Master Project Plan

Mapper for Android, Twin Cities Edition

IE 5541

February 28, 2012

**Authors:**

Andrey Bruslavtsev

Ian De Silva

Jared Swanson

Jonathan Lee

Kristin Mead

Sriram Rajendran

Usha Kumar

Weston Platter

# Table of Contents

[1 PROJECT OVERVIEW 3](#_Toc213217780)

[1.1 Potential Market 3](#_Toc213217781)

[1.2 Project Team 3](#_Toc213217782)

[1.3 Milestones and Deliverables 3](#_Toc213217784)

[1.3.1 Project Milestones and Deliverables 3](#_Toc213217790)

[1.3.2 Product Milestones and Deliverables 3](#_Toc213217792)

[2 REQUIREMENTS 4](#_Toc213217788)

[2.1 Functional Requirements 4](#_Toc213217789)

[2.1.1 Application Views 4](#_Toc213217790)

[2.1.2 Location Functionality 5](#_Toc213217792)

[2.1.3 Search Functionality 6](#_Toc213217794)

[2.1.4 Routing Functionality 6](#_Toc213217796)

[2.1.5 Application Usability and User Interface 7](#_Toc213217798)

[2.2 Non-Functional Requirements 8](#_Toc213217808)

[2.3. Project Deliverable Requirements 9](#_Toc213217790)

[2.4 Project Process Requirements 9](#_Toc213217792)

[2.5 Limitations and Exclusions 10](#_Toc213217794)

[3 RISKS AND CONTINGENCIES 11](#_Toc213217788)

[3.1 Risk Analysis 11](#_Toc213217789)

[3.2 Risk Assessment 14](#_Toc213217808)

[3.3. Risk Severity 14](#_Toc213217790)

[4 PROJECT SCHEDULE 15](#_Toc213217788)

[5 COMMUNICATION PLAN 15](#_Toc213217788)

[5.1 SCRUM 15](#_Toc213217810)

[5.2 Information Resources 16](#_Toc213217828)

[6 QUALITY PLAN 17](#_Toc213217788)

[7 RESPONSIBILITIES Error! Bookmark not defined.18](#_Toc213217780)

[APPENDIX A: PROJECT SCHEDULE BREAKDOWN Error! Bookmark not defined.19](#_Toc213217780)

**List of Tables**

Table 1: Risk Assessment …14

Table 2: Iteration Timeline …15

Table 3: Communication Plan …16

Table 4: Key Responsibilities …18

# List of Figures

Figure 1: Risk Severity Matrix …15

Figure 2: Iteration Workflow Diagram …17

# Project Overview

## Skyway and tunnel systems are often complex and difficult to navigate, especially for newcomers.  Many newcomers simply avoid these systems as they do not know the layout.  With the increasing number of Android smartphones in the market, there is an opportunity to address these issues through an application on Google’s Android platform that leverages GPS and other information to aid in navigating these systems.  The Mapper for Android, Twin Cities Edition aims to aid the navigation of the Minneapolis Skyway System and the University of Minnesota (UMN) tunnel and skyway system -- termed the "Gopher Way."  It will be one application with options for both systems.

## The Minneapolis Skyway System is a collection of enclosed footbridges that connect buildings in Minneapolis.  It is an expansive system and is known as one of the largest such systems in the world.  Newcomers have claimed that they often get “turned around” within the skyway system, often going in circles and sometimes even missing a scheduled bus. Similarly, navigating the UMN tunnel and skyway system is a daunting task for newcomers to the university.  Not only do travelers have to frequently switch floors, going from the skyway to the tunnels or vice versa, but signs and maps of the system are often difficult to find and only show a limited view of the system.

## **Potential Market**

The application is being developed for Android Smartphone users who navigate the Minneapolis Skyway system or and University of Minnesota’s Gopher Way. The application is targeted towards all skyway and tunnel navigators regardless of their current knowledge of the systems. The application will be made available to customers through the Android Marketplace. Kristin Mead and Justin Swanson will act as the customer representatives in lieu of having distinct sponsors.

## Project Team

This project began as two separate teams. After discussing the requirements, both teams felt that significant portions of the project were the same, so the teams were merged. This merge will have a significant impact on our risk and the project as a whole, and these concerns will be further addressed in subsequent sections. In spite of these potential problems, merging teams allows us to engage more team members with tasks that they will be more comfortable with. Those who are software developers can focus on that aspect of the project while those who are not can focus on the deployment strategies and take responsibility for non-developmental tasks (rather than having to learn how to develop software).

## Milestones and Deliverables

The following deliverables and milestones have been defined and are discussed further in subsequent sections.

