SE 4351.001 Software Requirements

Fall 2014

Project Management Plan



[*http://www.utdallas.edu/~axp120531/SE4352/*](http://www.utdallas.edu/~axp120531/SE4352/)

**Team Members**

|  |  |  |  |
| --- | --- | --- | --- |
| Joe Brown  *jsb100120* | Desmond Lee  *dcl102020* | Giuseppe Mastrolorenzo  *mxg106320* | Michael Raibick  *mxr110530* |
| Ryan Chen  *rxc109520* | Robert Lockwood  *rll092020* | Blessing Osakue  *boo102020* | Oscar Reyes  *oxr110330* |
| Travis Chun  *twc101020* | Michael Mugger  *mxm121531* | Andrew Pohlmann  *axp120531* | James Williams  *jxw110730* |
|  |  |  | Bennilyn Quek  *bxq120430* |

**Revision History**

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Comments** |
| 1.0.0 | 09/02/2014 | Initial version of Preliminary Project Management Plan. |
| 1.1.0 | 09/04/2014 | Updated project plan formatting to conform to template and edited content sections for structure. |
| 1.1.1 | 09/04/2014 | Formatted sentence structures in §1.1.1, §1.1.2, §2.3 for clarity. |
| 1.1.2 | 09/04/2014 | Revised placeholders for printing. |
| 1.1.3 | 09/08/2014 | Filled in §2.2.1, §2.4.1, §3, §4.2, §4.3 |
| 1.1.4 | 10/1/2014 | Added ToC  Updated §1.2, §2.4.1,§5.2 |
| 1.2 | 10/16/2014 | Updated §1.2, §2.4.1,§5.2 |
| 1.3 | 12/01/2014 | Updated §1.2, §2.4.1,§5.2  Final Check |

**Table of Contents**

[1. Introduction 5](#_Toc404690735)

[1.1 Project Overview 5](#_Toc404690736)

[1.1.1 Project Background 5](#_Toc404690737)

[1.1.2 Project Goals 5](#_Toc404690738)

[1.2 Deliverables 6](#_Toc404690739)

[1.3 Document Evolution 6](#_Toc404690740)

[1.4 References 7](#_Toc404690741)

[1.5 Definitions, Acronyms, Abbreviations 7](#_Toc404690742)

[2. Project Organization 8](#_Toc404690743)

[2.1 Process Model 8](#_Toc404690744)

[2.2 Organizational Structure 8](#_Toc404690745)

[2.2.1 Team Organization 8](#_Toc404690746)

[2.3 Organizational Boundaries and Interfaces 10](#_Toc404690747)

[2.4 Project Responsibilities 11](#_Toc404690748)

[2.4.1 Phase 1 Team Roles and Responsibilities 11](#_Toc404690749)

[3. Managerial Process 13](#_Toc404690750)

[3.1 Management Objectives and Priorities 13](#_Toc404690751)

[3.2 Assumptions, Dependencies and Constraints 13](#_Toc404690752)

[3.2.1 Assumptions 13](#_Toc404690753)

[3.2.2 Dependencies 13](#_Toc404690754)

[3.2.3 Constraints 14](#_Toc404690755)

[3.3 Risk Management 14](#_Toc404690756)

[3.4 Monitoring and Controlling Mechanisms 16](#_Toc404690757)

[4. Technical Process 18](#_Toc404690758)

[4.1 Methods, Tools and Techniques 18](#_Toc404690759)

[4.2 Software Documentation 19](#_Toc404690760)

[4.3 Project Support Functions 19](#_Toc404690761)

[5. Work Elements, Schedule and Budget 20](#_Toc404690762)

[5.1 Work Elements 20](#_Toc404690763)

[5.2 Schedule 20](#_Toc404690764)

[5.3 Budget 22](#_Toc404690765)

