

Post Mortem DAT255

Andreas Wahlström

Anton Bergman

Christoffer Matsson

Emma Håkansson

Emil Kristiansson

Linnea Andersson

October 27, 2013

Abstract

This is the Post-Mortem report of our project group in the course DAT255 at Chalmers University of Technology. The project spanned eight weeks, during which each group created an Android application from scratch. Each group got to choose freely what kind of application they wanted to create with some help from the teaching assistants. This report will cover the process of creating a Risk-style game for Android and our thoughts about the different techniques the group tried to use during the project.

CONTENTS

1	Process and Practices	3
1.1	Time Spent	3
1.2	Planning	3
1.2.1	Advantages and Disadvantages	3
1.2.2	Efficiency	4
1.2.3	Thoughts on future projects	4
1.3	Pair Programming	4
1.3.1	Advantages and Disadvantages	4
1.3.2	Efficiency	4
1.3.3	Thoughts on future projects	5
1.4	Meetings	5
1.4.1	Advantages and Disadvantages	5
1.4.2	Efficiency	5
1.4.3	Thoughts on future projects	5
1.5	Implementing parts of Scrum	6
1.5.1	Advantages and Disadvantages	6
1.5.2	Efficiency	6
1.5.3	Thoughts on future projects	6
1.6	Group Synergy	6
1.7	Future Projects	7

1 PROCESS AND PRACTICES

1.1 TIME SPENT

Each member of the group spent around twenty hours per week working on the project.

Emil and Anton spent most of their time working on the model and controllers adding new features and fixing bugs together with Linnea and Andreas who focused on the view and controllers.

Due to everyone except Linnea experiencing technical difficulties with running unit tests Linnea also spent a lot of time writing tests and testing the model.

After the core of the model was implemented by the whole group collectively Emma and Christoffer were put on the task of creating server functionality. It was decided that we would use google appengine and google cloud messaging but since no one in the group had any experience with building a server we all knew getting started would take some time. It turned out to be even more difficult than expected and took up a lot more time. After two weeks of getting practically nowhere Anton was assigned to help with the server. After another half week the server work was mostly abandoned but Christoffer kept trying and finally succeeded in saving data in datastore. To implement the server functionality into the game Christoffer and Linnea teamed up during the last weeks of the project.

When no longer working on the server Emma and Anton started finishing the project documentation and writing this post-mortem report.

1.2 PLANNING

We figured that we could get a head start with the project if we started and finished the planning phase as early as possible. A lot of time was spent during the first week to make sure we had a finished vision of what we wanted to accomplish at the end of the project and to make sure that week 2 and onward would be as productive as possible.

The group started out setting user stories as goals for the project, but it took us some time before we started dividing the user stories into tasks. A few weeks into the project we started using `www.trello.com` to keep track of what tasks needed to be done each week and who should be doing them.

One sprint were done each week with planning meetings on Mondays and due dates on Fridays.

To start planning and making decisions as early as possible was certainly a good idea but we feel that being given the lectures on project management and project processes earlier in the course would have helped us significantly in structuring our project.

1.2.1 ADVANTAGES AND DISADVANTAGES

As the user stories grew bigger and less concrete it became a disadvantage to have them as the most detailed plan of what we should be working on during the week.

Another issue was the definition of when a user story was done. Labels such as "Done but not tested" were created in Trello which was a great way of keeping track. In early sprints

some of these items would be left at the end of the sprint due to lack of time which delayed the start of the next sprint.

1.2.2 EFFICIENCY

It would most likely have been more efficient in the early sprints to have spent more time planning who should do what specific tasks. This would have spared us the down time when you don't know what to do next or when you have to wait for someone else to do something in order for you to continue.

1.2.3 THOUGHTS ON FUTURE PROJECTS

In a future project we would put some more effort into the planning for every sprint throughout the project while still getting the ball rolling early.

1.3 PAIR PROGRAMMING

The group started out thinking that Pair Programming would be the chosen way of working throughout the project. We divided the project group in three smaller groups of two people. Each group had different areas of expertise. To get started, we divided the model between all the groups to have the backbone of our project done early. When this was completed we split up larger areas between the three groups, such as the network part, the controller & model, and the view.

1.3.1 ADVANTAGES AND DISADVANTAGES

Obvious advantages with pair programming is that when working, you always have another person to ask and review code with. This was to our advantage when working with experimental things, such as the server part of the project. (Which none of us had worked with before). What we found negative about pair programming is that when you both have different schedules due to other courses it often gets difficult to plan your time. After a while we realized that planning time to work together at the same computer was getting increasingly difficult, and that it was getting in the way of the progress of the project.