### Project Milestones and Deliverables

* Statement of Work (2/14/2012)
* Master Project Plan (2/28/2012)
* Status Report (4/3/2012)
* Presentation (4/24/2012)
* Project Report (5/1/2012)

### Product Milestones and Deliverables

* Personal Development Environments Set-up (3/8/2012)
* Complete Development Environment Set-up (3/9/2012)
* Framework (3/27/2012)
* Completion of Minneapolis Skyway Map (3/20/2012)
* Completion of UMN Map (4/19/2012)
* Completion of Testing (4/20/2012)
* Application with Required Functionality (5/1/2012)
* User Documentation (4/18/2012)

# Requirements

The Android application shall provide users with three main feature sets, (1) map navigation within the Minneapolis skyway system, (2) map navigation within the UMN campus, and (3) directory listing of nearby businesses.

Product functional requirements are requirements that must be implemented within the final Android application. All functional requirements are written as user-stories. Non-functional requirements are requirements that must implemented by the project team in order to ensure that the final deliverable is acceptable and meets internal quality standards. Project process deliverables indicate deliverables such as SOW, Master Project Plan, etc. And finally, project process requirements are requirements that must be followed by the project team in order to ensure timely coordination and communication.

## Functional Requirements

**2.1.1 Application Views**

1. F.1 Map View

|  |  |
| --- | --- |
| As a user, I want a map on my Android phone because it will help me navigate unfamiliar locations. | |
| *Dependencies:* N/A | *Successors:* R.F.2., R.F.3, R.F.8 |
| *Scope:* Required | *References:* N/A |

1. F.2 Overlay Campus Map

|  |  |
| --- | --- |
| As a user, I want a map of the UMN campus (including tunnels and skyways) on my Android phone because it will help me navigate through campus. | |
| *Dependencies:* R.F.1 | *Successors:* N/A |
| *Scope:* Required | *References:* N/A |

1. F.3 Overlay Minneapolis Skyway Map

|  |  |
| --- | --- |
| As a user, I want a map of the Minneapolis skyway system on my Android phone because it will help me navigate the skyway system. | |
| *Dependencies:* R.F.1 | *Successors:* N/A |
| *Scope:* Required | *References:* N/A |

1. F.4 Help View

|  |  |
| --- | --- |
| As a user, I want the application to provide some basic usage instruction because I may not know how to use it. | |
| *Dependencies:* N/A | *Successors:* N/A |
| *Scope:* Required | *References:* N/A |

1. F.5 Favorites View

|  |  |
| --- | --- |
| As a user, I want the application to provide the ability to store a list of my favorite destinations so that they can be easily retrieved. | |
| *Dependencies:* N/A | *Successors:* N/A |
| *Scope:* Required | *References:* N/A |

1. F.6 History View

|  |  |
| --- | --- |
| As a user, I want the application to provide the ability to store a list of destinations that I have searched for so that they can be easily retrieved. | |
| *Dependencies:* N/A | *Successors:* N/A |
| *Scope:* Required | *References:* N/A |

1. F.7 Search View

|  |  |
| --- | --- |
| As a user, I want the application to provide a search view so that I can enter specific points of interest that I am looking for. | |
| *Dependencies:* N/A | *Successors:* R.F.14 |
| *Scope:* Required | *References:* N/A |

**2.1.2 Location Functionality**

1. F.8 Identify Current Position

|  |  |
| --- | --- |
| As a user, I want the map to provide an indicator of where I currently am. This should be accurate to within 10 feet of my position and should be displayed within 30 seconds of initiation. | |
| *Dependencies:* R.F.1 | *Successors:* R.F.9, R.F.10, R.F.12 |
| *Scope:* Required | *References:* N/A |

1. F.9 Continuous Position Update

|  |  |
| --- | --- |
| As a user, I want the application tocontinuously update my location so that I can monitor my location as I move. | |
| *Dependencies:* R.F.8 | *Successors:* N/A |
| *Scope:* Required | *References:* N/A |

1. F.10 Signal Loss Notification

|  |  |
| --- | --- |
| As a user, I want the application to display a notification if the device is unable to determine my position due to loss of a GPS signal. | |
| *Dependencies:* R.F.8 | *Successors:* N/A |
| *Scope:* Required | *References:* N/A |

1. F.11 Load Appropriate Map

|  |  |
| --- | --- |
| As a user, I want the application to detect my current location upon initialization. If I am within the bounds of an overlay, the application should load the appropriate map. | |
| *Dependencies:* R.F.8, R.F.2, R.F.3 | *Successors:* N/A |
| *Scope:* Required | *References:* N/A |

1. F.12 Display List of Surrounding Businesses

|  |  |
| --- | --- |
| As a user, if I search for a business, I want the application to display a list of results near my current location. | |
| *Dependencies:* R.F.8 | *Successors:* N/A |
| *Scope:* Out of Scope | *References:* N/A |