## 1. Introduction

## 1.1 Project Overview

## 1.1.1 Project Background

As people get older they tend to experience difficulties in either cognitive or sensory function. These difficulties can turn an independent and free-spirited person’s life into a continuous set of daily challenges. Assistive technology processes sensory data to provide meaningful information to the user. Today’s assistive technologies (known as aids) are expensive, trivial and bulky. Persons who have multiple difficulties require multiple devices. There is a growing need for advanced assistive technologies that are easy to use and convenient for the user. The solution is the use of a device many people already own: the smartphone. Smartphone-based aids do more than traditional assistive devices. For example, a hearing aid should recognize speech to create a visual image. A speaking aid should generate speech from visual data. Finally, a memory aid should produce visual queues based on a variety of input.

## 1.1.2 Project Goals

Team Total Recall’s Project Goal is ***to create an AAC application, running on an Android mobile device that is capable of enhancing a user’s communication abilities who is suffering from stage 3 dementia.***

### 

### 

## 1.2 Deliverables

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Phase**  **No.** | **Phase**  **Name** | **Deliverables** | **Internal Review**  **Date** | **Delivery**  **Date** |
|  | *Initial* | 1. Preliminary PMP | 09/03/2014 | 09/04/2014 |
| **1** | *Interim* | 1. Evolving PMP 2. Evolving WRS 3. Evolving PPT 4. Prototype Mock-ups | 09/29/2014 | 09/30/2014 |
|  | *Final* | 1. Evolving PMP 2. Evolving WRS 3. Evolving PPT 4. Evolving UM 5. Preliminary Prototype Mockups | 10/15/2014 | 10/16/2014 |
| **2** | *Interim* | 1. Evolving PMP 2. Evolving WRS 3. Evolving Process Specification Document 4. Evolving Vision Document 5. Appendix - Glossary 6. Appendix - References | 10/17/2014 | 11/11/2014 |
| **2** | *Final* | 1. Final PMP 2. Final WRS 3. Final Vision Document 4. Final Process Specification 5. Final User Manual 6. Final Prototype 7. Final Presentation 8. Final Appendix | 11/12/2014 | 12/02/2014 |

## 1.3 Document Evolution

This is a living document that is updated periodically. The document will be updated in response to feedback from the review process defined in the managerial processes (§3.4.1). This document is to be managed solely in the Google Drive repository established for the team. When changes are made the changes must be documented in the revision history table found at the beginning of this document. Changes must be dated, versioned, described and stamped with the team member who made the changes. The current version is specified on the bottom right of the document on all pages.

The versioning system used by the revision history table conforms to the semantic versioning scheme. The version numbers are indicated as a three decimal numerical in the form of “X.Y.Z”. All X, Y and Z are integers in the range zero to nine inclusive. Where X represents the major version, Y represents the minor version and Z represents patches. Major versions are incremented when the minor version number is exhausted by reaching nine and resetting to zero. Minor versions are incremented when content is added or removed. Patch versions are indicated when sentence structure changes are made but no content is added or removed.

## 1.4 References

Please see the document **References** in the Appendix.

## 1.5 Definitions, Acronyms, Abbreviations

Please see the document **Glossary** in the Appendix.

## 2. Project Organization

## 2.1 Process Model

The organization will be employing the Spiral Mode. Due to the emphasis on requirements analysis and negotiation, this model serves the organizational requirement of close client collaboration well.

## 2.2 Organizational Structure

The organizational structure of three teams and 4 team members each. The project leader is not directly part of a team, but will form a fourth pseudo group that involves the other team leaders. This pseudo team is in place to encourage high cohesion amongst management elements for more effective work unit scheduling.

