Mei Ha
Dina Zuko
Emma Dirnberger
Martin Augustsson
Simon Fransson

Refl.SE

Introduction
In the project, most things were new. Developing for android cellphones is an area which none of the project's members had been into before. Before the development of the project really took place, we had instructions of how different "meeting"-structures took place (mostly Scrum). As a consequence of the fact that most information on a lecture were about Scrum we also chose this, since there was less information about the "Waterfall"-structure and that Scrum gave you more time as a developer. Our general desire in the group is to make progress as a software engineer and learn as much as we can from the different programming languages, thus less time for documentation seems more appropriate than less time for programming.

Scrum
The methods of Scrum focus on daily meetings, short but informative documents and a weekly plan. The members of the group had different schedules, different obligations outside of school etcetera which made it impossible to have daily meetings. We had two official weekly meetings instead, on Tuesdays and Fridays, where we went through the sprint plan of present week on the first meeting and on the second meeting we did a check-up where the members could receive help and explain the status of their development.

Mainly, Scrum was developed for companies and for use within the job market, not for school projects. This made it impossible to fulfill the requirements of this technique. We are not working with a client at some company, we are students with other subjects to focus on too than only this project.

The length of a meeting varied depending on whether somebody in the group had to leave school at a specific time or not. This was not always positive, since one member or more of the team would miss out on the opportunity to receive help or simply miss important information. The exactitude of peoples' arrival and the actual start of the meetings could also vary, depending on personality and environmental interferences.

Two additional, unofficial, meetings were implemented for the availability to develop features or write documents together with the rest of the group. Even though not all members used this opportunity or had the need to do so, it was a good functionality to have.

Additionally, we decided to have a weekly Scrum-master instead of having the same throughout the project. The decision of the Scrum-master for every week was made on the Friday meetings. The responsibilities of the Scrum-master was in major parts to be in charge of the meetings' agenda, make sure what have been said is written on the Google-driver, structure the weekly sprint plan etcetera.

The concept of what Scrum is and what we were supposed to do if we wanted to adept this software development technique were not clear, at the beginning. After the lecture about the subject it seemed easily achieved to follow Scrum, but we did misunderstand some parts which lead us to redo necessary parts more than once.

The planning of the sprints was the greatest challenge. Creation of the product backlog was relatively easy, but to estimate how much time each feature in the backlog would take was not as easy. Therefore the sprint planning we made at the start of the project is not equivalent to the actual sprints. As the progress of the application escalated, additional features and problems occurred which made the sprints longer than intended. Comprehension for our development and its complexity grew with every week and the sprints became more accurate to what we wanted to achieve.

Opinion of the group is that Scrum is very useful for planning and structuring your development, you always have a predefined plan on what to do. It also got focus on letting every team-member be seen and heard which strengthen the social aspects of a group.

Additional tools

For the sake of deepening the bounds of the team members we included a tea-break-master on a weekly basis. The tea-break master was decided on the Friday meetings too, just like the Scrum-master (they would not be the same person though, except for once). The task would be to bring some pastries on the Friday meetings, which that person had baked him/her-self. The purpose, except for the mentioned one, was also to have this as a sort of reward for one week's hard work. All of the team members appreciated it very much.

A facebook group-page was created for the team-members only where we could share links, information or ask for help. Nowadays, this is a common way to communicate within groups of different kind. It is positive for the simple ways of reaching to everybody you want to communicate with about certain matters, as long as you are registered on facebook. The negative aspect is that you might miss a post from somebody if there are several posts displayed and that you might forget how to communicate with the group-members through other tools. We never exchanged mobile numbers within the group, which would made some communication easier. There was sometimes a bad communication within the group.

Instead of having facebook-pages you can create a sms-group where you text people added to the group at the same price as a simple sms. We never used this, but this is a communication tool that can simplify communication for all people involved in a project.

None in the group had previous experience of GitHub and therefore it took some time for us to learn this program. For some team-members the understanding of Git took a bit longer than the others, but all understood it at the end.

A negative aspects of git is that it is a lot to learn and remember, many commands for each purpose. A tired mind can easily forget some parts and therefore create a disaster in the project's workspace. Since Git is very complex with several parts to learn made proper usage of it a time-consuming process which was a problem. Some members focused on other parts of the project at start.

Using Git have several positive effects for the project as a whole, once you know how to use it. You can use data in the workspace and develop on your own by using branches, without fear of ruining the project. You can check branches in the project by simple writing a command. Possibility of resetting the branches to previous submissions was a relief when damage was done to the application.

Mei Ha
Dina Zuko
Emma Dirnberger
Martin Augustsson
Simon Fransson

Conclusion

This project let us take a hint of what our working life will be in the future. It also introduced us to new techniques, like Git and Scrum, which may be tools to manage later on as well. The new knowledge will be useful for everybody in this course and will help us finish projects we will be involved in.