Post Mortem

Group 13
Tobias, Per, Andrius, Christos, Magnus

Meetings and working

This project started out well for our group having a lot of meetings with everyone present. During these meetings we discussed things ranging from how we should work, what our meetings should be like how often we needed to meet and more specific things about the project. Early in the project we had meetings 2-3 times a week during these meetings we usually worked on other stuff but always had a short time set aside to talk about the project. As time went by and the other courses were close to being finished attendance to the group meetings became worse.

To combat this negative trend we set up a meeting where we discussed why this was happening and what we could do to make sure that everyone attended the meetings. Our conclusion was that communication was lacking and better communication would resolve the issues. After this we started setting the time and date for our meetings earlier and made sure that the information was always available at the same place. This partially worked, attendance was better but the problem was still there. However it worked good enough to keep working with the project.

Around this time in the project we had to put more time into actually doing productive things. We needed to get an ALPHA version of the project finished to show something and later a BETA version as well. These two versions of the app was done by two of the members, the final version was done by four with a little help from the fifth. The sixth member had completely stopped working with us during this time.

To minimize the risk of a similar situation in future projects some guidelines about how to work together, how the meetings will look, when they will be scheduled, how often do we meet depending on where we are in the project, what happens if someone drops out or don’t do their part etc. This would need to be agreed upon at the start of the project. Similar to how we did a social contract last term.

Scrum

This was the first time anyone of us used SCRUM in a project so a lot of what we did was made up along the way or changed if we found that it didn’t fit our group. The short meetings to get everyone up to speed and having a backlog are two things that worked very well for us. Our estimations on how much time our work would take were usually pretty far off target but it still made it easy to see how far we had progressed and what we had fallen behind on. The usage of the backlog however diminished the later we came into the project and during the last couple of weeks it was completely neglected due to stress. The diminished usage and the stress probably has strong ties and if we were to work SCRUM again we should definitely keep to the backlog, work on the things we actually decided on and be harder on the deadlines we set out for each sprint.

By not doing this we ended up trying to create new functions the last few days with time that could've been better spent on fixing existing bugs and making sure that the demoable version of the app worked without problems.

Working from home and as a group

During the project we had a lot time to work alone from home which worked great for some parts of the project and it's nice for certain type of work but we all agree that having one or two half days together in school each week to work on stuff together whatever it might be would've done us a lot of good. Having physical meetings makes communicating easier. It's easy to misunderstand someone's tone in written text or forget to include some little detail that means much in the large context. It also gives a great opportunity to share what you've learned so that the same work doesn't need to be done by different group members.

Being able to work both individually and in a group is important. Not having every day and hour of your week planned gives life quality. It gives all types of people the ability to work when they please, to some extent. As long as we actually had meetings every week that everyone attended this approach worked well for us.

Git

Working with git was a great improvement from they way we worked last term. Instead of sharing new code via messages or the whole project via some fileservice and having to fit new code with old manually every time. Not being able to see old versions of the code and the lack of overview makes you really appreciate using git. It took a while to learn how to use git. There was a lot of different terms that in the beginning looked like they were doing the exact same thing. Alot of uploads and syncs have caused problems when GitHub didn't want to merge automatically but as we learned more of how it worked the benefits became clearer and more in numbers. More than likely no one in our group knows all the features of it yet but it made it very easy to see and share code without being physically next to each other. Just the possibility to look at old code instead of trying to remember how a method looked earlier from memory was a major improvement.

Design

When it comes to UI design is is very noticeable we have different taste in the group and most of how it actually looks has grown fourth from just working on the app. Some design decisions made by single members have been voted down while others have become part of the final design.

How the layout of the different screens look was discussed and agreed upon beforehand. Everything was roughly sketched on a whiteboard and saved for later reference. Another reference for the design has been Google's own pages about material design to make sure that the app feels like an Android app and nothing else.

Vision

When looking back at the vision we created at the beginning of the project it's clear that we had many good ideas but not much knowledge about how it was going to be done. The major functions of the app are still the same with some minor tweaks. Most of the secondary functions also made it into the final app with some being cut due to time constraints and others changed or cut because we found out it was not possible to do it the way we first wanted. This was caused both by our choices regarding how we built the app and how Android works in general.

The best example of this is the way we utilize navigation in our app. When we made the vision and was talking about different ideas we assumed we could use Google Navigation inside our app but is not possible. We found this out when we were creating the ALPHA version of the app. After this we spent a lot of time researching and trying out alternatives to see how best we could solve this as the navigation was essential to making our vision and app work. In the end we still ended up using Google Navigation but launched as a separate app from inside ours. This in turn limited us in how much we could control in the user interface and made us rethink that part as well. In the end we are pretty happy with how it turned out especially with how big of a challenge the navigation was.

Some of these challenges could definitely have been prevented by doing more research before the vision was written while others required us to actually try it out and experience how the system behaved. Another thing that could be done is writing a vision that is a bit more open. Not having requirements in the vision for example and talking more about the general idea of the app.

What we've learned

It takes a lot more than a goal or an idea to finish a project. Cooperation, communication, ambition and motivation are a few things that is needed during the whole project to keep it going forward. Working in a new team like we did where everyone has equal weight in their voice and there is no person who is in charge and can dish out consequences to someone who is not doing their part. Actually agreeing on some guidelines on how to work together is needed to not experience the problems we had or at least have a clear plan on what to do when something happens.

Some other very good learning outcomes are how to use Git and many of its benefits over not using it. How API's work and how to use them in a project. Android programming and some basic PHP programming. We have seen the potential drawbacks of using online content in an app, be it slow a connection or Androids lackluster support for communication with MySQL. Much of what we learned about Android programming and the way you should build an app is not shown in our final version due to us having to re design large parts of the code to implement this. However it is something we now know and hopefully can use in future projects.

Time spent

This is a very rough estimation of how much time we spent on the project.

	Tobias	Christos	Magnus	Andrius	Per	Suzan
Initial Planning	20	20	20	20	20	20
Research	10	10	10	10	15	
Developer Documentation	15					
Database Design	2	2	2	2	2	2

Creating Database			3			
Android Coding	150	110	10	100	250	
PHP Coding	70	70	20	70		
Report	25	25	25	20	20	