## 2.2.1 Team Organization

***Phase 1 - Interim***

|  |  |
| --- | --- |
| **Team**  **No.** | **Team**  **Members** |
| ***-*** | 1. Andrew Pohlmann (Phase Lead) |
| ***1*** | 1. Michael Raibick (Team Lead) 2. Ryan Chen 3. Travis Chun 4. Oscar Reyes |
| ***2*** | 1. Michael Muggler (Team Lead) 2. Bennilyn Quek 3. Desmond Lee 4. Robert Lockwood |
| ***3*** | 1. Blessing Osakue (Team Lead) 2. Joe Brown 3. James Williams 4. Giuseppe Mastrolorenzo |

***Phase 1 - Final***

|  |  |
| --- | --- |
| **Team**  **No.** | **Team**  **Members** |
| ***-*** | 1. Michael Raibick (Phase Lead) |
| ***1*** | 1. Andrew Pohlmann (Team Lead) 2. Michael Muggler 3. James Williams |
| ***2*** | 1. Michael Raibick (Team Lead) 2. Joe Brown 3. Desmond Lee 4. Oscar Reyes |
| ***3*** | 1. Bennilyn Quek (Team Lead) 2. Blessing Osakue 3. Ryan Chen 4. Robert Lockwood 5. Giuseppe Mastrolorenzo 6. Travis Chun |

***Phase 2 - Interim***

|  |  |
| --- | --- |
| **Team**  **No.** | **Team**  **Members** |
| ***-*** | 1. Bennilyn Quek (Phase Lead) |
| ***1*** | 1. Andrew Pohlmann (Team Lead) 2. Michael Muggler 3. James Williams |
| ***2*** | 1. Desmond Lee (Team Lead) 2. Joe Brown 3. Michael Raibick 4. Oscar Reyes |
| ***3*** | 1. Ryan Chen (Team Lead) 2. Blessing Osakue 3. Michael Raibick 4. Robert Lockwood 5. Giuseppe Mastrolorenzo 6. Travis Chun |

***Phase 2 - Final***

|  |  |
| --- | --- |
| **Team**  **No.** | **Team**  **Members** |
| ***-*** | 1. James Williams (Phase Lead) |
| ***1*** | 1. Joe Brown (Team Lead) 2. Michael Muggler 3. Bennilyn Quek |
| ***2*** | 1. Michael Raibick (Team Lead) 2. Andrew Pohlmann 3. Desmond Lee 4. Oscar Reyes |
| ***3*** | 1. Ryan Chen (Team Lead) 2. Blessing Osakue 3. Michael Raibick 4. Robert Lockwood 5. Giuseppe Mastrolorenzo 6. Travis Chun |

## 2.3 Organizational Boundaries and Interfaces

Each team is intended to function as a self-contained entity that can complete its work units independently of other teams. Team leaders will function as interfaces to the project leader in management meetings for progress reviews and new workload assignments. If necessary, team leaders will also coordinate with other team leaders and the project leader for the purposes of synchronizing content production toward an aggregated work unit. The project leader is not directly part of a team, but will work closely with all team leaders for workflow synchronization purposes and clarification of project vision and direction. The project leader will also function as a liaison as necessary between all teams in the event that more precise communication is necessary.

## 2.4 Project Responsibilities

## 2.4.1 Phase 1 Team Roles and Responsibilities

*Phase 1 - Interim*

|  |  |  |
| --- | --- | --- |
| **Team** | **Role** | **Responsibilities** |
| 1 | Requirements Engineering | 1. Exploration of Domain, Stakeholder, Functional, Non-Functional Objectives. 2. Update WRS. |
| 2 | Requirements Engineering | 1. Exploration of Non-Functional Requirements 2. Update WRS. 3. Generate preliminary Prototype. |
| 3 | Requirements Engineering | 1. Exploration of Functional Requirements 2. Update WRS. |
| MG | Management | 1. Update PMP. 2. Generate preliminary PPT. |

*Phase 1 - Final*

|  |  |  |
| --- | --- | --- |
| **Team** | **Role** | **Responsibilities** |
| 1 | Requirements Engineering | 1. Create system specifications. |
| 2 | System Design | 1. Create UI of the front-end |
| 3 | Documentation | 1. Generate the Initial User Manual |
| MG | Management | 1. Update PMP, PP |

