



ECOTravel

BY SPOTI5

Project report for the course DAT255, Software engineering project

Examiner: Jan-Philipp Steghöfer

Group members: Emil Axelsson, Matilda Horppu, Johannes Hildén,
Erik Karlkvist, Hampus Rönström

Authors: Matilda Horppu, Johannes Hildén, Hampus Rönström

2015-10-29

Table of Contents

1 Introduction	2
2 Working process	2
2.1 Choosing an idea	2
2.2 Practices used during project	3
2.2.1 User Stories	3
2.2.2 Sprints	3
2.2.3 Meetings	4
2.2.4 Backlog	4
2.3 Collaboration	4
3 Reflection	5
3.1 Non-process specific decisions	5
3.1.1 Firebase	5
3.1.2 Google Maps	5
3.1.3 Size of the Project	6
3.2 Improvements that can be made	6
3.3 Thoughts About Course Administration	6
4 Contribution Report	7

1 Introduction

During this course we have developed an Android application called EcoTravel. EcoTravel is an application that enables companies to compete with each other in who is best at using public transport. The application automatically tracks how much carbon dioxide each user is saving by taking the bus instead of a car. If you are an employee of a company your savings are applied to that company's total savings. Thus making it possible for companies to compete in who are the best at riding public transportation to work.

During the process of developing this application we have tried new ways of working and received new knowledge of how to work with a project in a small group. In this report we will describe and discuss our working process and how it could be improved for future projects.

2 Working process

In this section our working process will be discussed. It will be described how we decided what kind of project we wanted to do and how we planned and structured our work.

2.1 Choosing an idea

The first step in our development process was to decide what kind of application we wanted to build. Since this was an entry for the ElectriCity Innovation Challenge there were a few guidelines we had to consider. The most important ones were to decide on which category we wanted to compete in and that our application had to have a clear connection to the ElectriCity busses. This was not a problem since the whole group had the most interest in the category "Efficient and Fun" and the connection with the busses came naturally with our ideas.

When we had decided on a category we started brainstorming ideas. During the brainstorming we tried not to think about how we would implement the ideas but instead focusing on producing as many good ideas as possible. Our brainstorming resulted in three good ideas which we continued to work on. The first idea was to make a small game that could only be played on the bus where the player collected as many points as possible between bus stops. The second idea was to make a chat application for the buses so that people could anonymously chat with each other while riding the bus. Finally the last idea was to track how much each user of the application rides the bus and then calculate how much carbon dioxide they save by doing so.

Our discussions about what idea to choose and develop mainly revolved around if we wanted to make the bus trip more fun for those who already rode the bus on a daily basis or if we wanted to attract new people to public transportation. We felt and still feel like it is more important to attract people who take the car instead of the bus rather than making it more fun for those who would choose the bus either way. The group of people that, according to

statistics¹, used public transportation the least were 40-49 year olds. We felt like this age group consisted of people in the middle of their careers with enough money to keep a car therefore needing new reasons to ride public transportation. To attract this target audience we figured they needed a push from their employers and we therefore tried making it more appealing for companies to make their employees ride public transport to work.

2.2 Practices used during project

To make the development of the application more efficient we decided to use the Scrum method. As this method of working was completely new to all of the members of our group it took a while to get used to. Scrum seems like a very good way of working on a software development project in a small group. It helped us with the structure in terms of workflow and how to split parts of the development so that we all could develop different parts of the application at the same time. However we do not think we managed to utilize Scrum to its full potential as it seems to require a lot of practice to do so. All parts of Scrum we used during the project will be discussed below.

2.2.1 User Stories

We started off creating user stories together and made a story for every feature that we thought would be relevant to get a complete application. This was not too difficult and we think we managed to create clear and solid user stories. Although at times it seemed hard to tell if the target of the user stories were us as developers or the actual user.

After creating the user stories we needed to prioritize them and this is where, for us, Scrum got tricky. We used two different methods to prioritize them. In the very first sprint we tried using planning poker as this is what we did in the lego-Scrum exercise. However we thought it took a bit too long for us to decide on what numbers to pick and decided we were too inexperienced thus making it inefficient. In the following sprints we instead discussed every remaining user story until we got to an agreed number. Being a group of five coherent people this worked out pretty well. However with more experience of estimating task size and difficulty, planning poker would most likely have been the more efficient option.

The hardest part about using Scrum for us was breaking down the user stories into smaller tasks. We did not separate the tasks well enough and did in some cases not give them a clear enough definition of done. This resulted in us constantly having a lot of unfinished tasks in our backlog which made it look a lot more chaotic than it needed to. The biggest advantage of using user stories is that you really get to think through what you want to include in the project and is able to grasp how large it will be. Estimating the size of the project is something that becomes easier with more practice though.

