DAT255 Software Engineering Project

Reflection for “Mentorite” by CID

Thorbjørn Bonvik	950405-T070, thorbjrn@student.chalmers.se
Niklas Brodén	940628-5313, brodenn@student.chalmers.se
Robert Felczak	930807-9111, robfel@student.chalmers.se
Kristoffer Johansson	830902-4639, krisj@student.chalmers.se
Valentin Lindblad	950522-0252, vallind@student.chalmers.se
Fredrik Mårlind	950309-5177, fremarl@student.chalmer.se
Jesper Nilsson	921208-4793, jesn@student.chalmers.se

Application of scrum

The main problems we have reflected upon over this project is that we did not have a real project manager and we did not follow any structure during the meetings. Had we decided on a project manager during our first meeting, his tasks would could consist of keeping Trello up to date, making sure that the scrum retrospectives was made and also make agendas for meetings. With someone managing the project, the problems of the meetings taking too long and give little would not have been there.

If we instead had as a group made sure that there was a logical and well structured agenda which we all agreed to stick to, the results would have been better. With more efficient meetings we could have used the time to sit down and program with each other in smaller pairs and between them decided on the given tasks. We were all caught up in the project and speaking freely during the meetings, which resulted in them taking so long that half was forgotten. We did not take any kind of documentation during the meetings, trusting that the trello page would provide the results of each meeting, so all the relevant discussions that did not end up on trello was lost.

Also many times we started discussing problems, features and design overall that would not be implemented the same week or even the next but even further away. This also led to time consumption during the meetings. This as well would have been fixed with a proper agenda/leader.

At first the group decided to divide itself into two main workgroups, with front-end and back-end as main focus. The intention was to make it easier to meet up and work together during a sprint. A person was chosen as the chairman for the meetings and another was set to be main contact with the interaction designers.

We came to terms with a social contract. This included time for a weekly meeting and some guides as; do not spam our group chat with one to one discussions. Answer a question in the group chat so all knows you have seen it. All files in this project should be in english. Also to not interrupt someone in a meeting.

People followed the social contract pretty well, with only minor breaks. It was good to have agreed on some rules to follow early on so we could define and comment on when someone did something un-ideal without appearing as an annoying person.

The thing we should have done differently is the planning and designing of the project, as we did not really know that our idea was going to work with the task as the project introduction was in the middle of week 3 of the course. This lead us to become quite stressed and much more focused on actually getting things programmed and done rather than fully designing and planning them first.

We also should have brought in and discussed more with the interactive design students more to not have to redesign things as much as we did. For example we realized that our initial version of profile creation was not very intuitive for newly arrived. This lead us having to redesign and make more than we had initially thought. Although it created a better product in the end, it still changed the planning and what some of us had to work on.

In scrum terms, we got more effort but had the same velocity. Which is a large negative with agile workflow, when things change or does not work. This made us put a lot of time into something that was not originally planned, which is one factor why our product lacked in some other areas.

At two occasions our team had programming sessions. Where everybody sat together in a room and did pair-programming. The ability to communicate directly and make use of each other's knowledge, made it extremely productive. But it was hard to find appointments for more of these kind of sessions.

Another problem we had, that probably made us steer away from the scrum and that way of working, was that we never really knew how our velocity was to be set. We started by setting a easily divided number between hours per week and weight of one week. This resulted in a velocity of 20/week. Then we started set times during the first meeting. But the problems started when we realised that our experience in this way of working was lacking. We did not know how long time something would take and even though we could make calculated guesses they still were guesses.

The freedom of setting the pace for ourselves made the process harder, since we did not know what the actual time for each user stories would be. We started to make bigger user stories with higher weights, which then led to us have a harder time to divide up the project to all the members of the group. That led to some of us not getting much coding to do during the course.

With only a few days left to the presentation we realised that we had done a lot of the code overcomplicated. The reason being was that we had different solutions on how to solve coding problems. When many brains are working together it can easily lead to misconceptions. One of us had an idea to restructure a lot of the classes and do an overhaul of the code. It was a big risk, but luckily it turned out to be a success. The lesson we got from this was to make sure that everyone were on the same page and to have a continuous dialog through the days of coding. Most of the time we didn't meet or talk for almost a week. Here is where Scrum meetings would have fit perfectly.