### 2.4.2 Phase 2 Team Roles and Responsibilities

*Phase 2 - Interim*

|  |  |  |
| --- | --- | --- |
| **Team** | **Role** | **Responsibilities** |
| 1 | System Design / Development | 1. Developing System Architecture 2. Implement the System 3. Generate Product Diagrams. |
| 2 | System Design / Development / Documentation. | 1. Develop System UI 2. Implement the System |
| 3 | Documentation | 1. Develop User Manual 2. Develop Appendix |
| MG | Management | 1. Develop Evolving Vision Document 2. Develop Appendix |

*Phase 2 - Final*

|  |  |  |
| --- | --- | --- |
| **Team** | **Role** | **Responsibilities** |
| 1 | System Development / Documentation. | 1. Finalize Prototype 2. Finalize WRS. |
| 2 | System Development / Documentation. | 1. Finalize Prototype 2. Finalize all Product and Process Diagrams |
| 3 | Documentation. | 1. Finalize User Manual 2. Finalize Appendix |
| MG | Management | 1. Finalize PMP 2. Finalize Vision Document 3. Finalize Process Specification Document 4. Finalize PPT |

## 3. Managerial Process

## 3.1 Management Objectives and Priorities

1. Accountability: Ensuring that weekly work unit deadlines are met.
2. Accuracy: Ensuring that work units are traceable to end-phase deliverables requirements.
3. Efficiency: Ensuring that no excessive or redundant work is performed.
4. Quality Assurance: Ensuring that all work units completed are free of defects.
5. Communication: Ensuring low latency communication between and within teams.

## 3.2 Assumptions, Dependencies and Constraints

## 3.2.1 Assumptions

1. Team members will be cognizant of project schedule deadlines.
2. Team members will stay healthy and remain physically capable of carrying out assigned duties.
3. Team members will respond to communications from other team members in a timely fashion.
4. Team members remain motivated through the duration of the project.
5. Team members will always have access to the necessary tools to complete their assigned work units.
6. The group will not lose any team members.

## 3.2.2 Dependencies

1. Management will effectively communicate the project goals.
2. Communication mechanisms to maintain low latency communications in the organizational structure
3. Management being effective at maintaining linear progression of progress.
4. Team members not procrastinating and delaying assigned work units.