**2.1.3 Search Functionality**

R.F.13 Destination Search

|  |  |
| --- | --- |
| As a user, I want the ability to search for a location and have its location displayed on the map. | |
| *Dependencies:* R.F.1, R.F.7 | *Successors:* R.F.15 |
| *Scope:* Required | *References:* N/A |

1. F.14 Out of Network Search Notification

|  |  |
| --- | --- |
| As a user, I want the application to display a notification if I search for a point that is outside of the UMN/Minneapolis network. This notification should allow me to cancel my request or transfer control to the native Android Maps application. | |
| *Dependencies:* R.F.1, R.F.7 | *Successors:* N/A |
| *Scope:* Required | *References:* N/A |

**2.1.4 Routing Functionality**

1. F.15 Route to Destination

|  |  |
| --- | --- |
| As a user, I want the application tocreate a route from my current location to a specified destination so that I can easily navigate to my destination. | |
| *Dependencies:* R.F.1, R.F.2, R.F.3, R.F.8, R.F.13, R.F.18 | *Successors:* R.F.16 |
| *Scope:* Required | *References:* N/A |

1. F.16 Persistent Route Display

|  |  |
| --- | --- |
| As a user, I want the application tocontinue to display my currentroutein the event that the GPS signal is lost. | |
| *Dependencies:* R.F.1, R.F.2, R.F.3, R.F.15 | *Successors:* N/A |
| *Scope:* Required | *References:* N/A |

1. F.17 Shortest Path to Destination

|  |  |
| --- | --- |
| As a user, I want the application to calculate the shortest path through the tunnels/skyway to my destination. | |
| *Dependencies:* R.F.8, R.F.18 | *Successors:* R.F.16 |
| *Scope:* Required | *References:* N/A |

1. F.18 Consider Skyway Closures

|  |  |
| --- | --- |
| As a user, I want the application to factor in building closures when calculating a route to my destination. | |
| *Dependencies:* R.F.18 | *Successors:* N/A |
| *Scope:* Optional | *References:* N/A |

**2.1.5 Application Usability and User Interface**

1. F.19 Indicate Floor Change

|  |  |
| --- | --- |
| As a user, I need a visual indicator to show when I should change floors, the direction of the change, and how many floors I need to traverse because I need to know where I’m going. | |
| *Dependencies:* R.F.2, R.F.3 | *Successors:* N/A |
| *Scope:* Optional | *References:* N/A |

1. F.20 Resume Previous State

|  |  |
| --- | --- |
| As a user, I want the application to restore its previous state upon interruption from another application. | |
| *Dependencies:* N/A | *Successors:* N/A |
| *Scope:* Required | *References:* N/A |

1. F.21 Responsiveness

|  |  |
| --- | --- |
| As a user, I want the application torespond to user-input within 5 seconds. | |
| *Dependencies:* N/A | *Successors:* N/A |
| *Scope:* Required | *References:* N/A |

1. F.22 Color Considerations

|  |  |
| --- | --- |
| As a user who is unable to view certain colors, I want the application to provide alternatives to color-coded indicators. | |
| *Dependencies:* N/A | *Successors:* N/A |
| *Scope:* Required | *References:* N/A |

1. F.23 Map Zooming and Panning

|  |  |
| --- | --- |
| As a user, I want the ability to zoom in and out and pan across the map so that the map can be seen from different perspectives. | |
| *Dependencies:* R.F.1 | *Successors:* N/A |
| *Scope:* Required | *References:* N/A |

R.F.24 Indicate Tunnel vs. Skyway on Campus

|  |  |
| --- | --- |
| As a user, I want a clear distinction between skyways and tunnelsso that it is easy to understand a specified route. | |
| *Dependencies:* R.F.2, R.F.3 | *Successors:* N/A |
| *Scope:* Required | *References:* N/A |

1. F.25 Change Screen Orientation

|  |  |
| --- | --- |
| As a user, I want to have the option of using the application in landscape mode or portrait mode. | |
| *Dependencies:* N/A | *Successors:* N/A |
| *Scope:* Optional | *References:* N/A |

1. F.26 Display Keyboard

|  |  |
| --- | --- |
| As a user with a phone that has no physical keyboard, I want the application to display a virtual keyboard when prompting for user-input text. | |
| *Dependencies:* R.F.7 | *Successors:* N/A |
| *Scope:* Required | *References:* N/A |

## Non-Functional Requirements

1. NF.1 Development Process

|  |  |
| --- | --- |
| This project shall use an agile development process to complete the development work because with the large team, the short amount of time, and the unknown technology for the team. | |
| *Dependencies:* N/A | *Successors:* N/A |
| *Scope:* Required | *References:* N/A |

