Phoenix Simulator

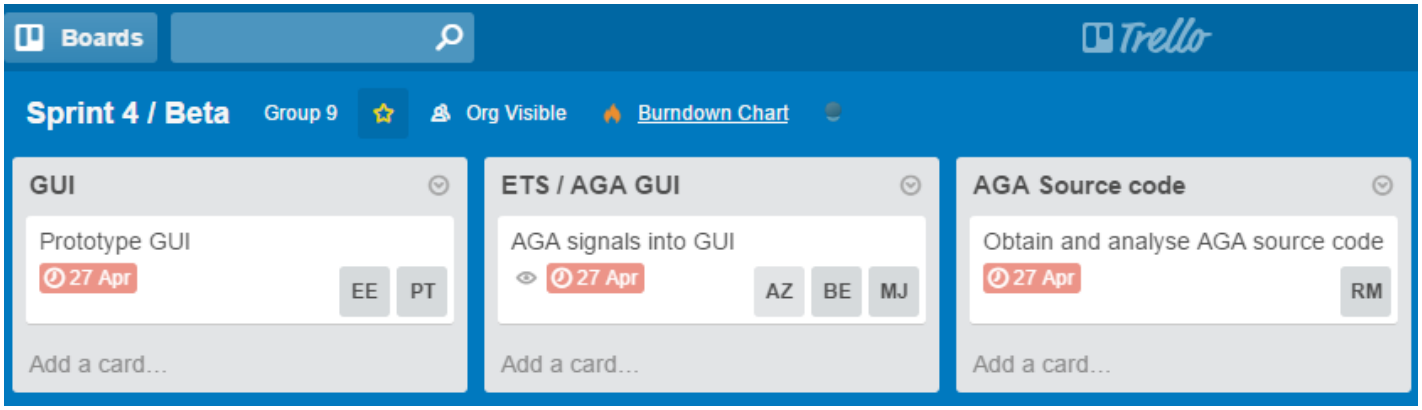
Post-mortem report

Alexander Zajac
Benedikt Einarsson
Eyþór Atli Einarsson
Michael Jones
Petronela Tănase
Rafael Martins

Processes and practices

When the project started all groups were informed that the software process would be agile with the use of scrums and sprints. Scheduled meetings every fortnight were set by the teachers and supervisors. During sprints, the group planned the next sprint and did a retrospective of the previous sprint. Every sprint meeting helped organise the work and see what could have been done better or differently.

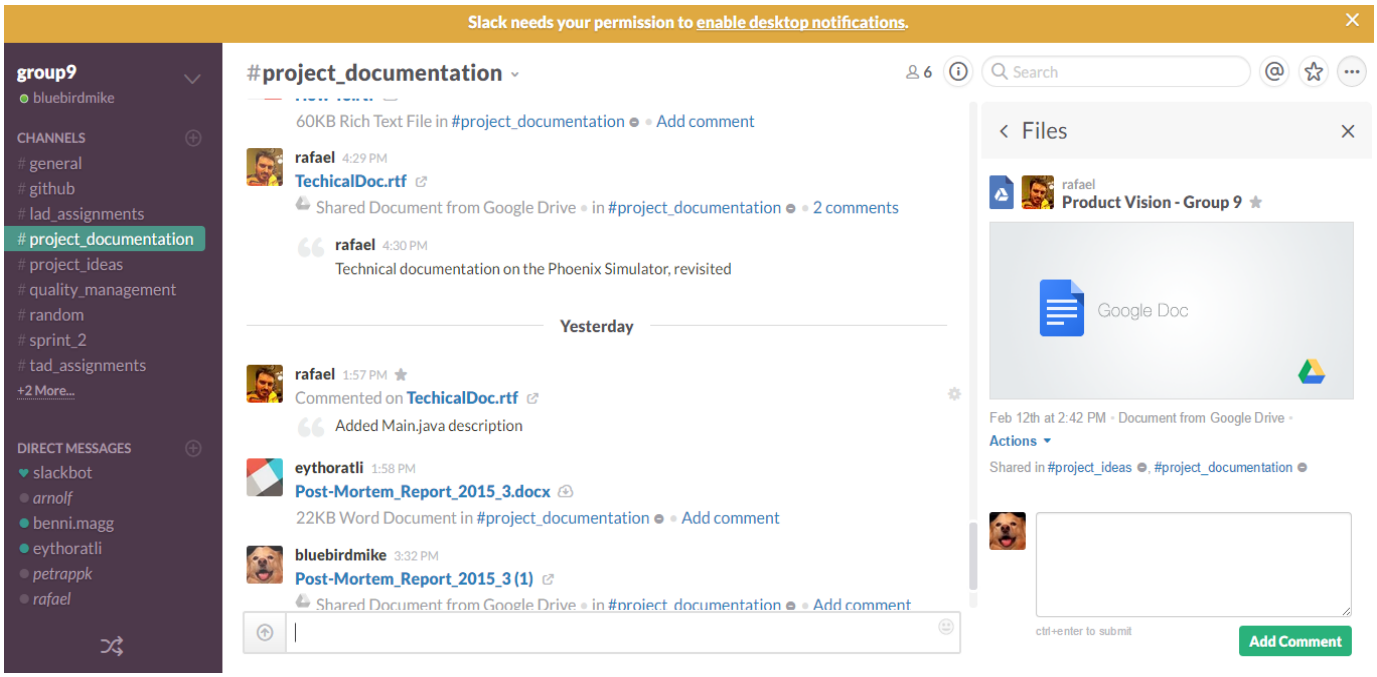
Whiteboards were used to write down any new ideas that occurred while brainstorming. When a decision had been made and the ideas had been evaluated, the upcoming tasks were divided amongst members. Tasks were allocated to each individual via the use of Trello, a free web based project management application. Sprints boards were created for each sprint and Trello was used in order to keep track of the overall process, however, it only became the main tool for managing sprints halfway through the development process.



