

# Individual Reflection

Pontus Engström

May 12, 2023

## **Week 5 8/5 - 12/5**

Something I would like to learn more about is how to be able to estimate time and complexity of something that I either don't have much experience with or have no idea of the process. I can often be surprised of how easy/hard something was that I thought was the opposite

This week I contributed by implementing a lot of persistence functionality throughout the application. It is now possible to save recently taken quizzes, the score of the quiz attempt is saved and you can query the database on a number of parameters for this such as order by date or score etc.

Since last week's sprint the sprint planning and the process of dividing up work has improved but it can still be better

## **Week 4 1/5 - 5/5**

This week I contributed by looking over our navigation in the app and implementing a NavigationStack in JavaFX, something that already exists as default functionality in many other code frameworks such as Xamarin. Something I can do to help my team going forward is try to make sure that everyone is on the same page before we end the sprint planning to better ensure we keep a working product at the end of each sprint

Since last week's sprint I feel like the amount of times we merge the different branches has increased which is good since we will avoid group members getting out of sync with each other

## **Week 3 24/4 - 28/4**

I would still like to learn more about the structure of user stories, epics and tasks on the scrum board since I did not have that much time to spend on that last sprint and the requirements of the course were changed mid week

This week I contributed to the team by making it possible to save and retrieve quizzes to our Firestore database, connect these to a user and make it possible to query for certain tags or the name of a quiz. Something I can do to improve is to be more proactive when it comes to asking others if they need help since when you are working on something it's easy to get tunnelvision and decide not to ask for help

Since last week's sprint I feel like the communication is better both from myself and between all the members of the team. I update the rest of the team on what I have completed, how it works, eventual problems I've come across and what I will be working on next.

## **Week 2: 17/4 - 21/4**

I would like to learn more about using user stories effectively and more about standard practice when it comes to async programming in general.

This week I contributed by doing a lot of research into Java's available library for asynchronous programming. I worked on refactoring, abstracting and extending the previous week's code so that we get rid of the Technical debt accumulated during the first week's hasty sprint and make it much easier to take on user stories and tasks, related to our database, in the future.

An improvement from last week is that in my “group” we had a clear start and end to our work. We created a few empty classes, divided it up and when we've written good code that is working at least at the same level as last week's sprint. I have also made some progress in understanding more about threads and how we might handle background tasks

## **Week 1: 12/4 - 14/4**

I would like to learn how to better work “incrementally” since I am quite used to sitting down to code for 8 hours straight to get classes completed and working instead of user stories/tasks. Now I feel I have to tell myself to reflect on what has been done and what should be worked on this sprint. Additionally I want to learn more about how Java handles asynchronous programming in order to run background tasks, optimize thread load and wait in critical methods

My assigned goal this sprint was to establish our cloud database and its connection with our code, more specifically users, with Alexander, Felix and Gustav. I setup a basic login and created user functionality with the help of our database library which required some additional setup since their compatibility with JavaFX was apparently not that great. Next sprint we can break down tasks even more or have less people working in the same “area” since there were times when someone did not know what to do or two people were working on the exact same thing.