## 3.2.3 Constraints

1. Personal schedule conflicts between team members.
2. Personal lives of team members.

## 3.3 Risk Management

|  |  |  |
| --- | --- | --- |
| **Risk** | **Impact** | **Mitigation** |
| Inaccurate Estimations about project costs, scheduling, and time to complete | These bad estimates can end up sinking a project into a sea of budget overruns, delayed deliverables, and wasted effort that can potentially end up in a cancellation of a project. | Apply crash techniques to speed up under-performing work units. Work overtime to ensure that work units get completed on time. Modify the schedule as necessary to apply more man hours to work units that are behind. |
| Bad Requirements Gathering | Bad requirements gathering in a project ensures that the resulting software architecture will not be compliant with what the client truly needs. This situation is sometimes not seen until later in the project and, when discovered, can lead to critical faults that threaten the success of the project. | Hire a requirements engineer to elicit the proper set of requirements from the client. |
| Executing Behind Schedule | Executing behind schedule virtually assures missing deadlines, wasting project resources, and provoking negative client feedback. The worst case of this scenario is that the client cancels the project. | We have taken a multifaceted approach in expecting and mitigating the risks of running behind schedule. Ample time has been set aside to collaborate with the customer on creating a detailed list of specifications and expectations with the project. Time has been set aside to review the project at regular intervals with the customer and additional time has been set aside to handle possible changes or issues brought up during these meetings. The project is internally scheduled to complete early so schedule setbacks don’t push us beyond the hard deadline. A change management system is in place to reduce the impact of changes. We believe these mitigations will be sufficient to handle most high-probability or high-impact scheduling situations that may arise. |
| Unforeseen Technical Complexity | Unforeseen technical complexity could introduce delays in implementation of the program, or even worse uncover flaws in the requirements. | Our project manager and developers will ask some questions during specification-gathering to ensure common sources of technical complexity are covered early. Integrations with third party systems, for example using Active Directory to handle user management, need to be identified during the specification phase. Change requests relating to third party integrations after the completion of the design phase will be denied unless an agreed-upon amount of time can be added to the five-month deadline. |
| Data Loss | Data loss can easily derail current project progress and make blown deadlines a guaranteed reality. Man-hours spent backtracking and redoing work units could easily lead to serious delays in the schedule and trigger another risk scenario “Executing behind schedule.” Regeneration of the lost data could also produce inaccurate or incorrect work that could cause future problems. | Backup database nightly, have a slave database ready at a different location. |
| Scope Changes | Scope changes can cause serious changes in the functional and non-functional requirements of the program. Scope changes almost always impact scheduled tasks and work-units because of the refactoring of the scope into all phases of the development process. | Expand timeline to accommodate additions or reject additions depending upon contract restrictions, feasibility, meetings with the customer, etc. |
| User Interface Usability Issues | The client may find the program difficult to use and then issue new requirements to “simplify” program operation. The client may not see the program as fulfilling its requirements and refuse payment of services. | We will make use of tools such as UserTesting.com to ensure our application is easy to use and follows today’s best practices for user interface design. |
| Personnel Loss | Project work units could be delayed. Other team members may become strained from having to take on the extra work from the departing employee. The project even may suffer schedule delays if management cannot replace the lost manpower. | Hire a qualified replacement, divide work between remaining employees while replacement is located. |

## 3.4 Monitoring and Controlling Mechanisms

Living Announcement and Meeting Agenda documents will be constantly updated on Google Drive as a means to support project vision and maintain schedule work unit deadlines.

Doodle will be used to schedule team member meetings.

Glassboard will be used as a group collaboration platform.

Github will be used for version control for all code.

### 

### 

### 

### 

## 4. Technical Process

## 4.1 Methods, Tools and Techniques

The following methods, tools and techniques will be used:

|  |  |  |
| --- | --- | --- |
| **Method** | **Tool** | **Technique(s)** |
| Communication | Email  Doodle Glassboard | Emails are used for direct communication between group members. Doodle will be used to schedule team member meetings. Glassboard will be used as a group collaboration device. |
| Storage | Google Drive | Any document pertaining to a deliverable must be stored on the team accessible Google Drive share folder. Including any preliminary or draft documents. All documents must be shared with the entire team. |
| Version Control | Github | Github will be the primary tool for storing the codebase, documenting changes and typing the code documentation. The user manual is to be stored in the Google Drive. The Github repository is exclusively for code and code related materials. |
| Documentation | Astah UML Editor  RE Tools  Star UML | Any documentation pertaining to the software must be done in Astah, Star, or RE Tools and submitted in the Google Drive. Do not place UML files under version control. |
| Visual Editing | MS Paint  Visio | Prototype Mockups and graphics will be drawn using these tool. |
| System Development | Ionic  JavaScript  HTML5  CSS5 | System development will be done using the Ionic SDK platform, which uses JavaScript, HTML5, and CSS5 to create mobile applications. |

## 4.2 Software Documentation

The software architecture will be documented with UML using Astah, Star, and RE Tools diagrams. A user manual will be created for the application that shows graphical images to demonstrate how to use the system.

## 4.3 Project Support Functions

Living Announcement and Meeting Agenda documents will be constantly updated on Google Drive as a means to support project vision and maintain schedule work unit deadlines. Doodle will be used to schedule team member meetings. Glassboard will be used as a group collaboration device. Github will be used for version control for all code.

