CptS 484 - Moderamen

Software Project Management Plan

Austin Marino, Cole Bennett, Freya Varez, Sean Cornia

Table of Contents

Introduction

Project Overview

Our project is to develop a mobile application currently titled Moderamen that will help visually impired people with navigating interiors of buildings. This document, Software Project Management Plan (SPMP), that breaks down the people behind this application, their roles and their plan to fulfill the project. The document also defines in written the deliverables along with schedule that we wish to deliver them and work on the project.

Project Deliverables

The project has various deliverables separated by a phase I and phase II. Currently, only the project phase I deliverables are listed and phase II deliverables will be listed when they are defined.

Project Phase I Deliverables Listed in Order of Submission Date

- Software Project Management Plan
- World Requirement Specification
- Software Requirements Specification - written descriptions of resolving issues within functional and nonfunctional requirements
- Prototype
- User Manual
- Meeting Records
- Prestation for phase I deliverable

Evolution of Document

The plan is currently used for the development of phase I of Moderamen. For the development of phase I the plan currently holds within its scope a general description of Moderamen, scheduling along with deliverables, a clear outline of the organization of our team, and our current evaluated risks. Throughout the project of SPMP will evolve so that it remains an information source that translates our technical progress into a readable format, which in turn will cause different versions to be mindful of. Risk Management, Monitoring and Controlling Mechanisms, and Technical Progress are the expected ones to receive dramatic evolutions throughout the SPMP's lifespan within phase I.

As a reminder the current version is 0.1.0. When the description of phase II has been provided, the SPMP will have its most radical change with major additions to our deliverables along with our scheduling.

References

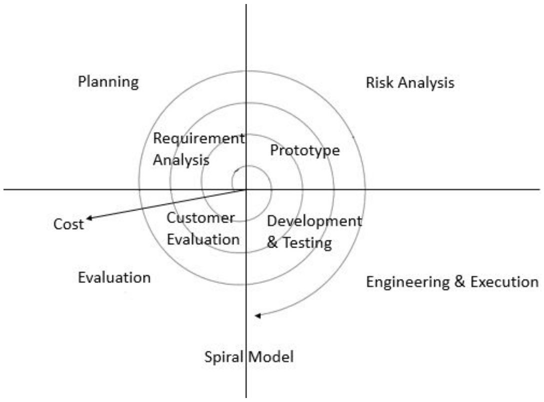
Definitions, Acronyms, & Abbreviations

SPMP Software Management Plan
WRS World Requirement Specification
RE Requirements Engineering

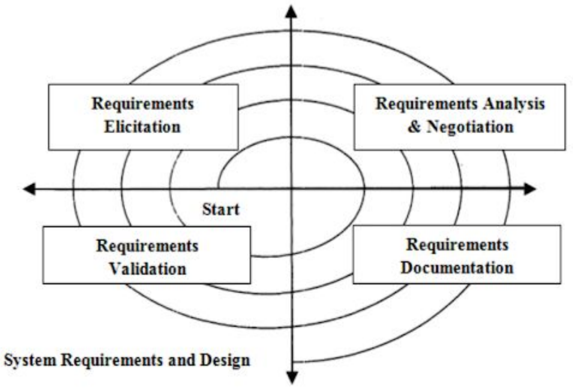
Project Organization

Process Model

The Moderamen team uses the Spiral software development method because it's iterative by nature like the requirements engineering process. This fundamental structural similarity between the two processes allows them to work well in unison during the development of software. Our team follows the RE (featured on the right) process during the "Planning: Requirements Analysis" section of the Spiral method (featured on the left). Additionally, the Spiral method places an emphasis on risk analysis inorder to reduce any unwanted or unforeseen negative outcomes from arising. There is also a review period at the end of each iteration to reevaluate what new features or updates should be proposed for the next iteration, RE; which avoids, wasted resources and money.