1. NF.2 Source Control

|  |  |
| --- | --- |
| The developers on this team shall store all code in a source control system to ensure that code is available to all developers and versions are available for back-tracking. | |
| *Dependencies:* N/A | *Successors:* N/A |
| *Scope:* Required | *References:* N/A |

1. NF.3 Testing

|  |  |
| --- | --- |
| The developers on this team shall write and update automated unit tests and integration tests for each completed function. This will reduce the cost of regression testing and bug-fixing. | |
| *Dependencies:* N/A | *Successors:* N/A |
| *Scope:* Required | *References:* N/A |

1. NF.4 Code Coverage

|  |  |
| --- | --- |
| Code-coverage tests shall be employed to ensure all code is tested. Testing shall be considered sufficient when it has achieved at least 85% block coverage. | |
| *Dependencies:* N/A | *Successors:* N/A |
| *Scope:* Required | *References:* N/A |

1. NF.5 Operating System

|  |  |
| --- | --- |
| This application will developed for use on Android OS 2.3.3. | |
| *Dependencies:* N/A | *Successors:* N/A |
| *Scope:* Required | *References:* N/A |

1. NF6 Documentation

|  |  |
| --- | --- |
| User documentation will be developed that details operating instructions for the application. | |
| *Dependencies:* Complete application | *Successors:* N/A |
| *Scope:* Required | *References:* N/A |

## Project Deliverable Requirements

1. PD.1 Statement of Work

|  |
| --- |
| The project team shall deliver a Statement of Work that outlines the opportunity, proposed solution, key deliverables, high level product requirements, and the method of team communication. |
| Due: 2/14/2012 |

1. PD.2 Master Project Plan

|  |
| --- |
| The project team shall deliver a Master Project Plan that builds upon the Statement of Work and describes (in detail) project risk, contingency plans, the project schedule, the communication plan, the quality plan, and task responsibilities. |
| Due: 2/28/2012 |

1. PD.3 Status Report

|  |
| --- |
| The project team shall deliver a Status Report that provides an overview of the project and describes its current status. All key changes and issues will be documented and an updated project schedule will be provided. |
| Due: 4/3/2012 |

1. PD.4 Presentation

|  |
| --- |
| The project team shall deliver a 15-minute presentation that outlines the project objective and provides an overview of the deliverable. Project approach, plans, results, and problems will be discussed. |
| Due: 4/24/2012 |

1. PD.5 Project Report

|  |
| --- |
| The project team shall deliver a Project Report that outlines the project and the problem that was solved. Project approach, plans, results, and problems will be discussed. All relevant project management documentation (including an example of team meeting notes) will also be included. |
| Due: 5/1/2012 |

## Project Process Requirements

1. PP.1 Android Market Deployment

|  |  |
| --- | --- |
| The team shall register, configure, upload, and publish the application within the Android Market. The configuration and publishing will be done through the Android Market Developer Console. | |
| *Dependencies:* Complete application | *Successors:* N/A |
| *Scope:* Required | *References:* N/A |

1. PP.2 Contact the “Minnesota Daily”

|  |  |
| --- | --- |
| The team shall contact the Publisher of the *Minnesota Daily* in order to determine whether an article about the application can be published. | |
| *Dependencies:* N/A | *Successors:* N/A |
| *Scope:* Required | *References:* N/A |

1. PP.3 Contact Student Services

|  |  |
| --- | --- |
| The team shall contact the Student Services to add information about the application to the email updates sent out to all university students. | |
| *Dependencies:* Complete application | *Successors:* N/A |
| *Scope:* Required | *References:* N/A |

1. PP.4 Distribute Flyers

|  |  |
| --- | --- |
| The team shall create and print out flyers and attach to bulletin boards around campus. | |
| *Dependencies:* Complete application | *Successors:* N/A |
| *Scope:* Required | *References:* N/A |

1. PP.5 Create Facebook Page

|  |  |
| --- | --- |
| The team shall add a page to Facebook with the application information and send out friend requests. | |
| *Dependencies:* | *Successors:* N/A |
| *Scope:* Required | *References:* N/A |

1. PP.6 Contact Skyway Newspapers

|  |  |
| --- | --- |
| The team shall contact the newspapers that distribute within the skyway system in order to inquire about an article or advertisement for the application. | |
| *Dependencies:* Complete application | *Successors:* N/A |
| *Scope:* Required | *References:* N/A |

1. PP.7 Contact Minneapolis Websites

|  |  |
| --- | --- |
| The team shall contact Minneapolis news websites (e.g. MyFoxTwinCities.com, MNNews.com) to inquire as to whether an article about the application can be published. | |
| *Dependencies:* Complete application | *Successors:* N/A |
| *Scope:* Required | *References:* N/A |

## Limitations and Exclusions

This application will have several limitations, including its operating platform, functionality, hardware, and deployment.  The application will be developed and tested solely for the Android operating system 2.3.3.  No other operating system will be considered for development or test.  Due to the scope of the project and based on available resources and schedule, the functionality will not exceed the defined requirements specified in Section 2.

