Amanda Belfrage, Pedram Talebi, Sebastian Lund
Jesper Carlsson, Martin Sigvardsson, Måns Lundberg

Moosecrunchers Reflection Report

Application of Scrum

Roles, teamwork and social contract

Under the first meeting we decided how we should work and who would be the scrum master. We all felt that Måns had the most experience with leading projects so he naturally became the scrum master to keep the others on track. We then divided the rest of the group in two teams, one handling the frontend and one handling the backend. The group responsible for frontend was Amanda, Pedrami and Martin. Amanda and Martin later focused more on the documentation along with Måns. The group responsible for backend was Sebastian and Jesper. We also did a social contract to make sure that everyone knew what was expected of them. The social contract contained, among others, agreements on how to contact each other, how to make decisions and to show respect for each other. To see the whole contract, look in the git repository. We did manage to fulfill the contract under the whole project. We all got along very well and no one felt unappreciated.

Used practices

The different practices that we used during the project include:

- Pair programming to spread the knowledge between the different team members, making us able to continue working even if someone would be absent.
- A burndown chart to easily keep track of where we are at in a completion perspective.
- An easily overlookable scrum task board using Trello.
- Stand-up meetings to know what everyone else was doing.

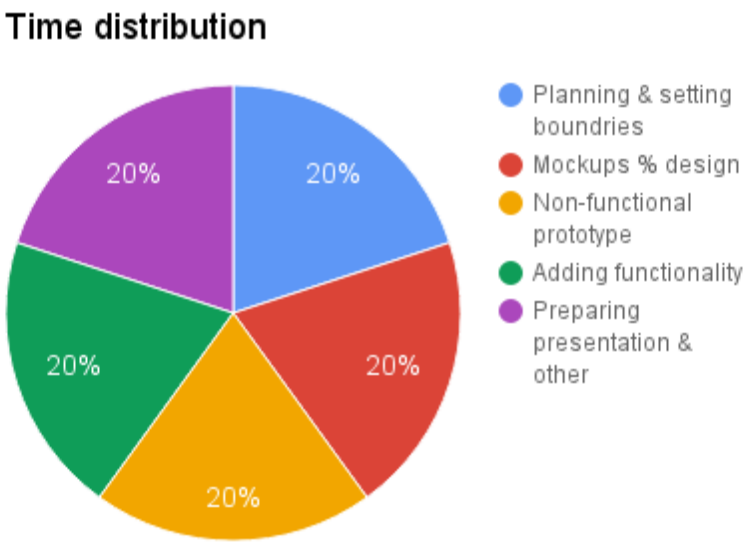
These practices worked well during the project and we would not do anything differently. It was good to use pair programming, since some of our team members already were familiarized with the technology we used. By pairing them with member that weren't as comfortable with the technology, really helped the members to work together and we learned a lot. The burndown chart was a good way to make sure we were on track and the scrum task board helped us getting there. The stand-up meetings were also helpful, since it gave a good opportunity to ask for help and to see what have already been done.

Time distribution

Name	**Responsibilities - Contributions**
Måns	Scrum Master, reports

Pedram	Frontend, design
Sebastian	Backend, App
Jesper	Backend and database design
Martin	App, frontend design, reports
Amanda	Frontend, reports

We started with setting the boundaries for the project and formulated a vision. It was important to set a definition of done and that everyone were agreed. We returned and changed the vision a couple of times when our conditions changed. Planning was around 20 % of the project. Then we created mockups to start with the design. We kept the design through the project and the design work was around 20 % of our project. Creating a non functional prototype was 20 % and then adding all functionality was also 20 %. We simultaneously continuously tested the functions. In the end we found some smaller things to add and change. That together with preparing the presentation was the last 20 %.



We choose to have many but small sprints, so each sprint were one week and in total we had five sprints. This was a good way to work, since it really made the goal reachable. However, we had some problems deciding velocity and effort. It was hard to estimate those, which in the beginning lead to that some members had tons of work while others had none. To solve this we decided that once a member was done with his/hers tasks and still were in a sprint, that person were free to continue on a task that we not yet had distributed. Once we had done this change, we got back the flow in the project. What was done in each sprint can be found in the git repository.

Effort and Velocity

The velocity of the project has been very linear to the effort we've put in. We had some weeks where everyone were very busy with the bachelor thesis and the Cortège-week. Those weeks we set low goals and prioritized the work for those who had more time to put into the project. Our supervisor was a bit worried about us finishing the project but we were confident and afterwards we can establish that the project was still a success.