Regarding writing the code, pair-programming was generally the first choice. Although occasionally certain tasks were solved in an individual manner, pair-programming was the practice used most often.

Since all group members were present at school most days, the main type of communication was face-to-face. Ideas and others messages that needed to be

communicated rapidly were shared through the group's chat on Facebook, which was used on a daily basis. The files, codes, documents, etc. were uploaded to Slack. This served as an archive with different channels as shown below. Slack was a highly effective tool and would certainly be used in any future projects.



Github was not used to it's full potential during the project, we resorted to using Slack as a repository for large periods of the project, particularly the final week. Difficulties were encountered when work was done at the same time on the same subject and it was laborious to keep track of the different versions of the code. It was then understood that managing Git should have been handled in the beginning in order to make our work easier.

That is something to learn from and take to the next project - the effective use of Git.

Scrum meetings and sprints

At the beginning of the semester, the time spent on the project was limited due to other courses occurring at the same time. The project was left behind when the time required for completing the other courses was more than expected.

The first step was to set up Slack and GitHub accounts and started brainstorming for ideas. After many good ideas it was decided to create some form of game for truck drivers which would help improve their driving skills. Apart from this, the thought was also to help businesses keep track of their employee's progress; time management, kilometers driven, braking and fuel economy amongst others. Using this, the company could possibly reward their drivers for good driving over the month or year. It would also motivate drivers themselves to improve, and give them challenges and potential rewards for good driving behaviour. The other idea was a quiz game for truck drivers to play whilst driving; with a focus on the application being safe but also entertaining. This idea was disregarded as certain members of the group were not in support of it.

The idea of connecting an application to Euro Truck Simulator 2, to get more accurate data of how the driver is actually doing by measuring a truck as opposed to a racing car, was always something the group had in mind. Although it was a great idea it turned out to be a challenging thought process.

The first sprint was all about starting the initial project idea and writing a few user stories. Getting in contact with the people from AGA to get access to their source code was close to first on the list. It turned out to be very difficult to access the source code, and it could have been dealt with better, i.e. access to the source code should have taken less time than the couple of weeks it took. This also meant postponing it to the second sprint meeting.

Sprint number two had the focus of AGA - the priority was to analyse and understand the signals. Trello was set up during this sprint, it turned out to be very helpful, since it kept all assignments individually for the group to follow and execute. It also kept the documentation well in line and accessible for group members at any given

time. Using an actual sprintboard gave the members of the team's deadlines to work to and reminded them of their exact task. The GUI work for the application was kickstarted and the plan was to have a mock-up ready for the next sprint. A meeting was held with Christopher Pavlic from Combitech who works closely with the AGA project. He explained the architecture of the AGA source code and also gave some ideas on how the simulator could be modified. This meeting was tremendously useful, although it still took a considerable amount of time afterwards to get the source code set up and ready to work on.

