CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

DAT255/DIT543 Software Engineering Project

Grupp Tesla

Post–mortem Report

Authors: Jonas Arvidsson, Kim Berger, Patrick Franz, Sebastian Kvarnström, Michael Tran

BusExplorer

Table of contents

Processes and practices

In our project we quickly adopted Scrum and separated our sprints into one week periods. This left us with five sprints including one final week for testing and documentation.
Since we were not able to directly communicate with a product owner, we decided that each group member would get a chance to be both the Scrum master and the product owner. However, this structure was admittedly hard to maintain and it ended up being easier if all of us acted as the product owner. Of course, as it is a vital part of Scrum, we often held daily Scrum meetings. This helped us keep track of what everyone was doing.
Our user stories and tasks were separated using Trello. This made it very easy to prioritize the tasks at hand and also to sort them appropriately, as well as mark them as done by a simple drag and drop.

Time spent

When dividing the work and the different tasks among the group we started by asking ourselves who has the knowledge to complete this task and that person would then work on that particular task. This made it so not everyone worked the same amount of hours, but if someone completed a task they could offer help to another group member which meant that the group member could accept and receive help if he needed it.
With the least programming knowledge Jonas took on the simpler tasks when writing the app and tried to make up for it by taking on more organizational work such as documentation. For more difficult tasks we used pair programming as a tool to complete the tasks. Time spent on the project each week since we started programming was approximately 15-20 hours including time spent learning Android, lectures, the LEGO exercise and going to ElectriCity events.
Kim had an average of 16 hour/week, which is from approximation of calculating the time spent for the course which consists of coding, problem-solving and what was mentioned above. And time-wise, Kim had a slow start, mainly because we were waiting for the API and mostly discussing exactly what we should do. Due to sickness over a week when we were halfway through the course, the time spent was slightly decreased but made up for by adding another 8 hours on during the weekends. The last three weeks Kim worked about 10 hours per week on top of my weekend-work. Kim had to spend a lot of extra time because he didn't have an Android phone, and also never got some of the external utilities to work, like the Google Maps or our database.
Patrick spent on average 15-20 hours per week on the project (incl. attending all events). How much exactly varied a bit and it often depended on the amount of research and testing of new features that had to be done. Patrick worked roughly 12-15 hours a week on our app and in addition to that spent up to 10 hours a week researching and testing new possible features for our app.
Sebastian spent roughly 15-20 hours per week on the project (with some additional bursts of productivity some evenings). Around 60% of his time was spent adding new features with 30% of the time spent testing and debugging. The remaining 10% were also spent refactoring the project and taking photographs for our app.
Michael spent about 15-20 hours a week on average, working on this project. That includes everything course related (lectures, workshops, meetings, EIC events, etc.). Most of the time spent was researching, implementing futures and debugging/testing the application. Also planning (mostly at the beginning), user testing and documenting (later on) was a big part.
For the project as a whole the team would meet about 3 times a week with meetings lasting from 1 hour to 4 hours. This time was spent programming together, deciding on ideas for the app and writing documentation.

Thoughts on Scrum

We really like the small sprints (we used weekly scrums) to get smaller chunk of code (user stories) into working order right away.
The daily Scrum meetings were also very helpful, because it made the project extremely adjustable so we all had an steady and sustainable pace. If you got stuck, you could by the next Daily Scrum, just ask for assistance or even trash a feature because it would take too much time to implement or if the result would look really bad.
If you happen to be done with your part of the sprint and no one is in need of help, you could just pick a new feature that we've added to our priority-queue, which we had on Trello, during our meetings, that no one else is working on.
Another advantage we found was not having our project plan "written in stone", meaning you have more freedom to modify, add or remove a user-story.

Disadvantages

Our biggest problem was it took some time to adjust to all the techniques Scrum uses, because there were so many.
Another problem were that we only had some rough ideas for design-patterns to follow in the beginning, but after adjusting the code and adding more features, we lost the design as the time went, and had to do some code refactoring to have a tidy code in the end.

- *How efficient was the technique given the time it took to use?*
  We found that the techniques were really efficient, as soon as we got the hang of how we should use them properly.
- *In which situations would you use this technique in a future project?*
  For larger projects, with a minimum of two months work time, even if the group only exist of one member.
- *In which situations would you not use this technique in a future project?*
  For smaller projects where no big structure/plan is needed.
- *If you had the practice/technique in a part of the project and not the entire project, how was using it compared to not using it?*

It would be way to stressful to implement all or even most of the techniques properly if we only had to use them for an even shorter time.

Our work process - the good

With specific goals set we were able to get started quite quickly and it was harder to leave things for next week since we had fairly static deadlines. Fortunately we all we planned our meetings and talked about other project related things using Slack. Slack is available both as an app and a website so we could communicate at all times, which was helpful when someone had a problem they wanted to resolve quickly and someone would respond quickly after the problem was posted to Slack.

Taking responsibilities worked in a very natural way. We often only needed one group member to take on a task or responsibility and the rest pretty much followed automatically. There were hardly any long discussions about, who is going to do what.

We all work very well together, and could easily fell to jokes and laughter, but we were also well focused and completed the project without ever having a dispute, argument or even an issue.

Other than what the others mentioned, a vital part was that we had our daily scrums and sprint-end meetings. We could discuss and solve problems pretty quickly since we had these kind of meetings several times a week. We used Slack and Trello to communicate and setup/assign tasks even when we weren't in the presence of each other, and did so frequently.