This led us to the decision of having two people focused on the same area, but not having to work together all the time. Communication came to happen with meetings within the small groups or mostly over Skype while working.

1.3.2 EFFICIENCY

Working efficiency got significantly better once we made the decision to not use "real" pair programming. Each member of the team could just pick up their computer at home and start working without having to schedule together with their partner. This worked very well as long as the 2-man groups kept up good communication with each other, which they mostly did through Facebook, Skype, Google Drive and such.

1.3.3 THOUGHTS ON FUTURE PROJECTS

In future projects we will probably look over every member of the teams individual schedule before deciding to use pair programming. If the technology to be used in the project is new to everyone, pair programming might still be the chosen way of working since it makes it so much simpler to discuss and learn.

1.4 MEETINGS

Throughout the project we have used different kinds of meeting techniques. The group started out using stand-up meetings, but phased them out as the project moved on and the 2-person groups started communicating more with each other and less with the rest of the group. We moved from stand-up meetings to having one or two "design choice" sit-down meetings every week. There we would talk about what to do this week, and who should be doing it. We also discussed if there were any problems during last week and how those problems could be fixed and avoided this week.

If anyone had problems or faced difficulties and needed someone else to look at it, they would simply post it to our Facebook-group and as soon as someone saw the post they would take a look at it and respond.

1.4.1 ADVANTAGES AND DISADVANTAGES

Meetings are good because everyone gets to know what they are supposed to do. Everyone got to voice their opinion and come with feedback on what the others in the group has done, everyone got involved.

One clear disadvantage with meetings is that they are very time-consuming. Another is that they are very schedule-dependant. Everyone must have a free slot in their schedule, and if someone doesn't everything becomes delayed.

1.4.2 EFFICIENCY

Early on in the project the meetings had high efficiency and not that many sidetrack, this created a solid platform for the project. A well defined meeting agenda was not necessary for the project planning early on. Though towards the end of the project we started to realise that our way of having meetings was not very structured which most likely lead to a drop in efficiency. A common problem we had was that someone brought up a bug and then the group would spend 15 minutes talking about that instead of the task at hand.

1.4.3 THOUGHTS ON FUTURE PROJECTS

The best way of handling meetings in a project, regardless of what technique you are going to use, is probably to decide beforehand, in the first week, what kind of process you are going to use for the rest of the project. Since we experimented quite a lot with different techniques in all areas of the project this was hard for us, and it ended up with us "freestyling" much of the meetings.

1.5 IMPLEMENTING PARTS OF SCRUM

Although we did not follow scrum very strictly we used some parts of it, and we felt that was worth mentioning.

When planning we created a backlog with fuzzy stories, epics and user stories with story points. A burndown chart was updated at the end of every sprint to visualise the progress of the project. In the beginning of the project we also selected a product owner(Emma) and a scrum master(Emil) but these roles were not strongly upheld and therefore were not very significant to our work. As mentioned earlier We also worked in sprints.

1.5.1 ADVANTAGES AND DISADVANTAGES

It was really good to have a prioritized list of everything the application should be able to do. Having everything on the format "As a X i need to be able to do Y" was time-consuming and static to write down. To begin with we wrote user stories for all features we wanted to see in our application, even if they were fuzzy items. The story points were quite difficult to keep track of when we would add or remove user stories. Which in turn made updating the burndown chart problematic although when things were sorted out it was a good tool for keeping track of our progress. The sprints, especially the deadlines, were very helpful in making us structured and getting things done weekly.

1.5.2 EFFICIENCY

From our experience in this project some parts of the scrum process are very efficient but it could have helped us a lot to have known more about it before trying to use it.

1.5.3 THOUGHTS ON FUTURE PROJECTS

We would use parts of scrum in a future project, especially since we know it better now. Still it seems that scrum in its entirety only fits a very specific type of group and project. Best alternative seems to pick and choose the parts you like as a group. Scrum works better for people who work with a project full-time when for example daily stand up meetings would be a great tool.

1.6 GROUP SYNERGY

Even with differences in our schedule the group synergy has been seamless. Our cohesiveness and good communication both in the whole group and in the respective pairs is what made the project work out well.

1.7 FUTURE PROJECTS

- In future projects we would specify a handmade process by picking the best parts of SCRUM, XP etc and then try to stick to it.
- Testing is also one important part we would prioritise more in the future, maybe even write tests before writing the code (TDD).
- Everyone should test their own code.
- Having a test version of the product to make testing smoother.
- When taking on new technology, everyone should be involved or else risking bottlenecks further down the road.
- Have more structure on meetings including a well defined agenda and a person responsible for organising the meetings. This person should also be responsible for killing sidetrack threads during meetings.