If modifications are needed, a change request shall be required and documented.  Since the software will be developed for the Android OS, hardware will also be limited to any smart phone that is able to run the Android OS. Since the application requires the use of a positioning system, usage will be limited to phones that have an associated data plan with their phone carrier as well as GPS capability.  Lastly, the deployment of the application will be limited to the Android Marketplace.

# Risks and Contingencies

Risks of the system have been identified and coded into four categories: Technical, External, Organizational, and Project Management. Technical risks involve risks associated with the development, integration, and testing of the application. External risks involve factors which are uncontrollable and out of scope of the project requirements. These risks can be monitored, and risks can be reduced, but the project management team does not control them. Organizational risks involve issues that may arise throughout the scheduling and information sharing processes within the team. These risks are monitored and can be mitigated quickly. Lastly program management risks involve the program management team and the planning of the project in its entirety.

## Risk Analysis

|  |  |
| --- | --- |
| **Risk Id** | **Risk Description** |
| T.1 | **Quality -**The quality of the system is identified as a risk. Within project limitations of schedule and resources, the quality of the system could indirectly be compromised due to oversight.  **Risk Mitigation –** A testing plan will be developed in conjunction with software development. Since we will be adopting an agile/iterative project methodology, continual testing of new functionality as well as regression testing will be conducted throughout the implementation phase to reduce risk  **Risk Trigger –** The risk will be detected and identified if the weekly application JUnit tests achieve a pass rate of less than 80%.  **Risk Contingency Plan –** If the event of the identified risk occurs, the baseline code will be rolled back to the last previous successful version and modifications will be manually merged and tested to identify the cause. |
| T.2 | **Scope and complexity** - The scope and complexity of the system can cause the project to run past its defined schedule. Due to the amount of functionality and limited resources that will be required to implement the application, complexity from the software as well as unforeseen scope creep can contribute to additional risks.  **Risk Mitigation –** Weekly meetings will be conducted to assess current task status. Any blocking issues will be identified and documented. A plan will be instilled for each blocking technical issue and carried out. Project scope will also be assessed on a weekly basis to verify the project will be completed on time with all requirements implemented and tested. Additional analysis on specific tools and APIs will be completed early within the lifecycles to identify any issues.  **Risk Trigger –** The risk will be detected and identified if during a weekly meeting the schedule is projected to be delayed, or if a task has taken 25% longer than estimated.  **Risk Contingency -** If the event of the identified risk occurs, the schedule may be revised to reduce the scope of the project. If the task is close to completion, additional resources will be reallocated to help complete the task. |
| T.3 | **Licensing** – As some developers on the project have licensing conflicts with their employment, it has been identified as a risk. Methods of licensing regarding deployment can cause legality risks for the project management.  **Risk Mitigation -** Discussions with employee’s legal/managerial teams will be conducted to outline limitations of deployment licensing.  **Risk Trigger –** The risk will be detected and identified if notification from team member’s employment has rejected the uses of the project’s chosen form of software licensing.  **Risk Contingency** – If the event of the identified risk occurs, the form of licensing will need to be revised to meet the requirements of the team member’s employment. |
| T.4 | **Limited Domain Knowledge** – Developers within the team have limited knowledge of development on the Android OS and available external APIs.  **Risk Mitigation** - Major technical setup and software tutorials will be established early in the project lifecycle so that we can mitigate the learning curve and build momentum in the development phase. We will establish a strong communication plan to share information regarding technical issues and set up a software repository to quickly share current code.  **Risk Trigger** – The risk will be detected and identified if multiple team members express an inability to complete tasks throughout two consecutive workweeks due to improper training.  **Risk Contingency –** If the event of the identified risk occurs, a mentoring plan will be developed and amended to the communication plan. Key team members with more experience will be identified and assigned as mentors to help struggling team members complete assigned tasks. |
| E.1 | **Reliability –** With the Android OS deployed on many models of cell phones by many companies, the reliability of the phone’s WIFI or GPS geo-location capability is at risk.  **Risk Mitigation -** The application will be tested on several different phones running on a diverse set of Android OS. Testing will be conducted using WIFI enabled Android phones as well as GPS when applicable.  **Risk Trigger –** The risk will be detected and identified if during field-testing the functionality is affected by the Android phone’s hardware.  **Risk Contingency –** If the event of the identified risk occurs, additional analysis will be done to see if there are any software workarounds available. If there is, a change order will be created for additional work on these fixes. If not, the specific phone’s make and model will be documented and added to a list of known hardware limitations. |
| E.2 | **Testing Resources** – The majority of testing will be conducted through simulation. Once testing is conducted in the field, there will be many external factors, which will affect the usability of the application such as environmental, obstructions (buildings and tunnels), and/or software bugs.  **Risk Mitigation** - To mitigate this risk, we will be conducting field-testing as part of our agile development. As we test with our simulations, so will field-testing. This will allow ample time within our schedule to resolve any software fixes that are not identifiable through simulations.  **Risk Trigger –** The risk will be detected and identified if a “blocking” issue arises during field-testing.  **Risk Contingency –** If the event of the identified risk occurs, a requirements change request form will be created to add work-around fixes to the issues or reduce the scope of the application. |
| O.1 | **Team Scheduling** – The project team has grown to a size of eight total. With conflicting schedules, determining meeting time and locations will pose risks.  **Risk Mitigation -** Utilizing our communication plan will enable quick and efficient resolutions. However, if team members are unable to be present at scheduled meetings tools such as Instant Messaging, email, texting and conference calls will be used to mitigate scheduling conflicts.  **Risk Trigger** – The risk will be detected and identified if individual team members are unable to attend three consecutive meetings.  **Risk Contingency** – If the event of the identified risk occurs, meeting times and locations will need to be revised. If needed multiple meetings may need to be created to accommodate team member schedules. |
| O.2 | **Information Sharing** – Each task and work package will be assigned to individuals. Added risk is identified due to multiple areas of independent tasks.  **Risk Mitigation –** We will utilize tools such as Google Docs and Google Sites to be able to share information quickly and efficiently to other members of the team  **Risk Trigger** – The risk will be detected and Identified if there is continual expressed confusion by three or more team members on the uses and location of tools such as Google Docs and Google Sites.  **Risk Contingency –** If the event of the identified risk occurs, a team member will be identified as the information hub. The primary responsibility of this position includes organizing and merging information and documents prepared by all the members of team, and will be the sole owner of the document repository. This will create a manageable and clean document repository where all will be able to download the latest documents and code without the confusing of collision with other team members. |
| PM.1 | **Project Schedule** – With an inexperienced project team, schedule estimates as we as software development estimates will be at risk for inaccuracy.  **Risk Mitigation -** Project schedule and task statuses will be evaluated on a weekly basis to verify the project will be completed on time. Additional experienced resources will be pulled into the process as consultants if issues are not resolved within a pre-determined amount of time.  **Risk Trigger –** The risk will be detected and identified if more than three independent tasks run off schedule within three consecutive weeks.  **Risk Contingency** – If the event of the identified risk occurs, schedule estimates need to be reworked. Resources may be reallocated to other tasks. If project schedule and estimates continues to be inaccurate, change request forms will be created to reduce the scope of the project and its requirements. |