With hindsight we didn't make a lot of progress the first 2 weeks. But at the time we didn't realise it, we thought that we were on schedule. This is where a burndown chart could have helped us. But it had to be done for the whole project, because then we would have seen that most of the work hours was put in the last weeks. As we mentioned before, our inexperience made it difficult for us to plan and estimate how many work hours that was needed to complete the different parts in the project. However, we can't escape the fact that we could have done better research about app development. Because then we could make a more solid plan that wasn’t made up on pure guesses.

Our unsuccessful attempt to apply Scrum to our project ultimately led us to not truly fulfill our vision. Most of the major parts were completed but some of the functionality that was important to the product were never finished. We did make our own goals and maybe they were too difficult for us to reach given our experience and timeframe. It was not that we didn’t have time but that we used our time poorly. The bad planning also added some extra workload in the middle of the project. However, if we would have used Scrum in a better way we surely would have gotten a better result.

1. Reflection on sprint reviews/retrospectives

Our form of sprint reviews became a part of our weekly meeting because of mismatched schedules. We only had one afternoon every week where everyone was schedule-free, this led us to combine all meetings into a large weekly meeting. Thinking back, we can feel that this might have been one of our biggest mistakes, since we quickly went off-track during the meetings and this might be a cause for us falling away from scrum after a while.

Also, the meetings did not appear defined in any way as they started when more than one person was present at sometime between 12 and 12:30. Beginning the meeting at any other time than 12:30 was a violation of our social contract, but this part was forgotten as the meetings always started around then and were held.

We did talk about what we had accomplished so far at some point of all the meetings. But the Sprint reviews became a small part and we were not aware in the way we should have. It was mostly to see how the progress of the project was going.

We did show the current versions of the app in the beginning, before the whole layout was made it was split in different apps/activities that had their own purpose and own customer value. Later when they had been merged we “showcased” the functions/features that each workteam/person had developed.

Our planning was a bit lacking which also was seen here, we thought that everything was going fine as we saw progress but never really calculated on how much progress that had to be made each week. Had we made a proper planning at the beginning we would have been able to see during these meetings that our work pace was not going to cut it.

The first week we did not really have a retrospective to discuss, so the meeting was instead spent on planning. We realised that our monday meetings were too unorganized and they needed an agenda.

The second week we split the team into two groups that was going to do differents parts that would be merged at the end of the week. One of the groups had great success, while the other had some major technical problems. Leading to a stall of the project. We solved all the problems by uniting in a programming session. Although we had an agenda for this weeks monday meeting, it was not followed very well but it still helped knowing which points had to be discussed.

The third week we were closing in on the half time report and the first project demo. By this time we had a clearer vision on how our application was visually going to look. On trello, we continued the agenda from the last meeting. But when we started, the agenda was once again somewhat forgotten. This was not because we did not have it in front of us, but more because we did not have a president who could keep the group on track. We started to discuss our past week as we were supposed to. But when we realised that we had moved away from the decided user stories, we tried again to reevaluate our project management.

The fourth week the meeting mostly consisted of us talking about what we had started to implement from the feedback we got when we had the demo the previous week. By this time we did not even talk about our velocity and not about the coming week. We just said to keep doing what we had started and merge it when possible. At this point, it started to get hard for everyone to work since the tasks was no longer broken down and split up but instead they were flowing into each other and the structure was therefore lost.

The fifth week was even more chaotic when we did not have a valid product for the last demo. This was still done without any structure to the meeting and therefore we did not have a scrum retrospective at all. We just discussed what we needed to do before the demo, not reflecting at all on the previous week. We inserted as much meetings and group programming much as we could to finalize as much of the product as possible.

The last weeks meeting was two days before the final presentation. We had still not decided on a name and the some code was not completed. This was probably the result of us not working with burndown charts etc. during the process and therefore we had to prioritise our user stories.

In the end, we can not make a proper reflection on our sprint reviews and retrospectives since we never took notes and never properly went through the plannings over the weeks. We never made a written review of each person's sprint during the meetings. Everyone was present during most meetings and we did discuss how everyone was doing, but the focus went most on the programming and what needed to be done rather than really planning in a Scrum manner.

2. Prototype, process and stakeholder value

The point of always be able to deliver something to show for, really got through in the start but over time the bar for implementing new features to show for felt quite high.

It might have been a mistake to dive right in and dividing the project task and features to different views, at least without making a bare-bone skeleton of the app flow to base everything on. This would have made it easier to merge in the feature branches, which was difficult.