Our work process - the bad

We got the project started a bit too late. We felt like we couldn't do much without the API so we didn't manage to get much done before the API was released. In truth, we could've started the project earlier than we did. However, we also had a lot of issues getting the API to work. We were getting the same system-id values for all the buses when they should have been unique, and often times various other things would stop working randomly, especially the GPS.
Kim didn't have an android phone, and also had some issues with the emulator and couldn't get services such as google-maps or the database to work. Kim had to build his own version of the project to be able to test certain aspects. Even if it worked flawlessly on Kim's version there were usually some errors for some the members with an android phone when Kim's code was submitted into the repository. This meant Kim was always dependent on others to help fix his code so it would work with an android phone.
While we hardly had any technical difficulties using Github, we maintained too many branches and didn't "clean them up" properly. It was hard to follow, which branches were active and which ones weren't. We could've also committed/merged some code earlier to the master branch, which led to that we had to "assume" that certain methods were working, not to mention a couple of merge conflicts.

Non-process specific decisions

The LEGO training was very useful in teaching us a lot about working with Scrum. Unfortunately, there were of course parts of Scrum that we were not able to apply well in this course, such as having a product owner or working with other groups.
We learned a lot about working with other tools as well. Initially, we were planning on storing all our data on an internal database, but later decided to mix things up by using an internal SQL database as well as an external Parse database.
The idea behind this was that we could easily change the database without updating the app, and by storing data internally and externally lets the user stay up to date without having to download a new database every time they start the app.

Working as a team

Sometimes it was a bit hard to maintain to focus on a topic to finish a discussion without drifting into another topic, but overall we kept a certain focus throughout the project that resulted in a fairly trouble free development process.
We got our communication tools started very quickly and could easily start to organize meetings. At every meeting we talked about what everyone is currently working on and someone needed help with anything. For example getting started with github was a challenge for some and those in the group with knowledge of github could offer their help and everyone could get connected and start working fast and without too much trouble.
The online tools we used were very helpful for us to organize our project. Using Trello we wrote everything from user stories, tasks that needed to be done, tasks that were done, and pictures we had taken from our meetings. This worked well to let everyone know what everyone on the team was working on, and if someone had time over they could easily choose a task to start working on.

For future projects

The big issue was the overwhelming amount of knowledge you had to gather from scratch to properly know how to use all techniques correct. so we didn’t grasp all the concepts some techniques were after. But other then that, I would say try to avoid what we mentioned above, that is, trying to uphold the design patterns we wanted to use.

Innovation Challenge workshops

At the start of the project each member of the team decided which ElectriCity workshop they could to attend so that the time spent on the events would be divided as fairly as possible.
The first workshop we attended was the Concept generation factory, where 2 of us went on Friday and 3 of us on Saturday. We had already had a good idea about our app, but none-the-less we got a good deal of inspiration of how to slightly adjust the app and how to come up with new ideas. Unfortunately we weren't fully aware of how valuable the first day would or could be for us. In retrospect we should've attended this event with as many group members as possible, since it was both productive and inspiring for our group. We didn't get much out of the second day.
Michael & Patrick attended the 24 Hours with the Bus workshop. While it didn't help us a lot with our app, it was very interesting to see which challenges the whole ElectriCity-project provides and what is done to deal with those challenges both for the manufacturer and the customer.
Jonas went alone to the Innovation platform event where about half the event was about the innovation platform and the other half was about start-up companies and the reason they fail. The second half was not very relevant to our project and while the part about the innovation platform was interesting and educational it was not especially helpful when working with the project.
We skipped the other 2 workshops, since we didn't deem them important enough for our project. Those who attended an event would then informed the others at the next meeting about the event.

Reflection

Working as a team

While some members of the team had worked with Android before no one had built anything of this scale. That meant we all had to familiarize ourselves with how Android development worked, and because it was something we all had to do it we could talk about what we have learned during our meetings. During this project we had to do a lot of self-learning and one of the major advantages of working as a team in this course was the fact that we could learn together.
Almost everyone in the group were in different places in their education, with 4 GU students and 1 Chalmers student the knowledge areas of the group was quite diverse and someone could often offer their knowledge when someone asked for help.
We got along as a team with everyone quickly becoming friends and having no arguments during the entire course. This made communication fun and easy between us and we all felt it made the course more enjoyable knowing you didn't have to worry about what your teammates are doing.

BusExplorer

We decided very quickly on the basic idea of our app and soon after decided on a temporary name - BusExplorer, which then become the name of our application. We all felt satisfied with the idea and thought it was something that a lot of people would find useful, including ourselves.

ElectriCity

After getting the early versions of our prototype done we realized that we had a real chance to win the ElectriCity Innovation Challenge. For the final day of the contest we had prepared our presentation so that the jury would see that our app could easily be implemented into Västtrafiks systems and that it would be something that would incentivize people to ride the electric buses and that it was economically viable. We all felt the presentation went well and that we got to say most of what we wanted.
For the public exhibition part of the day we got placed next to the team that would end up winning the entire contest. We saw that our concepts were extremely similar but felt that ours was a bit more developed. We discussed how their app was more aesthetically pleasing and their video explaining the app was very well made, while we had trouble creating our video and ended up with

one without any audio. We agreed that this team would be our biggest and perhaps only competition in our category. In the end their application beat ours and we went home feeling disappointed but determined to develop our app further to a releasable state.

Conclusion

The very first challenge we had to conquer would at first seem to be one of the biggest, how to work as a team. This challenge turned out to be one of the easiest challenges we would overcome. Using Scrum gave us the tools to organize the project and we could later modify the Scrum model to what fit us best as a team and made getting started as a team very easy.
The course combined with the Innovation Challenge has been very educational in showing us how a project would work outside of the school environment and how a team would work together to reach a certain goal, which in this case was developing an Android app, from scratch with little knowledge about Android development to beginning with. In the end we felt very satisfied in what we accomplished and even though we didn't win the Innovation Challenge we are determined to continue with BusExplorer until we feel it would make us proud to release it to the world.