## Risk Assessment

Each risk has been assessed for its likelihood of occurring, the impact it will have on the overall project cost and schedule, and the detection difficulty. The determination criteria are estimated based on previous experience and prior work in related areas. Figure 3.1 shows the results of this analysis. The results are assigned a number from 1 (low) to 5 (high).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk ID** | **Likelihood** | **Impact** | **Detection Difficulty** | **When** |
| T.1 | 2 | 4 | 4 | Test |
| T.2 | 3 | 5 | 3 | Implementation |
| T.3 | 1 | 2 | 5 | Startup |
| T.4 | 5 | 4 | 4 | Startup |
| E.1 | 4 | 3 | 1 | Test |
| E.2 | 3 | 3 | 1 | Test |
| O.1 | 2 | 2 | 1 | Entire Lifecycle |
| O.2 | 3 | 1 | 1 | Entire Lifecycle |
| PM.1 | 5 | 3 | 5 | Startup |

**Table 1: Risk Assessment**

## Risk Severity

Based on the risk assessment in Section 3.2, a visualization of risk severity is shown in Figure 3.2. The risks identified by risk ID T.4 and T.2 have a high impact as well as high likelihood of occurrence. Due to these factors they have been identified as red zone risks and shall be monitored very closely. Risks with medium impact and medium to high likelihood of occurrence have been identified by risk ID PM.1, E.1, and T.1. These risks are within the yellow zone of the matrix and will be monitored periodically but pose no “show stopping” potential. Risks in the white zone, which are O.3, O.1, and T.3, have low impact and low likelihood of occurrence. Due to these factors they will be monitored on a monthly basis. However all mitigation plans will be enforced throughout the project lifecycle such that no contingency plans will be initiated.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | PM.1 | T.4 |  |
|  |  |  |  |  |
| O.3 |  |  | E.1 | T.2 |
|  | O.1 |  | T.1 |  |
|  | T.3 |  |  |  |