2.2.2 Sprints

We worked with sprint lengths of a week and think this was suitable for a project of this length and size. We chose one week because we felt like it would be difficult to finish a

¹ Västtrafiks varumärkesmätning 2014 (n:1165)

whole user story in less time and if we would have taken a longer sprint we would only been able to have very few during the course of the project. In between every sprint we had a meeting where we both discussed the finished sprint and planned the upcoming sprint. Splitting this up into two meetings might have been a good idea considering how we often went straight to the planning of the next sprint, instead of reviewing the last one, in order to quickly go back to writing code.

2.2.3 Meetings

We did not use the practice of a daily standup meeting often used within Scrum. We found it hard to match our schedules to plan meetings this often. Instead we had about two meetings per week where we discussed issues related to the development of the application. In addition to this we used a text based chat to communicate on a daily basis. After doing this it is clear that it is a lot harder to describe an issue in text than it is in person. You often have to spend more time explaining what your problem is than the time needed to come up with a solution. However compared to not having any daily communication at all, it is alright. If we were to have a project over a longer period of time with longer sprints it would probably be necessary to have daily stand up meetings to not get out of sync.

2.2.4 Backlog

To help us keep track of what our backlog looked like we used the web site Trello. It worked very well and gave a great visual representation of what was in progress and who was assigned to what task. The problem with Trello is that every person in the group must continuously update what they are working on and what status that task currently has. This part could sometimes have been done better since people occasionally forgot to assign themselves to a task or update its status. Trello is an easy tool to use so you gain a lot from using it in relation to the time it takes to learn and use. Using this tool is probably beneficial to all projects where you work in small groups.

2.3 Collaboration

We have been very lucky with how well we have been able to collaborate during this project. Some of us have been working together previously but even adding new people to the group has proven to work very well. We have had good communication throughout the project which has resulted in good discussions during meetings and that it has been relatively easy to make decisions. We feel that everyone have been able to express their opinions during discussions. When someone has had something they wanted to talk about it has been met with respect and has been discussed within the group until a democratic agreement has been made. This is very important when making decisions in a group. If everyone is able to share their opinion, aspects that other team members have foreseen may be uncovered and may change the outcome of the discussion. It is also important that the whole group feel included in the decision making process.

All group members have attended almost every meeting. When someone has not been able to attend there has been a valid reason. This has resulted in there being a good spirit within

the group. Not attending meeting is otherwise a thing that can create hard feelings. Another thing that could result in hard feelings is if there is a member that does not contribute to the work as much as the others. We have not had this problem either. Even though some persons have produced more work than others all group members have been equally engaged in the project and equally interested in making the application as good as possible.

3 Reflection

As in every project or group assignment there are things that could be improved in order to achieve more efficient work. In this section we will reflect on some decisions we made that are not directly connected to the working process but also how we could improve our working process for future projects. We will also give some opinions regarding the course administration and the ElectriCity-competition.

3.1 Non-process specific decisions

During the project we had a few problems that had to be solved, for example storing data and measuring distances. We were forced to decide on how to solve these problems which will be discussed below, as well as the size of the project.

3.1.1 Firebase

We realized early on in the development process that our application would require a database since there is a lot of data being produced. There are many options to what kind of database to use. Since the group had very little experience with databases in general we were looking for something simple. At first we were looking at MySQL but decided that it would be too complicated to always have a server running somewhere. We then found Firebase which provided a server online therefore making it easy to get started. Because of this we decided to use Firebase. Even though it required a lot of time to get Firebase to work the way we wanted to it was probably the best option since it was fairly easy once we understood how it worked.

3.1.2 Google Maps

For calculating the driving distance between two locations we initially chose to work with the GraphHopper Directions API, since it seemed to be simple as well as fitting for our needs. However using GraphHopper turned out to be more troublesome than we first thought. After trying to make it work for way too long we abandoned GraphHopper and instead turned to the Google Maps Directions API. This API worked out for us almost instantly.

At first when GraphHopper was not working for us we assumed it was because we were doing something wrong, or because we were inexperienced in working with API requests and JSON. After the switch to Google Maps however and how it started working pretty much instantly that was probably not the case. The time spent on trying to work with GraphHopper felt wasted and we should probably have started looking for an alternative earlier than we

did. The fact that we did not delayed parts of the project because some of our team members were tied up in trying to solve the problem.

3.1.3 Size of the Project

We knew that we had a limited amount of time to finish the project. Unfortunately we did not consider this enough when deciding on how to implement our ideas. We simply made the scope of our application a bit to big. When we were planning our project and estimating the amount of time all of our features would take to implement we probably underestimated a few of them. Once again because of inexperience with breaking down user-stories into smaller tasks to really see what was required of every user-story. There were also a few things like the database that we in our early stages of planning did not realize that we needed to use.

3.2 Improvements that can be made

If we were to redo the same project again there are a few things we might have done differently, both regarding decisions we have made and about the working process.

