

# **Project Postmortem Document**

Team Red (Student Nutrition Mobile App)

Paul Abts  
Gage Askegard  
Donovan Beckmann  
Brett Chastain  
Jordan Falcon  
Connor Fradenburgh  
Zach Grosz  
Tyler Johnson  
Victoria Kyereme  
Grant Moe  
Sethu Monick  
Ryan Nelson  
Mitchell Olson  
Jaron Pollman  
James Raboin

Date: 11 December 2016

## **Project Description:**

This project focused on developing a mobile health tracking app to ensure a healthy lifestyle and choices. The purpose of our project is to create a phone application that can record a user's daily intake of food and then show the user some basic feedback on that user input.

## **Project Success:**

One measure of success is whether all members of the team were involved in the project. In every sprint throughout the project, each team member had work assigned. For the majority of the project, all members were involved, making the project a success.

Another measure of success is whether we gained experience in all major steps of a typical software development project. As a team we did gain experience in Requirements, Design, and Implementation, and Project Management. We didn't gain experience in Testing or Maintenance. The project was a mix of success and failure from this point of view. If viewed from an individual team member's point of view, very few team members got to experience all of the stages we did work on, so the project is more of a failure. It failed to get every team member experience in all major stages of software development.

Another measure of success is whether we were able to complete our project goals, as stated in the Project Specification Document at the beginning of the project. It should be stated that many of these goals were optimistic, given our lack of insight into the process of software development, personal schedules, and inexperience at app development.

- The first two goals were making the system easy to use without training and convenient by simplifying the steps. We haven't really gotten to the point where we could assess whether these goals were reached or not.
- The goal of making relevant and useful recommendations could also not be assessed, as we didn't complete that feature.
- The goal of providing a comprehensive list of all available food items on NDSU campus was over-optimistic, and we weren't able to do this because of the time it took to develop a working system first.
- The goal of using best practices to do dietary analysis might have been achieved. We did use standards from the FDA, but haven't gone through the testing and feedback to know if we were successful.
- The goal of making the system easy to maintain might have been achieved. The Server and App Engine were designed with this in mind. Because of our inexperience with developing an Android app, it is difficult to assess whether the app is easy to maintain or not.
- The goal to complete a prototype and get feedback during the project was missed entirely. We were unable to complete this task, as certain pieces of the project took longer to develop than we had anticipated. As of this writing, we still have not completed a fully-working prototype of all major features.
- The goal of making the database extensible was achieved.

- The goal of keeping a constant velocity and avoiding sleepless nights was achieved. We achieved progress during every sprint throughout the project, and had adequate pace to finish the project without putting in a lot of overtime.

Judging by the ratio of accomplished goals, the project was a failure. We did not achieve very many of the goals we set out to achieve.

Overall, if you ask the team, we would probably say that the project was a success. Our team of inexperienced undergraduate students was given a significant challenge with many pieces, and for the most part, we solved those challenges. We built a server, an app, and an app engine, coordinated work between 15 people for a semester, and gained experience at the process of software development. We also produced much more work than any of our peer teams, which shows our level of commitment to the project.

## **Project Highlights and Accomplishments:**

The project began with a great start because a schedule was laid out early. A schedule put perspective on the semester and set reasonable deadlines for phases of the project. The first phase included a lot of documentation. Our persistence with design documentation wound up giving us a good idea of how we wanted the app to look and behave before implementation. Design documentation was also something to adhere to throughout the project and facilitated our continuous progress throughout the semester. In the end, we now have an opportunity to showcase an application that represents our vision for this project.

The system that the application relies on for shared data was created in a way to maximize uptime and reliability. Documentation was drawn up for usage of the shared data system and helps current and future implementers and maintainers work effectively.

We utilized Git version control throughout the project to manage two repositories. Having impressively knowledgeable team members to work with source control streamlined the process of distributing new code to developers in addition to tracking progress and incremental changes.

## Project Lowlights and Problems:

Here are a number of problems we ran into in the project, as well as some questions and metrics that may help future teams.

| ID        | 001  |
|-----------|--|
| Goal      | Reduce the amount of time and experienced needed to create a high-quality GUI <ul style="list-style-type: none"> <li>- We had a high-quality vision of what the GUI should look like, but fell short of achieving it due to lack of experience and lack of time</li> </ul>   |
| Questions | <ol style="list-style-type: none"> <li>1. How long did it take to develop the current GUI? (How many hours)</li> <li>2. What is the previous experience each programmer had who worked on the GUI?</li> <li>3. How do we measure the quality of a GUI?</li> <li>4. What problems did we run into when trying to build the mockup vision?</li> <li>5. Are there any alternative options/software we could use to reduce the development time of the GUI?</li> </ol>   |
| Metrics   | <ol style="list-style-type: none"> <li>A. Develop standards of GUI quality measurement                             <ol style="list-style-type: none"> <li>a. Survey users to determine preference between multiple GUI options</li> <li>b. Survey users to determine GUI ease of use (without training, with minimal training)</li> </ol> </li> <li>B. Compare time it takes to develop GUI to similar level using different programming languages (Visual Studio, iOS Swift, Android Studio, etc.)</li> </ol> |

| ID        | 002   |
|-----------|---|
| Goal      | Reduce the amount of deadlines missed during development <ul style="list-style-type: none"> <li>- We had issues with taking too long to finish certain parts of the project, which reduced the overall ability of the team to complete a prototype at midterm or to add more features to the project</li> </ul>   |
| Questions | <ol style="list-style-type: none"> <li>1. What prior experience in software development did team members have before this project?</li> <li>2. How much time did each group member spend on the project per sprint, and when was that time spent (beginning, middle, end, throughout)?</li> <li>3. How much time did each group member estimate a particular task would take?</li> <li>4. Why were deadlines missed? (procrastination, poor time management, difficult task, inaccurate estimate of the amount of time the task would take)</li> <li>5. To what degree were deadlines missed? (a few days, a whole sprint, multiple sprints, still not complete)</li> </ol> |
| Metrics   | <ol style="list-style-type: none"> <li>A. Compare different sprint lengths (maybe have more success with one week</li> </ol>  |