Likelihood

**Figure 1: Risk Severity Matrix**

Impact

# Project Schedule

The project will be divided up into five two-week iterations. Table 4.1 outlines the specific dates for each iteration. For detailed information regarding each project schedule task, refer to Appendix A: Project Schedule Breakdown. Each iteration will begin with a kick-off meeting where the team will determine which stories should be included in the iteration, and each supporting task for the story will be assigned to a team member. During this meeting, any new requirements will be estimated as well as any existing requirements that the team feels needs reevaluating. The set of all stories to be completed will be termed the “product backlog.”

|  |  |  |
| --- | --- | --- |
| Iteration | Start Date | End Date |
| 1 | 25-Feb | 9-Mar |
| 2 | 10-Mar | 23-Mar |
| 3 | 24-Mar | 6-Apr |
| 4 | 7-Apr | 20-Apr |
| 5 | 21-Apr | 30-Apr |

**Table 2: Iteration Timeline**

At the end of the iteration, a wrap-up meeting will be held that will allow the team to perform a post-mortem on the iteration and address what they liked, disliked, and what could be done better. This time will also be used to compare actual times and estimated times for each story so that we can improve our estimates and project schedule.

# Communication Plan

Due to the size of this project team and the complexity of the project, timely and consistent communication is imperative to project success. Because the development team is adopting an agile development methodology, the decision has been made to adopt a communication style modeled after the SCRUM methodology. In addition, to facilitate document preparation and ensure that the final deliverables are readable and consistent, one person will be responsible for merging items created within Microsoft Word, Microsoft Project, and Microsoft Excel.  Initial review of all written deliverables will be done by two people, and upon agreement, uploaded into Google Documents for team review. An overview of all meetings and information stores are outlined in the following table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **What Information** | **Audience** | **Frequency** | **Method** | **Responsible Person(s)** |
| Action Items/Deliverables | Team Members | Weekly | Repository | Assigned Team Members |
| Deliverable Issues | Team Members | As Needed | Email/Phone/Meetings | Team Members |
| Iteration Kick-off Meetings | Team Members | Bimonthly | Meetings | Team Members |
| SCRUM Meetings | Team Members | Weekly | Meetings | Team Members |
| Milestones | Team/Professor | Bimonthly | Email/Phone/Meetings | Assigned Team Members |
| Change Requests | Team Members | Anytime | Email/Phone/Meetings | Team Members |
| Project Status Report | Team Members | Weekly | Meetings | Assigned Team Members |

**Table 3: Communication Plan**

## SCRUM

At each project iteration kick-off, a SCRUM master will be appointed from the project team. This individual will be responsible for facilitating SCRUM meetings as well as resolving identified issues that are blocking progress. In the scheduled SCRUM meetings, this individual will be responsible for keeping the meeting on track, ensuring side conversations or elaborations be held after the SCRUM is complete, and logging the progress/blocking issues. To ensure openness, all logs (excluding task/story completions) will be internal to the team and will not be provided in project or product deliverables. The log will contain information on when a story or sub-task begins and when it is completed. This will provide the team a mechanism for evaluating the project schedule and making adjustments as early as possible.

The team will hold weekly in-person SCRUM meetings and at least 1 additional electronic SCRUM with additional ones as determined by the SCRUM master. During these meetings, each team member must quickly address the following three questions:

* What have you done with regards to this project since the last SCRUM meeting?
* What do you plan to do before the next SCRUM?
* What issues are impeding your ability to perform your work as effectively as possible?

This will allow everyone to know what is being worked on so that any issues that arise can be addressed efficiently. During the SCRUM, team members should address the team and not the SCRUM master. The SCRUM master will ensure that the meeting is orderly and that only one person speaks at a given time. If a team member cannot make an in-person SCRUM, the absent member must either attend remotely or make other arrangements with the SCRUM master to provide status in a timely manner – this will typically be provided as an e-mail addressing all of the concerns. Electronic meetings will involve each person sending an e-mail to all other team members addressing each of the SCRUM questions. Submissions will be due by 8pm on the day designated by the SCRUM master. At the end of each SCRUM meeting, the identified risks will be reviewed and new ones can be appended to the list. Triggers will also be reviewed to check if a contingency plan needs to be followed.

## Information Resources

## To provide additional support to team members, internal documentation, such as where to find information on a given topic, will be maintained based on its need. This is currently stored in Google Documents. Given the diversity in backgrounds of the team, additional mentoring may be required. The SCRUM master will facilitate finding a mentor on the team for those who need it and will set up mentoring meetings to fill the knowledge gap.



Source code and related technical info will be stored in a GIT repository. The repository issue tracker will be used to manage the technical aspects of assigned tasks, and the repository wiki will be used for development specific notes.

# Quality Plan

The project will be managed using the Agile development strategy. The project quality plan will be implemented at two levels. Project-based quality will be managed at project management level. Product-based quality will be managed through iterative processes and deliverables.