User stories and tasks

After deciding what project we wanted to do, we created user stories. Since our project were both for the traffic leaders and the bus drivers, we did two different user stories, see git repository. Looking at the driver's perspective we almost fulfilled the user story. The only thing that we didn't fulfill was to show information only if the information were relevant. It was more complicated than we thought and we chose to prioritise the basic functionality instead. However, we did manage to put in support for it, so it should be relatively simple to implement it if the project continues.

Looking at the traffic leader's perspective we did fulfill the whole user story. We did manage to create a simple and intuitive website where the traffic leaders easily can send a new route to the bus drivers. We also have removed the language barrier, only some basic swedish is required for the traffic leaders. However, since they do not have to speak with the drivers the main language barrier have been overcome.

To fulfill the user stories we broke them down into smaller tasks, to see them all look in the repository. These tasks were then distributed amongst the group members from week to week.

Reflection on the sprint retrospectives

At every Wednesday meeting we went through these main retrospective questions.

1. What went well during the sprint cycle?
2. What went wrong during the sprint cycle?
3. What could we do differently to improve?

The first two points were usually issue free, everything went well and very few things went wrong, however we quite often ended up at a point where we had to mix up the different coding teams to be able to continue as efficient as possible using the shared knowledge from the frontend/backend development teams.

Reflection on the sprint reviews

During the development of our product we were in contact with the stakeholders two times. The first time we pitched our idea to get feedback and to see if it was a good idea. Turns out the stakeholders liked it and also gave us some features that they wanted to see in the product. We feel that it was a good choice to contact them in the beginning since they gave great response on what they wanted. Without speaking to them we would have missed some basic information, like that they didn't want the bus drivers to be able to contact the traffic leaders through the application. The next time we met them were when we had developed a prototype. This time we let a traffic leader try out the website and then give us feedback. Over all he liked the website and the application. He thought that it was easy to use and had all the basic features. However, he did give us some great feedback on other things that would help the traffic leaders even more. Some examples are some kind of favourite reroutes so they do not have to remake a reroute that often appears, when they click on a road a list of all the lines that lies on that road should appear and they want to be able to show which bus stop the bus driver should stop at. These are all great ideas and it is doable. Unfortunately they are a little too complex to implement during this project, but if the project continues they would definitely be implemented.

We feel that we have had a good interaction with the stakeholders. They have given us a clear picture of what they want and need, then we have chosen the most basic functions to create a soild ground. We have succeeded to develop a website and application that they are satisfied with, so the communication have been good during the project. The only thing that would have been even better is that it would have been great to meet the traffic leaders earlier, so that we actually would have had time to implement some of their ideas. That is something we will think of in the future.

Best practices for new tools and technologies

Considering the limited time frame it was important at an early stage to reduce the development time as much as possible and find the best suited technology tools which facilitates our development. The project group chose to work with a Fullstack-Javascript for the web application; a backend with Node.JS, database with MongoDB and a frontend utilizing the Angular framework. These technologies were familiar by part of the group which made it easy to use them, and also mainly the reason why we chose them.

Further to be able to communicate between each "layer", third party libraries were used, mainly due to not reinvent the wheel. The frontend Angular were using the "$http" service to communicate through a restful-API with the backend. The backend where using a web framework called Express for Node.JS and a library called Mongoose to be able to communicate with the MongoDB database.

In conclusion, we are satisfied with our selected technologies we chose to work with, mainly for the ease of creating web applications with the powerful MEAN-stack (MongoDB, ExpressJS, AngularJS, NodeJS) technologies.

Regarding the use of APIs, Google Maps was the only third party API that were used in this project. Google Maps were chosen due to its large extent of documentation and large number tutorials explaining the API. The Google Maps API was a great help and easy to use and now, in hindsight, we are satisfied with the choice of the Maps API.

The application for the bus drivers is developed in the programming language Java and is only runnable on phones and tablets with the Android mobile operative system. The group decided at an early stage that functionalities for the application should be limited, and instead to focus for a functioning prototype which presents our concept and idea. It appeared that the group made the right decision regarding limiting the functionalities, since there was no time left for development after the completion of the prototype.