## 5. Work Elements, Schedule and Budget

## 5.1 Work Elements

1. Project Management Plan
2. WRS
3. Process Specification
4. Product Specification (Vision)
5. Prototype
6. User Manual
7. PowerPoint Presentation
8. Appendix

## 5.2 Schedule

**Phase 1 Interim**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Deliverable(s)** | **Team**  **Assignments** | **Assign Date** | **Internal Due Date** | **Official Due Date** |
| 1. Evolving WRS | **RE Team 1:** WRS §2.1  **RE Team 2:** WRS §2.3  **RE Team 3:** WRS §2.2 | 09/8/14 | **09/14/14** | 09/30/14 |
| 1. Evolving WRS | **RE Team 1:** §3.1  **RE Team 2:** §2.3  **RE Team 3:** §2.2 | 09/15/14 | **09/21/14** | 09/30/14 |
| 1. Evolving PPT | **RE Team 1:** Traceability | 09/22/14 | **09/28/14** | 09/30/14 |
| 1. Evolving WRS | **RE Team 3:** §3.2 | 09/22/14 | **09/28/14** | 09/30/14 |
| 1. Evolving WRS | **RE Team 2:** §3.1 | 09/22/14 | **09/28/14** | 09/30/14 |
| 1. All Deliverables | **Management** | 09/29/14 | **09/29/14** | 09/30/14 |

**Phase I Final (Due 10/16/2014)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Deliverable(s)** | **Team**  **Assignments** | **Assign Date** | **Internal Due Date** | **Official Due Date** |
| 1. Evolving WRS | **Architecture Team:**   1. Develop System Specifications | 10/01 | **10/8** | 10/16 |
| 1. Evolving Prototype | **UI/UX Team**   1. Develop prototype wireframes from specifications. | 10/09 | **10/13** | 10/16 |
| 1. Evolving User Manual | **UM/Testing Team**   1. Develop user manual. | 10/11 | **10/14** | 10/16 |
| 1. Evolving PMP 2. Evolving PP | **Management**   1. Update PMP, PP | 10/15 | **10/15** | 10/16 |

**Phase 2 Interim (Due 11/11/2014)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Deliverable(s)** | **Team**  **Assignments** | **Assign Date** | **Internal Due Date** | **Official Due Date** |
| 1. Evolving Prototype | **UI/UX Team**:   1. Develop System UI 2. Generate Appendix. | 10/16 | **10/25** | 11/11 |
| 1. Evolving Prototype | **Architecture Team:**   1. Develop System Architecture 2. Implement the System. 3. Generate Diagrams. | 10/16 | **10/25** | 11/11 |
| 1. Evolving User Manual | **UM/Testing Team**   1. Develop User Manual | 10/16 | **10/25** | 11/11 |
| 1. Evolving Prototype | **UI/UX + Architecture Team**   1. Develop Prototype | 10/27 | **11/10** | 11/11 |
| 1. Evolving Vision Document | **UM/Testing:**   1. Develop Vision Document | 10/27 | **11/10** | 11/11 |
| 1. All Deliverables | **Management** | 11/10 | **11/10** | 11/11 |

**Phase 2 Final (Due 12/05/2014)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Deliverable(s)** | **Team**  **Assignments** | **Assign Date** | **Internal Due Date** | **Official Due Date** |
| 1. Evolving Prototype | **Architecture Team**   1. Finalize prototype. 2. Finalize Diagrams. | 11/12 | **11/15** | 12/05 |
| 1. Evolving User Manual | **UM/Testing:**   1. Finalize User Manual | 11/12 | **11/15** | 12/05 |
| 1. Evolving Vision Document | **UI/UX**   1. Finalize Vision Document. | 11/12 | **11/15** | 12/05 |
| 1. Appendix | **UM/Testing:**   1. Finalize Appendix | 11/16 | **12/01** | 12/05 |
| 1. Process Specification Document | **Management**   1. Finalize Process Specification | 11/16 | **12/01** | 12/05 |
| All Deliverables | Management | 12/02 | **12/05** | 12/05 |

## 5.3 Budget

No budget is allotted for this project.