At the project level, Microsoft Project 2010 will be used to track scheduling, task dependency relationships, and critical path issues. The specific technical and software-related details will be managed at the product level, not the project level. Verification testing will be performed upon completion of the following major feature sets:

1. Basic UI layout and map UI shows current user location
2. Application provides directions between two points
3. Application differentiates between tunnels/skyways and provides UP/DOWN directions for multileveled environments.

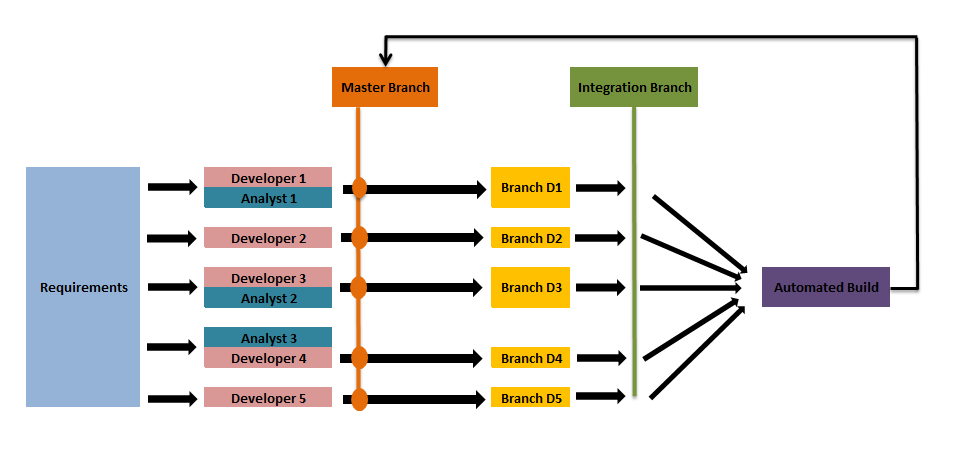
Verification testing will employ manual device testing and automated UI testing. At the product level, quality control measures will be executed with each development iteration. See **Figure**  for a visual guide to the development process.

Figure 2: Iteration Workflow Diagram.

Iterations will focus on incrementally expanding the application’s functionality. Iterations will begin by developers and analysts receiving requirements, which will be accompanied by user stories to help flesh out features to be coded. Developers will branch from the Master Branch to store their own respective code changes. This process will add some complexity to the project, but will decrease risk at merge time. Developers will code user stories adhering to team-established coding standards and will develop unit tests to achieve 85% code coverage. At the end of the iteration, developers will run regression unit tests. If unit tests pass, the developer will locally merge the integration branch on their branch. Assuming no merge conflicts, the developer will push commit changes back to the GIT repository.

Committing to the GIT integration branch will initiate an automated build process via the Travis-CI service. This service will compile the project and run unit tests. Results will be emailed to team members. This automated build will serve as a secondary check to ensure that all committed code is tested and that no code regression has occurred.

By using the SCRUM Agile methodology, we are choosing to adopt several additional quality-control practices. One method we are employing is the iteration post-mortem during the iteration wrap-up meeting. This method will enable us to evaluate current quality and development practices and identify areas of improvement. Also, the use of SCRUM meetings will encourage open communication among team members. This, in turn, will reduce the risk of surprises later in the project schedule.

Non-software materials such as documentation and advertising will be peer-reviewed by one or more people on the team to ensure quality and consistency with the project goals.

# Responsibilities

Due to the large team size, distribution and management of the team responsibilities will be critical to successful project competition. Additionally, special care will be taken to pair developers and analysts together to mitigate separation of the team along developer and non-developer lines. Many of the developer tasks require input and consideration from an analyst’s perspective. Therefore, the team will intentionally look for ways to join developers and analysts to efficiently accomplish tasks. The key project responsibilities are outlined in the following table:

|  |  |
| --- | --- |
| **Responsibility** | **Contact** |
| Prepare Status Report: Coordinate the preparation of an official status report midway through the project schedule. | Kristin |
| Prepare Presentation: Coordinate the preparation and delivery of the project presentation. | Jared |
| Prepare Final Project Report: Coordinate the preparation and delivery of the final project report. | Ian |
| Infrastructure Setup | Weston, Usha, Ian, Sriram, Jared |
| Software Development | Jon, Kristin, Usha, Ian |
| Software Test | Jared, Sriarm, Andrey |
| Application Deployment | Andrey, Weston, Sriram |
| Application Marketing | Jared, Andrey, Sriram |

**Table 4: Key Responsibilities**

**Appendix A: Project Schedule Breakdown**

Attached is the project schedule as rendered in Microsoft Project 2010. Included is the WBS, all milestones, predecessors, duration estimates, and work package responsibilities.