Reflection on the relationship between *prototype*, *process* and *stakeholder value*

The prototype was very important, because we were able to effectively show our idea to the stakeholders. For this reason we placed emphasis on starting to work on the frontend immediately, to ensure that we wouldn't end up in a situation where we had a functioning backend that we were unable to display.

Overall the process was very smooth, and we were able to conduct acceptance tests with Keolis before the final presentation of our prototype, which was very valuable. We received a lot of feedback - most of them ideas on how to improve our app (or rather, what the traffic leader would like to have if he were to use it). Some suggestions were recurring - for example, we were often asked why we didn't have GPS-support, which was something we had set as a secondary feature. This could suggest that perhaps we should have planned to support it baseline.

The potential of our application is huge - our idea is not limited by neither geography nor language. This was something the traffic leader we spoke to also realised, and he urged us to keep working on the project. One thing we didn't consider is that while our app is primarily intended to aid in redirecting traffic, it could also help in everyday driving - today the drivers are expected to learn and remember the routes , and this can cause problems when a driver goes from driving one line to a different one. It can also be used as an educational tool for new drivers.

Relation to literature or guest lectures

The main thing that we understood from the first weeks lectures was that it is all about the project managing. Things will not go your way and you will end up with a bad product if you

do not plan beforehand. When we finally decided what task we wanted to solve and agreed about an idea and vision for a product we thought about what had been said in the lectures and tried to make the project according to the Open/closed principle. We agreed about the essential functionality for the product to work and then a lot of extra functionality which we could add but wasn't necessary.

Project management consist of a trade off between scope, time and cost and we believe that we during our planning had good understanding of these trade offs. Especially regarding the scope. First we wanted the app to show how much delayed the bus will be with the new reroute but very fast we realised it would not be possible within our time and cost span.
We believe our project have followed the literature way of creating software in a good way but if it really was caused by the literature or if there is no correlation we can't determine.

The design lecture gave us very interesting perspectives on the daily apps that we use but it didn't matter to our app because of the extremely simple functionality we use. Most of our design work have been done to the web frontend which the lecture or literature haven't concerned. We can't say that we can base our web frontend design on some real facts or literature more than our own experience. But that is a decision we've taken and we don't think it will bother anyone.

The testing lecture gave us good tips of how to confident about our testing. We already had the idea of trying to get a real bus driver at their stop at Sven Hultins Gata to try our app and just hear if there was some major misunderstandings. We validated the idea with the traffic planner at Keolis and he said that they had had the idea earlier but they have never tried to build the software. The difference between static and dynamic testing was enlightening and we changed the way we did our testing to be sure to cover a greater span of static and dynamic testing.

Comments to D1 and D2

D1 was the first submission we had. We reflected on what we learned in the lego session. The main points were that we started building without a plan. That resulted in many building the same part and everyone had a different vision about how the house would look when finished. Second realization is that our plans were to big. That made the project way to complicated. It's better to build something small with possibilities to add features. The last big learning from the lego session was to talk to the product owner. Several times during the process you should make sure with the product owner that you're heading in the right direction.

Our vision was to eliminate the language barriers between the bus driver and the traffic leader. It was necessary that the bus driver only received visual information. We was able to hold on to the vision through the project. We neglected some of our initial extra features but that didn't affect our vision.

When we had the vision we created an initial product backlog. It was pretty basic because we had to rely on third party APIs which we at that point didn't knew how well it would

cooperate with us. Afterwards we can with ease say that we shouldn't had worried. The APIs were easy to use and the biggest challenges was to create the frontends and build the whole backend. But that was also the fun with the project.

At the same meeting when we created the vision and the backlog we also wrote a social contract within the group. It contained our shared ambition, how we were going to have contact with each other (slack was selected), how decisions were going to be made and most important of all that everyone will "respect each other's opinions and won't dismiss ideas without consideration".

After two and a half sprint we wrote a halftime report regarding our work so far. It was a good way to reflect on how the group was working so far. But nevertheless it would've been better if that work would've been done a week later because we had just had a sprint with not so much work planned to do nor done. As we saw indications of already at that time many in the group had a lot to do with the other responsibilities in the Student Union. Those responsibilities together with the bachelor thesis didn't leave much time over to work on this project and even harder to meet in person and discuss solutions. It was great that we still managed to meet every Wednesday. Here we also saw the importance of respect and understanding for each other. Those who had a little less to do was able to take greater responsibilities when our resources were reduced.