Also, by dividing the tasks into “implement this view” we failed to break down the task any further. Our android experience was limited, so it was maybe the uncertainty of *how* to divide it further that led to this, but it did create some unnecessary stalling in our process as a lot of code got coded multiple times by different people and big pieces of work might be hard to do by oneself. Almost all views turned out to be much more work than expected.

This work method also led to our first prototype was spread out over different branches and we needed to demonstrate this using different phones. Connecting and merging these different parts together took some time.

And then, a question that we got in retroperspective his who was really the “stakeholder”? We did get a lot of input from different people and we did our best to adapt to it all.

Especially at the half time evaluation we got confused. Most of the time we spent with the interaction designers and a test subject. This gave us some insights and we knew what we wanted to change. But then another person came and wanted to give us feedback on the application, when we were pretty much on our way out.

Maybe the abstract person “a newly arrived” was the stakeholder? This is slightly unclear in retrospect.

We however took the interaction designers feedback most into our development.

3. Tools and technologies

When we started the planning of this project we knew we had to make sure that everyone could communicate and follow the schedule of the week. The tools we decided on was GitHub, Trello and the Google chat application Hangouts.

Github worked very well for us since all of the group members had used it before. Although we did have some issues with merge conflicts and such it was nothing that couldn't be resolved. We used git with a feature branch style, so every new feature should be in a new branch and when it is complete it should be merged into master. This allowed us to better work on different things on the app at the same time and also makes it so there always is a working version on master.

We used Trello in the beginning of the project but as time went on we used it less and less. The reason for that was that it was difficult to update and maintain it, it took time and no one in the grupp was assigned to do it. If we actually used it, everyone in the group would have a better overview of the project , we would be able to see how much we had done. We would generally gain a better insight and understanding of our project if we had used Trello as it was intended.

Google hangouts was the chat application that we used for group communication, we choose it because everyone had a google account. But it was not the best choice since no one hade used it before, it became another chat app for us and thus the effectiveness was reduced.

Our way of implementing new technologies such as databinding was not really ideal as we worked in pairs and only one pair programmed using databinding to start with, which meant that the code was a bit hard for the others to understand. Also everyone else added databinding to their part of the code for consistency in the code which took some time. So our implementations of new technologies was not really ideal.

We should have had a crash-course during the next meeting or the meeting before the implementation. Also the usage of our backend service; Firebase, was not really brought up on any meeting. Later we wrote a help-class to be able to easier call what was needed from the database. This was also not really explained in the group on how to use but rather one had to learn from existing code on how to use.

4. Literature and guest lectures

We didn’t really use the provided course literature we instead found our own sources of information. With Android development we mostly used Google’s own documentation and stackoverflow. Since we had members that were proficient with git there was no need to read the literature for it since they could show the rest of the group.

However what we should have done was to read the literature on agile development and scrum. When skimming through at the end of the project we found that there were a lot of useful information within. Such as breaking down task and getting a better overview of your project, both of these was things that we were struggling with. This information was also content from the lectures but it is good to have it in written form so that you can review it at your own pace.

Although we felt that the guest lectures gave insights and that they explained some principles used to make a better product it felt very hard to implement them for us. For example Spotify’s A/B tests where they push different versions of the app to different groups of users seemed genial. But at the same time impossible without a large user base, so for our purposes for developing the app the guest lecture were not that useful. Though for later in our lives when we are developing for a large company or similar these insights might prove very useful.

We got some good desgin ideas from the HCI lecture. For example on how the buttons for the views should be placed and different standards etc.

So all in all the guest lectures provided a valuable insight in how software development projects are run on different companies and maybe what we can expect later when we get our first job, but they didn’t feel so strongly connected to what we were working on in so far that knowledge and insight gained from them was hard to apply to our project. In this sense the “regular” lectures we had was much more helpful.

