Project Status Report

Mapper for Android, Twin Cities Edition

IE 5541

April 3, 2012

**Authors:**

Andrey Bruslavtsev

Ian De Silva

Jared Swanson

Jonathan Lee

Kristin Dahl

Sriram Rajendran

Usha Kumar

Weston Platter

# Project Overview

As was outlined in the Master Project Plan, our project is focused on developing a navigational application for Android Smartphone since such an application does not currently exist. The purpose of the application is to help Android users navigate the Minneapolis Skyway System as well as the University of Minnesota’s Gopherway. Table 1 lists the major deliverables for this project. All associated deliverables are currently on schedule:

|  |  |  |  |
| --- | --- | --- | --- |
| **Deliverable** | **Work (hours)** | **Start** | **Finish** |
| Android Mapper | 889.5 | Thu 2/2/12 | Tue 5/1/12 |
| Project Management | 724 | Thu 2/2/12 | Tue 5/1/12 |
| Environment Setup | 45 | Mon 2/27/12 | Wed 3/21/12 |
| Application | 72.5 | Mon 3/12/12 | Fri 4/20/12 |
| Deployment | 48 (estimated) | Thu 3/1/12 | Thu 4/26/12 |
| Post Deployment Options | 58 (estimated) | Fri 4/20/12 | Thu 5/17/12 |

**Table 1: Major Deliverables**

The major project deliverables are Project Management, Environment Setup, Application, and Deployment. We also defined a post deployment deliverable which, as evidenced by the finish date, will not be possible to complete by the last day of classes. However, this task will be attempted if the project is ahead of schedule.

Our team is quite diversified as far as academic and professional spectrum; we have team members from electrical, software, industrial, and mechanical engineering fields. Having such an academically diverse team we have set up three major sub-teams: software development team, deployment/marketing team, and project management team. Project Management is handled primarily by our elected project manager Jared Swanson and others help as assigned. The sub-teams have very dispersed boundaries which often cross-over in order to help out the major burden carriers: the software development team. Non-software teams help out with tasks that are part of application development but don’t require code writing skills (e.g. layout mapping, color considerations, and testing).

Because our group has about twice as many team members as the average group in our class (due to the project merge), we emphasize communication a great deal. We communicate through the use of weekly Saturday meetings during which we discuss weekly task accomplishments and issues. In addition, a virtual Wednesday SCRUM meeting is held via email. This meeting is handled by the SCRUM master, and all team members are expected to comment on current their current status and any blocking issues.

# Status

Table 2 provides an overview of the current status of each major project task. The status of each task has been determined by calculating the percent completed. This percentage is determined based upon the number of tasks completed per deliverable. For a complete breakdown of each task, please refer to the attached Work Breakdown Structure.

|  |  |  |
| --- | --- | --- |
| **Task** | **Notes** | **Percent Complete** |
| **Project Management** | On time and under close control | Ongoing |
| Weekly recurring meetings | On track | Ongoing |
| Class deliverables | On track as they become due | Ongoing |
| SCRUM | On track | Ongoing |
|  |  |  |
| **Environmental Setup** | Complete | 100% |
| Create GitHub repository |  | 100% |
| Personal environments |  | 100% |
|  |  |  |
| **Application Development & Test** | On schedule |  |
| Framework |  | 54% |
| Minneapolis Map |  | 100% |
| UMN Twin Cities map |  | 0% |
| Device field testing |  | 33% |
|  |  |  |
| **Deployment & Marketing** | On schedule |  |
| Application readiness for marketplace |  | 0% |
| Application marketing |  | 75% |
|  |  |  |
| **Post Deployment Options** | Optional - app improvement |  |
| Framework |  | 0% |
| Minneapolis map |  | 0% |
| UMN Twin Cities map |  | 0% |

# Key Changes

**Table 2: Major Task Status**

This section describes some key deviations from the Project Master Plan and justifies why these changes are necessary for the success of the project.

