Underestimating deliverable work
* Inability to produce quality / lack of knowledge
* Unexpected loss of team members (not necessarily by death)

Solution: (See Group Contract)

## Close-out Plan

[Describe the activities for the orderly completion of the project, including staff reassignment, archiving of project materials, post-mortem debriefings and reports, and so forth.]

# Technical Process Plans

## Development Case

[Enclosed by reference.]

## Methods, Tools, and Techniques

[List the documented project technical standards, etc., by reference:

* Business Modeling Guidelines
* User Interfaces Guidelines
* Use-Case-Modeling Guidelines
* Design Guidelines
* Programming Guidelines
* Test Guidelines
* Manual Style guide]

## Infrastructure Plan

[Enclosed by reference]

## Product Acceptance Plan

[Enclosed by reference]

# Supporting Process Plans

## Configuration Management Plan

[Enclosed by reference]

## Evaluation Plan

[As part of the **Software Development Plan,** this describes the project’s plans for product evaluation, and covers the techniques, criteria, metrics, and procedures used for evaluation— this will include walkthroughs, inspections, and reviews. Note that this is in addition to the Test Plan, which is not enclosed in the **Software Development Plan**.]

## Documentation Plan

[Enclosed by reference.]

## Quality Assurance Plan

[Enclosed by reference.]

## Problem Resolution Plan

[Enclosed by reference.]

## Subcontractor Management Plan

[Enclosed by reference.]

## Process Improvement Plan

[Enclosed by reference.]

# Additional Plans

[Additional plans if required by contract or regulations.]

# Annexes

[Additional material of use to the reader of the **Software Development Plan**.]

# Index