When it came to sprint number three everyone was busy with upcoming LAD and TAD exams, as well as weekly assignments. It didn't take long to realise that every group member had to take a break from working on the project, and focus on the other courses.

The project became the main focus during sprint four; after all the other courses ended. Finally there was the smell of the roses and the project began to come together. The source code difficulties were resolved and there was a breakthrough in analysing and understanding what needed to be done with the AGA simulator. The progress of the GUI was not a success since many screens that were developed were not used in the end. The AGA connection to the GUI was also unsuccessful.

During the fifth sprint, the GUI was made and remade a couple of times, therefore many screens that were supposed to be used for the final version of the application never made it through the bottleneck. Things were not coming together as hoped, however, it was noticed that the GUI was made as an "activity" and it had to be changed to a "fragment" instead. After that change everything started to fall into place regarding the GUI. Many things in the back end were discovered as well in the code, so changes could be implemented to make things simpler. That came in strongly for implementing the GUI and merging everything together to make a running application.

During the whole process the whole group worked together. Most of the times in pairs, but also occasionally as three in a group. During the GUI development, a long time was spent on researching fragments and that is something the whole group worked

on together. In the end there were four user interfaces made before realising that fragmented screens were needed for better “real-time” usage of showing the signals from the Phoenix Simulator. The last week (also the last sprint) was hectic but everything was up and running after a lot of split decisions and throwing stuff in the “trash”, however, the app had strayed a little bit from the vision and many things that had been planned were not executed.

Techniques

The communication part, which was mostly face-to-face contact, worked out very well, and Facebook chat came in handy otherwise. All members of the group were usually at the same place working together but on separate assignments. Working with Github was not without problems and it was all but time efficient instead of helpful. There were at least two or three days that were spent on trying to get GitHub working, and in hindsight it was not worth the time put in it. On the other hand, if working with Github would have been solved it would have been a tremendous help for documents and files. The group agreed that before the next big project, GitHub is something everyone needs to take a closer look at for further projects so it could be used properly as the main file-sharing place. So it came down to using Slack mostly for file-sharing and that turned out to work quite well, but again it would have been better if GitHub had worked out as was planned.

Working in Scrum was good for the most part, and the sprint meetings organized by the teachers were very useful. The supervisor was also helpful and throughout the course of the project she gave many good suggestions on how to handle different things. It helped get everything organized, as well as making the project float in the right direction by having assignments mapped out to complete before the next sprint.

Pair programming was used at times, but could have been used more often. It was helpful in some cases, like working on the GUI and then working on getting the signals to work. It could have been used more often and if so there could have been a

“better” product and since this is a learning process that is something that will be helpful for the years to come. Although while using pair programming a lot more than was initially anticipated, it still could have been used more effectively to meet the vision and efficiently connect more throughout the project.

A database was needed for our app to keep record of journey information (error information, etc.). SQLite was used as we only needed to store the database locally on the android device.

Overall experience

What went well in the project was mainly that the outcome was a working product, even though it might not have been as the vision indicated. The group worked well for the most part, however personality clashes and friction were sometimes present. Overall the group performance was smooth and the team synergised well. During the most hectic weeks there was a lot of stress but everyone stayed calm and worked towards completing their individual tasks,

The thing that did not go so well for example was following up on the initial vision. The group work, as previously mentioned, was not always the best; ideas were conflicting a lot as it often does in group projects. Although there was a lack of group communication at times, the documentation was on point and everyone was following what was set out to-do. In spite of these problems our project came together.

Non-process decisions

Most of the time Scrum worked quite well for the group and most of the decisions made during sprint meetings were followed through. However that was not always the case. During time constraints towards the end, many decisions were made on-the-fly so to speak. The concept of our GUI and features of the actual app were frequently changing and this made it difficult for some of our team members to work on their tasks. Other

features such as the app database were dropped due to having insufficient time to integrate them into the app.

Future projects

As usual, there is always room for improvement regarding how to handle the management in a project. As a team, despite the very few moments when all the members were under pressure in terms of delivering a good product, group members managed everything quite well. The group as a whole will take this experience further and improve the next project regarding good management in a productive environment.

In a future project, it would be better to approach the product development in a slightly different manner, which would be to establish from the beginning a clear and well-defined structure of how everything should occur. Although it is common for changes to appear with every step, these should not affect the structure of the initial plan, and by no means to completely change it. This would save a lot of time to make the project better and keep everyone motivated and focused on the project. It would also decrease the amount of work that had been done that would end up not being used.