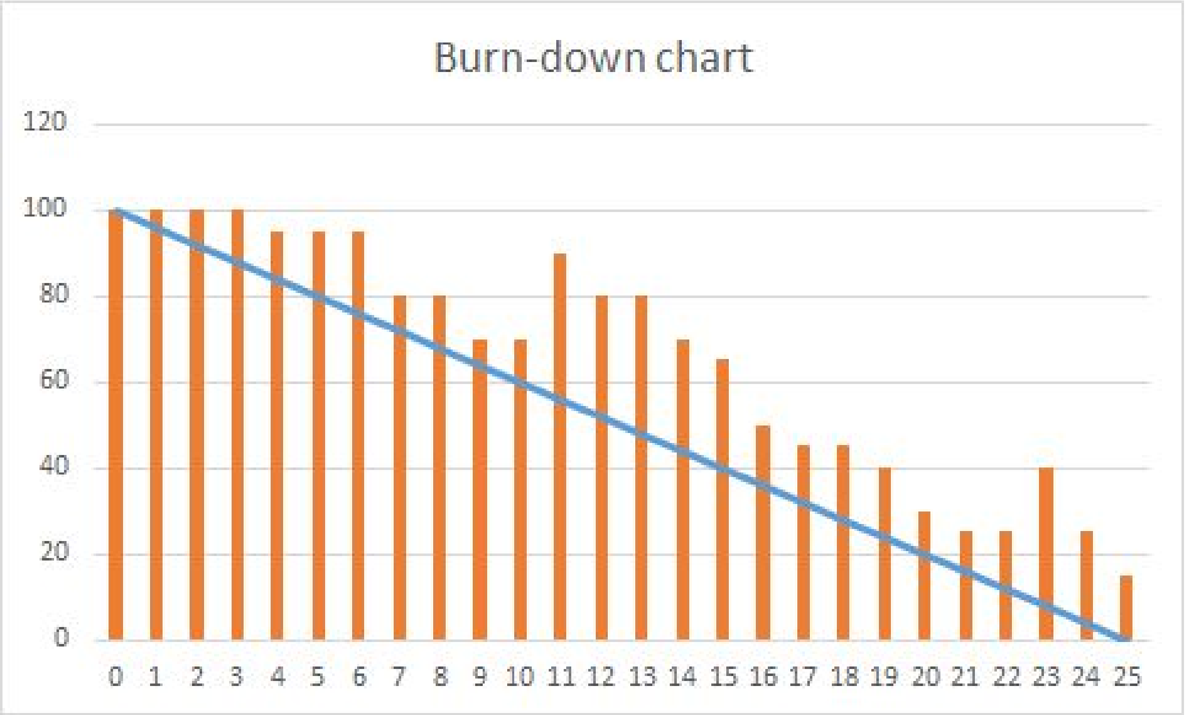
5. Evaluation of D1A and D2

We mentioned in our D1A report that we would divide our tasks into smaller ones. Everyone would either work individually or in pairs on one particular task, in order to work effectively. We also mentioned that our group would strive to have good communication, not to take on too many tasks at once and use Trello to have a clear overview over what can be implemented into our project. Our team seemed to have worked well between D1A and D2 in relation to what we mentioned in D1. We went over our last sprint and what tasks we needed to do for the next sprint. We discussed different approaches to solve problems and used Trello to have an overview on our user stories. However, it was a bit challenging to divide the major tasks into smaller ones so everyone would have something to do. But when some didn´t have tasks that involved coding, those members would start working on the deliverables which is mandatory for the course. The rest prioritized helping on completing the deliverables when the deadline approached.

It may look like we have implemented Scrum well into our project from these two deliverables alone, but after D2 our effectiveness as a team started to decline. We stopped using Trello to go through our sprint reviews and plan our next sprint with it, which made it unclear over which tasks that were finished or not and which tasks that were available. It also made it difficult for our reflection report, where we have to estimate our velocity without a backlog to support it. We realized how important it is to have a backlog and reflect on every sprint review so we would take necessary changes if needed. Looking at this evaluation near the end of the course, these evaluations shows that we were optimistic that our group was doing well, when in reality we didn't implement Scrum as effectively as we first thought.

One thing we should have done during D1 and D2A, would be to have more scrum meetings besides our sprint meeting right from the start (as mentioned was hard because of our schedules). It would ensure that people would be finished faster (or at least finished a big part of it) with having a shorter deadline, in order to prevent postponement. People would more likely to work more hours as a byproduct, regardless of a member´s knowledge and difficulty of the task. We had a brief scrum meeting on the app “Hangout”, but it wasn't very reliable because it didn´t make sure everyone was present at the same time. So it's more effective to have personal scrum meetings, while Hangout can be used to contact and ask others for help rather than having slow, text-based meetings. Another thing that we could have done was to make a another rule in our social contract, so it would be mandatory to do a sprint review before the sprint meeting ends. That would make it easier to do it because it´s a rule where everyone in the team has agreed on, and would prevent us from missing doing a review.

6. Burndown chart



This the burndown chart for the whole project that we did at the end of the project. As we expected we were always behind in our work. We did two major design changes during the project. Which is the increases in the amount of work to be done. This was one of main reason why we never finished.