Spiral Model[1]


Requirements Elicitation
Requirements Analysis & Negotiation
Start
Requirements Validation
Requirements Documentation
System Requirements and Design

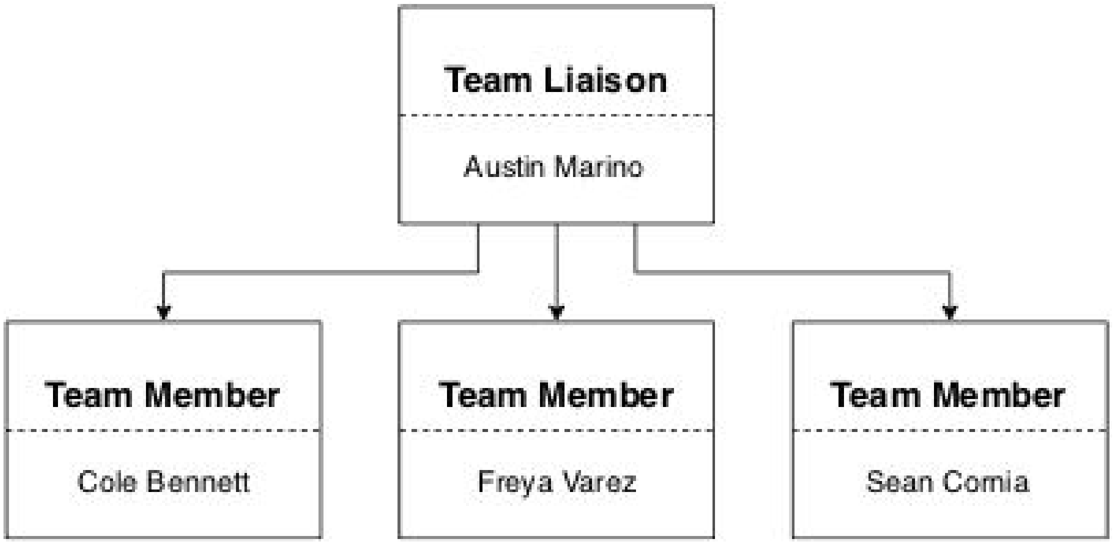
RE Process[2]

[1] Naveen. "What Is Spiral Model in Software Testing and What Are Advantages and Disadvantages of Spiral Model." Testing Freak, January 30, 2015. http://testingfreak.com/spiral-model-software-testing-advantages-disadvantages-spiral-model/.

[2] "Requirements Engineering Processes, Tools/Technologies & Methodologies." Google Sites. Accessed August 5, 2019. https://sites.google.com/site/richchihleese/home/se-research/requirements-engineering-processes-toolstechnologies--methodologies?tmpl=/system/app/templates/print/&showPrintDialog=1.

Organizational Structure

The Moderamen team is composed of four individuals and thus does not need a complex hierarchy. We rely on all four individuals working in tandem with one another to ensure we produce the highest quality of work possible. The central oversight comes from the team liaison, Austin, who ensures that the whole team is staying on the outlined schedule the team unilaterally agreed on. Each major milestones workload is equality partitioned amongst the group. During our weekly meetings we conduct code/work reviews to ensure that each member of the team met their goals. The image below shows the minimalist structure the group is broken up into.



Organizational Boundaries & Interfaces

Austin Marino - Team Liaison / Developer

Austin's role is as the team liaison and a software developer. He communicates with all stakeholders, such as Bolong Zeng, and parlays project information back and forth between his team and outside clients. When Austin is not communicating with clients he is assisting his team in the development of their projects documents and software.

Cole Bennett - Developer

Cole's role is as a software developer. He spends most of his time working alongside his other teammates developing software and creating and revising the original documents that guide their development process.

Freya Varez - Developer

Freya’s role is as a software developer. She spends most of his time working alongside her other teammates developing software and creating and revising the original documents that guide their development process.

Sean Cornia - Developer

Seans role is as a software developer. He spends most of his time working alongside his other teammates developing software and creating and revising the original documents that guide their development process.

Project Responsibilities

All the team members will be involved in all phases of the project life cycle.

Managerial Process

Management Objectives & Priorities

Moderamens is, first and foremost, a medical device - classified under FDA guidelines of a *Mobile Medical Application*.[3] As the software must follow various regulations before deployment as well as requirements set forth by stakeholders. A liaison will remain as a contact with all parties to assure new information and guidelines are communicated with the development team.

The development team must design and implement a user-friendly, extensible application; thus this team will be less focused on agility - and more on testing and redesign. A team lead will assist in assuring all guidelines set forth by stakeholders are met.

