Team 11 - Final Report

Members: Fanny Sjöström, Jacob Westerberg, Johanna Vestgöte Norell, Jonathan Svantesson, Maja Larsson, Thea Kraft, Vendela Palmqvist och William Frisk.

Introduction: We all entered this course and the project associated with little knowledge about agile software development. We also had quite different starting points, both in terms of skill-set related to software development and working in large teams consisting of 8 people with a relatively tight project time frame. Accordingly, this project has brought us many valuable tools and experience about working in agile teams, and this report will discuss our key take-aways.

The Team

Our teamwork has been excellent during the whole project with continual improvements every week. From the beginning we were two distinct groups from two different programs, but the dynamics have changed for the better whilst we have come to know each other. Now we are all comfortable with each other and different group constellations amongst us form naturally and work seamlessly. We have adhered to the social contract with good communication and decision making. As a form of compensation to the late arrivals to our scheduled meetings, fika will be brought to the last meetings during the last weeks. This project experience has been valuable to the whole team, and has taught us a lot about the dynamics and nature of the agile teamwork methodology.

We had three primary learning experiences together while making the game. Our first lesson emphasizes how crucial effective planning is for success, and how using user stories as a tool plays a key role in achieving this. It was important to break down the project into smaller, more manageable user stories to communicate better within the team. During one of the sprints we unnecessarily rushed the sprint planning and as a consequence the user stories were too large and/or unspecific. For example, we planned to implement the quiz about the UN's sustainability goals with the following user story: *"As a curious user I want a quiz so that I can learn about the UN's sustainability goals"*. In the middle of that very sprint, when we started to develop the software, we discovered that this user story was too vague and severely lacked details that could better direct the workload and project planning. This made it hard to split the work amongst the team members efficiently and allow for effective parallel work. Therefore, through this experience, we learned that it is very important to make a plan for the sprint and separate bigger user stories into smaller, more specific ones. This in the end enables teamwork and hence more efficient and effective work towards the final product of the project. The agile workflow with delivering small things often and continuously was preferred to be able to go back and change previous steps, instead of using the waterfall approach.

Our second learning experience is connected to the previous one. In the initial sprint the roles were not well-defined and consequently the workflow was somewhat obstructed and inefficient. Subsequently, in the following sprints, we decided to appoint a product owner. We had different team members as product owners every sprint. The product owner was responsible for setting a goal for the current sprint and deciding which user stories were more important to complete and which were of less importance. Then we decided which tasks each team member was responsible for. To facilitate our work we put the responsible team members initials as a marker on the user story in the Trello board. This marker was put so if any questions arose about a task there was no confusion about which person to ask. The learning experience is therefore that in a larger team - consisting of 8 or more people - it is much easier to make a decision if there is a leader within the group. When all the members are working synchronously in unison, you instead get 320 hours in a week rather than 40. It is sometimes not comfortable appointing a leader whilst in an educational project setting where you see your fellow group members as equals and looking for the same outcome. What we came to find out is that leadership is a vital part of any larger group setting where some amount of work needs to be completed.

Our final learning experience is about general teamwork and how to work in an agile team. We all have different unique backgrounds and special skill sets that complement each other. This ranges from programming, github, design and product management to teamwork and ideas about social contracts. This also means that each member can learn a lot from one another. We have utilized this and did pair-programming to learn from each other and make sure that the software was good and code smells were detected and dealt with. For example in sprint three, we encountered problems with the library which was crucial for the progress of our product. In order to solve it, everyone paused their own tasks and we solved it together like a team. One thing that could be improved until next time when working in agile teams is to set the scene properly in the beginning. To make sure everybody knows about each other's previous knowledgement, but also what everybody intends and wishes to learn would have been useful in the process. That way we could both have utilized all of our strengths and divided tasks according to what everybody wished to learn. It would also have been useful when splitting the group into pairs, thus matching someone with great knowledge about a certain area with someone wishing to learn about a specific tool.

The Product

We are very pleased with how our product turned out at last. We reached our Minimum Viable Product which was our goal, and we also got to do some touch ups. As previously said, we have really been taking advantage of each other's knowledge in the team, which made the work with the product smoother and effective. When looking back on the description of the project scope, it came pretty close in the end with some changes that just came naturally when we started developing the product.

One important learning that we will take from this project is that the choices we made in the beginning might not be the best in the end. We did some research for what library we could use for developing a game in Java and we found one that suited this purpose. We started to

play around with it but as we got further with the development, the library was a bit hard to understand and manage. A lot of things were required for it to work which took plenty of time to understand. We got the hang of it eventually but we did feel that maybe it wasn't the best library for our project afterall. It was quite an extensive library which we maybe didn't need, but it looked appropriate. It might have been easier if we chose a different approach towards developing a game, maybe by using a more extensive game engine, e.g. Unity, which also would've taken some time to learn but maybe would've been easier to work with. Still, the game turned out as we hoped and we learned that the choices we made in the beginning might not be the most optimal for the purpose of the project.

Another observation we did regarding the product was how it changed during the course of the sprints. The iterative nature of the agile process creates a constant discussion regarding the product which often allows for new ideas to be born and for plans to be changed. However we also noticed that working in iterations can be tricky for those who are inexperienced with the agile workflow, i.e us. During the weekly check-ins and discussions it is easy to lose track of some of the fundamentals of the projects, especially since everybody is focusing on their part of the product. For example we feel that in the end we did not emphasize the overfishing theme as much as we would have liked. We also realized that we did not have a decided target group and therefore some elements of the game are a bit of a mismatch. While we believe that the group that would enjoy this game the most is quite young, the questions might be a bit advanced for that group. It was a valuable lesson we learned, that you should make sure that as a group you pay attention to the fundamentals of the product while working on separate details. We believe that the problem might have been relieved if we had chosen to have one designated product owner throughout the entire project. Then the product owner would be able to dedicate more time to make sure that the project is heading in the right direction and have more control over the project scope.

The last important learning we learned was that code quality is difficult to maintain when creating easy solutions to complex functions. This is a consequence of trying to put a product to use as soon as possible, and then when iterating the next cycle setting aside code quality to as fast as possible creating code that works. Our idea from the beginning was to follow various design patterns to keep the quality of the code as good as possible, but as the sprints passed it became harder to follow because of the quick fixed code that had been done. We were aiming to follow the SOLID-principles as well but it didn't work all the way through. If we were to have an agile project in the future, these problems would definitely be taken more into account when starting off the project. The foundation of the software is essential for the development of the game, but even though ours isn't perfect we did our best and we are satisfied with how it turned out. We believe that if our overall knowledge about how to maintain these design patterns was higher, we would probably have succeeded to a higher extent.

Conclusion: Our final conclusions from this course are:

1. A team with great individuals is a key to success.
   Even though we might not have nailed the process itself, we have had so much fun thanks to everybodies positive mindset and willingness to learn about working in agile teams.
2. There will always be changes from the original plan, but with great planning and an iterative workflow, one can manage them.
   All of us thought we could do more in these weeks, but we have managed to restructure and re-plan to achieve a nice product at the end of this project.
3. There are good tools, such as user stories and trello boards, to enable an effective agile workflow.
   These tools might be simple, but yet very powerful, to get started in the agile workflow. Continuing to build on this experience and using these tools, we can all agree that next time working in agile teams we will manage the process even better.