Student Nutrition Mobile App

|  |   |
|--|---|
|  | <p>sprints and smaller tasks)</p> <ul style="list-style-type: none"> <li>B. Record initial estimate and actual time tasks took to complete</li> <li>C. Record deadline dates, and by how much they were missed</li> <li>D. Compare requiring small groups to have a leader who subdivides the tasks into more manageable components</li> <li>E. Compare motivation strategies like: peer reviewing work each week, individual presentations of work each week, setting deadlines with the professor to increase seriousness of deadlines</li> </ul> |
|--|---|

| ID        | 003  |
|-----------|--|
| Goal      | <p>Reduce amount of time Server and App GUI teams need to wait after completing their work before solving integration problems with App Engine</p> <ul style="list-style-type: none"> <li>- We waited until after the server and GUI had been well established before working on the App Engine that linked the two, and the overall project is held up until the App Engine is completed correctly</li> <li>- Would have been better to complete a small part of each of the modules and establish a working prototype, so all three teams were at the same place in development</li> </ul> |
| Questions | <ol style="list-style-type: none"> <li>1. How much time have we spent trying to communicate between the various modules (Server, App Engine, App GUI)?</li> <li>2. When were each of the modules started in the project?</li> <li>3. When were each of the Server and GUI modules completed?</li> <li>4. How long have Server and GUI teams been waiting for the App Engine to be completed?</li> <li>5. What percentage of each module was complete before the next began?</li> </ol>   |
| Metrics   | <ul style="list-style-type: none"> <li>A. Compare using different development models like Agile, Waterfall, Prototyping</li> <li>B. Compare approaches like complete prototyping, and starting small and working on all modules simultaneously</li> <li>C. Record amount of time each individual spent on the project per sprint (can see periods of waiting)</li> </ul>   |

Student Nutrition Mobile App

| ID        | 004   |
|-----------|---|
| Goal      | Improve the understanding of Android Studio from the perspective of testers, implementers, and bystanders   |
| Questions | <ol style="list-style-type: none"> <li>1. Survey each team member to understand whether they have used the IDE before</li> <li>2. What are the common mistakes made when unfamiliar with this IDE? (i.e. Pull / Push vs. commit)</li> <li>3. What are the most frequently used features in the IDE? (i.e. debugging, creating a new class, creating a unit test class)</li> </ol> |
| Metrics   | <ol style="list-style-type: none"> <li>A. How much time did it take a user to get comfortable with using the IDE without problems?</li> <li>B. How many of the features in the IDE is the user comfortable with?</li> </ol>   |

| ID        | 005  |
|-----------|--|
| Goal      | Improve the formatting process for documenting from the perspective of the specification and requirement teams   |
| Questions | <ol style="list-style-type: none"> <li>1. Was an example project given as a standard?</li> <li>2. What are some of the major hurdles in formatting these documents?</li> </ol> |
| Metrics   | <ol style="list-style-type: none"> <li>A. How many times did a document have to be reformatted?</li> </ol>   |

| ID        | 006  |
|-----------|--|
| Goal      | Build a communication pipeline between implementers and documentation team to improve team awareness of changes being made to code or documentation, especially improving awareness of the people changes affect.  |
| Questions | <ol style="list-style-type: none"> <li>1. How does the team make changes known?</li> <li>2. What are the most frequently made types of changes?</li> <li>3. Who is the most frequently affected by changes?</li> <li>4. When a change is made, how clear is it what has been changed and who it affects?</li> <li>5. Does the team have a specific individual or role dedicated to keeping the team on the same page (configuration management)?</li> <li>6. Is there a priority system for changes? (i.e. some changes are major, some are very minor, is there a way to tell the difference?)</li> </ol> |
| Metrics   | <ol style="list-style-type: none"> <li>A. How soon after a change is made does the affected member get notified?</li> <li>B. What percentage of project changes does an individual team member know of?</li> </ol>   |

## Ongoing Issues and Risks:

Going forward, the major issues include:

- Working out communication issues between the App Engine and the Server (we weren't able to finish all methods, such as ConstructListOfMenuItems() and ConstructListOfRecommendations())
- Improving the quality of the GUI (requires a higher level understanding of Android than we had)
- Creating a database on the phone to store user profile information to improve app performance (we weren't able to fully implement a SQLite database and built the storage in a way that is probably less efficient than it could be)
- Building a 'permanent' version of the Server add page (currently the page is built inside a team member's personal online repository and won't be editable by future classes), while also adding a 'modify' feature
- Limiting access to who can edit the stored data on the server (anyone can do it right now, and nothing can be modified - only added)
- Needing to work more closely with stakeholders like Dr. Stastny and Dr. Asperin / letting 'real' people use the app (need to fix bugs, add features, be reliable)
- Improving the development teams' understanding of Android Studio as well as general Android development.

Risks include:

- Future team not being able to develop knowledge of Android mobile apps, which may result in the GUI being unusable (Engine and Server still would work)
- Future team not being able to modify and add to the Server, resulting in the need to rebuild the entire server
- Future team wanting to work in a language or platform other than Java, which would result in the need to rebuild all of the code work done so far