# Individual Project Reflection - Philip Salqvist

First of all, I would have given myself the advice to spend more time on research and planning before starting to write code. I'm aware that a big art of scrum is to be agile, and thereby give the advantage to test assumptions early before deciding definitely on a specific issue. It is also thereby important not to be to thorough in the plans that are made, you should rather aim to try different approaches and fail often and thus incrementally approach the desired result of the project. But I think there are some foundational parts of a project that should be quite well established before starting to write code. For example, the architecture could have been more thoroughly outlined in the early stages of the project. This would have saved us a lot of time and hassle in the end, since we had to spend a lot of time changing working code when we realized that the architectural aspects of the project had to be changed. I think a more clearly defined architecture would have made the project members more productive in there work, since it would have cleared up a lot of uncertainties that we instead had to deal with continually during the project.

Also I think that the process modeling of the system should have been well established early, as well as the design of the product before we went to deep into programming tasks. Our approach, was to make a few wireframes early on, to establish some sense of the products design, and that this could be used as a template when designing the rest of the product. But in a way, this led us to end up with a product that lacked continuity from a design perspective since different people designed different parts of the product. Having a design established early would also have enabled us to find issues regarding the product earlier in the process, and thus would've given us more time to solve these issues. I think that the best solution to this problem, would've been to meet with the entire group early and discuss the product in detail together. While discussing the product, we could have outlined the process flow on a white board. In this manner, we could have established a clear sense of what the product should do and how it should work among all team members. The simplicity of drawing the process flow on a white board, would also enable us to get the details down in a quick manner, so that the process of defining the product doesn't take up too much time. I have realized that it is more important than I thought to have a clear description of the product that is shared among the team members. I think this is due of the fact that when we discuss ideas in broad terms, without going into details, we might think that we are all in agreement of what we are talking about. But when we sit down and force us to put down the ideas on paper, the misconceptions between team members become more obvious.

Furthermore, I think that some rules that describe how code is going to be formatted, and how the program is going to be organized should be established. A big part of scrum methodology is to not trade quality for quantity. But unfortunately we did this quite often, especially in the beginning of the project. For example, we realized in the middle of the project that different team members where creating very similar components but in different ways, instead of creating one component

that could be reused during the project. This created a lot of extra work and forced us to rewrite a lot of code at a pretty late stage of the project. Establishing an understanding amongst the team members that everything that could be used more than once should be modularized as a reusable component, would have saved us a lot of time, but would've also made the code and design seem more consistent. Although this was a frustrating problem to face during this project, I am now convinced of the importance of writing reusable, understandable and more readable code. I think the reason that we often sacrifice quality for quantity, is because it makes us feel more productive in the short term, although it really renders us less productive in the long term.

Amongst the methodology that we tried, I think User Story Mapping was the most successful one. Working out the users journey when using the product helped us a lot when trying to establish the backlog. It proved to be a very useful tool to first create stories, and later break those stories into smaller tasks. Although I think that this should have been accompanied by some sort of more rigorous process modeling to make the process flow more detailed. Moreover, I think that we could have done a better job with product backlog refinement during the project, since we ended up with duplicate stories and realized quite often that some big function of the product wasn't explicitly explained enough.

Another tool that proved to be very useful was version control with git. The ability to work on your own branch without having to worry too much about effecting the main branch in a negative way gave you more freedom to test different solutions. Also, it provided some sort of structure since all team members only worked on the specific story that they where responsible for at a time. Finally, the ability to create pull requests made sure that the code had to be reviewed by others before it was integrated into the main branch. One area of improvement is the actual quality of the reviews that we performed. Instead of practically testing that the new code could be properly integrated with the main code, we all too often only made a quick assessment of the code and approved it. Finally, it would have been better to decide that multiple developers had to review the code in order to approve it, rather than one.

I think partly that a project like this could benefit of having a clear project client. This would make you more urgent to really follow the sprint goals. We only had a responsibility to our team members that the sprint goal was met. I think a clear project client that would hold you accountable to the sprint goal would create more urgency to prioritize the stories that where included in the sprint. We unfortunately picked a too large sprint goal during the first week. Because the goal almost covered the entire product, the goal became to broad and unspecified which made it quite difficult to prioritize among stories. I also think that it would be of benefit to everyone if you surveyed the class before the project to see what they where interested in. This way you could group people more precisely based on there interest, and could thereby increase the motivation of all team members.