## 3.1 Quality Plan

After attempting to writeautomated unit tests for the project, we found that the frameworks needed to perform automated testing (such as a mocking framework, code coverage measurement tool, and the user-interface (UI) testing framework) do not provide the expected functionality. Moreover, we believe that it will cost more time to set-up than it will to perform manual testing and informal code inspections. We, therefore, amended the verification process to the following:

1. Before creating/changing a UI element, an acceptance criteria must be developed from the requirement (this part is the same as before). UI testing will then be performed manually on the application until the tester feels that the UI element sufficiently satisfies the acceptance criteria. Given the fact that structural coverage criterion is insufficient to actually show test adequacy or to show that a requirement is sufficiently satisfied in the application, this seems like a reasonable compromise.
2. When any functionality is added or changed within the system, manual test steps will be created by the person developing the code and both the test cases and the code will be inspected by at least one other developer on the team. All issues will be either fixed immediately or a new bug report will be added to the defect tracking system and tracked from there. The test case will be executed either manually or through an automated test mechanism by a tester (this may be the person doing the inspections or another individual on the team as decided by the scrum master during task assignments for the iteration). Bugs that are found through testing will be reported in the same way as bugs found during inspections.

In addition to the changes in the test process, the automated build tasks were abandoned since there was a significant cost associated with setting this up and the team felt that this time could be used to develop and test key product features. Instead of using a centralized build system, each individual is set up to build the entire project and execute tests within the emulator. At the end of the iteration, a developer creates an internally deployable package that can be used for the product demo and additional testing.

If there is additional time after the core functionality is complete, we may go back and attempt to automate more the testing and use a code coverage measurement tool.

## 3.2 Other Task Changes

A task to integrate EMMA (code coverage measurement) with the builds has been added as an optional task at the end of the project. Please refer to Section 3.1 (Key Changes: Quality Plan) for additional information. Also, a marketing plan deliverable was added to the project so that the marketing team could develop a central strategy and better define how the different marketing venues support the strategy.

Before we changed directions, we added a task to set-up a mocking framework for automated unit testing. This was completed for the Eclipse builds. If we wish to use this with EMMA, it will need to be set up in an ant build script. This will be added as a subtask to the set-up “Automated build” task.

During the development in Iterations 1 and 2, code was written without unit testing. This was prior to the development of the initial quality plan. In order to ensure that this code was properly tested, we divided the unit testing for tasks in those iterations. In addition, this effort was added to the Work Breakdown Structure as a new task.

# Key Issues

# 4.1 Build Environment Test Issues

Testing cannot completely identify all of the bugs within the software and, as a result, this task has been more time consuming than expected. Manual testing should take less time than automated testing to ensure application stability. In addition, black-box testing will be performed using a physical Android device. The purpose of this testing is to uncover any user interface issues that may arise.

## 4.2 Marketing Issues

The newspapers contacted for both the University of Minnesota and the Minneapolis skyway system would like the application to be completed or close to completion before any article will be written. Some of the options that we targeted for advertising simply stated that advertising the application would not happen (Student Services and some of the downtown Minneapolis media).

# Risk Management

Risks have been monitored throughout the duration of the project. Some risks have been mitigated by addressing them directly. For instance, risks associated with software licensing have been mitigated based upon the software license that was chosen and tools that are being used.

The second risk that has been mitigated is the risk of limited domain knowledge. Momentum was gained early in the development cycle which allowed rapid development and a rapid increase in domain knowledge of the Android development environment. All other risks have decreased in likelihood due to the weekly meetings conducted and the frequency with which risks are assessed. Table 3 provides the current assessment of identified risks.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk** | **Likelihood** | **Impact** | **Detection Difficulty** | **When** |
| Quality | 2 | 4 | 4 | Test |
| Scope and Complexity | 2 | 5 | 3 | Implementation |
| Licensing | 1 | 2 | 5 | Mitigated |
| Limited Domain Knowledge | 1 | 4 | 4 | Mitigated |
| Reliability | 3 | 3 | 1 | Test |
| Testing Resources | 3 | 3 | 1 | Test |
| Team Scheduling | 1 | 2 | 1 | Entire Lifecycle |
| Information Sharing | 2 | 1 | 1 | Entire Lifecycle |
| Project Schedule | 3 | 3 | 5 | Startup |

**Table 3: Risk Assessment**

# Project Schedule

An updated copy of the project schedule has been provided as part of this status report. Please see the attached schedule.