Assumptions, Dependencies, & Constraints

Assumptions:

1. Moderamen must be usable visually impaired users (primary stakeholders) as well as non-visually impaired users (caretakers, staff and emergency response).
2. Moderamen is primarily used indoors or between buildings.
3. Funding is approved.
4. Necessary equipment and software is available.
5. A team of 5 is available for development and communication.

Dependencies:

[3] “Mobile Medical Applications” (United States Food & Drug Administration, September 4, 2018), https://www.fda.gov/medical-devices/digital-health/mobile-medical-applications.

1. Moderamen must be available to a wide variety of users on different operating systems. Moderamen must also continue to work with updates to underlying systems.
2. API endpoints are available to gather necessary information such as routes, language translations etc.

Constraints:

1. Moderamen must be available across language barriers and as ubiquitous as possible for users.
2. Moderamen must follow FDA guidelines.

Risk Management

	Risk	Likelihood	Description
Market Risks	Market saturation	Medium	One or more applications are available that compete with ours causing financial non-viability.
Financial Risks	Over-budget	Low	Insufficient planning or changes in requirements cause the project to run over-budget.
Technology Risks	Data corruption or loss	Low	Important software or data loss due to system corruption or improper version control.
	Insufficient hardware or OS	Low	Final app is unsupported on customers phones.
People Risks	FDA disapproval	Low	Application does not follow FDA regulations and guidelines - is not approved for deployment.
	Software leads to injury	Low	Software is unable to correctly identify obstacles causing tripping or injury.

Monitoring & Controlling Mechanisms

Risk	Monitoring and controlling mechanisms.
Market saturation	Significant preemptive research will be done before development is progressed to assure that this product has a financially viable customer base.

Over-budget	Planning will be done before development is progressed to determine the products requirements and budget.
Data corruption or loss	GIT version control will save all necessary data to the cloud as well as local repositories and/or servers.
Insufficient hardware or OS	Multiple versions of the app will be created and tailored to popular operating systems.
FDA disapproval	Research will be done during planning to determine the apps feasibility with respect to FDA restrictions. The team liaison will be in contact with FDA officials to ensure these regulations are met.
Software leads to injury	Significant (re-)testing will be done to assure the application has a low failure rate.

Technical Process

Methods, Tools, & Techniques

The project will be implemented using the Spiral software process model. Following the methodologies of Spiral, we will perform a review process at the end of each phase. Git (specifically GitLab) will be utilized as primary tool for enforcing this review process through branching. Semantic Versioning 2.0.0 will be utilized as the versioning scheme for the project, which entails that we conform to the *MAJOR.MINOR.PATCH* format.[4] Each phase, feature, and bugfix will hold their own branch:

- Each phase, indicating a major version increase, will have its own branch. All features (minor version increase) and bug fixes (patch version increase) associated with a phase will be merged into this branch.
- A formal review process will be performed for each feature and bugfix: A pull request to merge (or rebase) a feature/bugfix into a phase branch must be opened. All team members must review and approve the request for it to be accepted.
- At the end of each phase, an additional pull request must be opened to merge the phase branch into master.

Software Documentation

Documentation of the project will be present in the following ways:

1. All functions, classes, and variables of importance will be associated with concise and descriptive comments, formatted according to the industry standard conventions of the programming language(s) chosen for the project.

[4] Preston-Werner, Tom. “Semantic Versioning 2.0.0.” Semantic Versioning. https://semver.org (accessed September 6, 2019).

2. Markdown and PDF files will be utilized as the formats for documentation files in the project repository. The following documents will be included:
    a. A readme document containing general information and installation/build steps.
    b. A user manual document describing how to use the application from an incoming user's perspective.
    c. Software Requirements Specification defining functional and non-functional requirements, and also use cases of the application.

Project Support Functions

Technical support of the project will include testing, configuration control, and quality assurance:

1. Testing of the application will be derived from the project's requirements specifications.
2. Software configuration control will be facilitated by Git for program, requirement, design, and version release changes.
3. Quality assurance measures will be taken to validate that our application is adhering to the requirement specifications.

Work Elements, Schedule & Budget

Schedule

Team Meetings

We have meetings from 5:00pm to 6:00pm every wednesday.

Project Phase I

Preliminary Project Plan	September 8th, 2019
Checkup Meeting	September 18th, 2019
Final Submission	October 13th, 2019

Project Phase II

Checkup Meeting	October 30th, 2019
Final Submission	December 15th, 2019

Budget

Currently not available.