Even though we implemented all of the features we wanted to implement, some of them are not as polished as they could have been due to us having too big of a scope. With a smaller scope we could also have had more time to improve the visual side of things and make the application more user-friendly. Another solution we could have used to improve the user-friendliness would have been to have one person be in charge of the general look of the application, to give it a more complete look. This ended up happening in the end anyway as we had to make the different views look more like each other but having that agreed from the start would have been more efficient.

Something else that would have been helpful to have right from the start is some kind of UML diagram over the general project structure. It would have greatly reduced the amount of time spent on refactoring code over the course of the project. Also, it would have been easier to understand how things are connected.

We would have loved to be working more closely together more often. First of all to improve the communication within the group but also because it is more enjoyable. Being together more often would also have enabled us to take greater part in each other's work therefore reducing the truck factor. We knew that seeing each other a lot would be beneficial but our different personal schedules kept us from doing so more than we did.

3.3 Thoughts About Course Administration

We think it was a good idea to combine the course with the ElectriCity Innovation Challenge. It gave us an insight in not only what kind of Android applications that are requested in the public transport sector but also what kind of innovations in general. We also appreciated the opportunity to speak with large companies like Volvo and Västtrafik to get input on our thoughts and ideas. They were very helpful and came with suggestions on how we could

solve problems and further evolve our application. One thing that could improve regarding the collaboration of the contest and the course is that the contest started several weeks after the course started. This made us feel like we were wasting a few weeks time since we were done with our concept way before we got access to the API's provided by the contest.

The contest provided several interesting workshops and we tried to attend as many as possible. The workshops in general seemed very well prepared and it seemed like they put a lot of work and thought into them. On account of this we got a lot of useful information from almost all of the ones we attended. The workshop "24h with the bus" was very interesting and provided a lot of useful information, for example statistics of people who travel with Västtrafik which supported our prediction that 40-50 year olds do not take the bus very often. It also provided a chance to talk to stakeholders and thus we got assurance that our idea was interesting. The final workshop about selling a concept was also very interesting. The fact that you got pitch for multiple people and then them giving you feedback from it was great. It also gave us ideas of how to make our final product more unique. The one workshop that did not provide any useful information for us was the one about the bus stops, since we did not focus on the bus stops in our project at all.

All group members were not able to attend the workshops so we needed to select who would be given the opportunity to go to them. We decided this based on who was interested in going to a specific workshop and who we thought could make the best use of the information. For us it was not a problem that only a few members of each team were able to attend the workshops since all of the useful information was shared upon the following meeting.

We have discussed the fact that there were no supervisors in this course and got to the conclusion that there are both pros and cons to that. If we would have had supervisors some parts of the work would have gone a lot easier, for example how to structure an Android project, which we did not have any previous experience of. On the other hand, now that we did not have any supervisors we have had to figure these things out ourselves which probably gave us a deeper understanding of how things work.

One thing that we really appreciated was the Lego-scrum exercise. It was a very good way of pointing out common problems when using scrum and it also provided some training in how to think when splitting up user stories into smaller tasks and prioritize them. Without that exercise we would never have have gotten the knowledge needed to use Scrum to our advantage that quickly.

4 Contribution Report

During the planning phase, i.e. the first weeks before the contest started, every member spent an equal amount of time since all the planning work took place during meetings. We had approximately three meetings a week, 1,5 h each. We also attended lectures and did some setting up of the project for approximately 4 hours each week. This sums up to about

17 h per person. On top of this each person also spent some time learning the basics of Android programming.

During the coding phase the time devoted to project work deviates a bit from one another. Matilda spent approximately 15 h per week, i.e. 70 hours in total. Emil spent approximately 20 h per week, maybe 25 h in the last two weeks, 110 h in total. Erik has approximately 20 h per week with a bit more the last few weeks and as much as 30 hours the last week before the finals, in total about 100-110 hours. Johannes spent somewhere between 13-18 h per week, coming closer to 13 hours in the early weeks and 18 hours in the final weeks, 80 h in total. Hampus spent about an average of 18 hours per week on the project, but it was very different from week to week, with the low being around 10 hours in the early weeks and the high being around 35 hours in the week before the challenge finals. Estimated total time of 90 hours spent on this phase.

In the last phase, the wrap up, when focusing on writing documentation as well as tidying up the code, we tried to split up the work so that everyone got a task they thought they could manage during that week. Matilda spent approximately 16 hours. Johannes spent approximately 18 hours. Hampus spent about 22 hours. Erik spent around 18 hours. Emil spent about 28 hours the last week,

In the project as a whole we have spent the following amount of time per person:

- Matilda: ~106 hours
- Emil: ~155 hours
- Erik: ~150 hours
- Johannes: ~115 hours
- Hampus: ~129 hours