

Customer Value and Scope

- the chosen scope of the application under development including the priority of features and for whom you are creating value

The goal of this sprint was to make some exercises visible to the user and to make a calculation view so that the user can calculate their BMI. We decided the goal after a short meeting with the project owner during communication that helped us to determine the main goal of the sprint and to decide the important user stories needed for the sprint. The product owner also thought it was important to filter the exercises in terms of body parts so we included that to the goal of the sprint.

- the success criteria for the team in terms of what you want to achieve within the project (this can include the application, but also your learning outcomes, your teamwork, or your effort)

To complete all the user stories and for all 7 developers there were 4 user stories so for 2 user stories there were multiple coders. So for some people it was the first time pair programming together in the group and it went well.

- your user stories in terms of using a standard pattern, acceptance criteria, task breakdown and effort estimation and how this influenced the way you worked and created value

In this sprint we delivered four user stories that were related to the goal of the sprint. they were about the view of exercises and the BMI calculation

- your acceptance tests, such as how they were performed, with whom, and which value they provided for you and the other stakeholders

We tested the product by showing the product owner the results of the user stories and we got feedback that helped us to determine the quality of the work that has been done on the application. The feedback from the product owner was mostly positive so we do not have a lot to modify or change during the next sprints.

- the three KPIs you use for monitoring your progress and how you use them to improve your process

We have 3 KPI. Each person gave a number to their down time spent on the project along with their measured stress level (1-10) 10 being the highest. Also progress measured in user stories.

The group average:

The average time: **5.** (1-10) (10 being 20 hours *and more* of efficient work—the % lowered based on the work time available in the week — with regards to holidays or sickness)

The average stress level: **3.** (1-10) (1 being so relaxed we are bored, 10 so stressed we are reaching levels where we are having serious physical symptoms or feel unable to work because of panic)

The progress measured: **8.** (1-10) (1 being nothing, 10 being all accomplished user stories and tasks—in a satisfactory fashion—assuming that we feel the user stories were a correct work load in hindsight)

Social Contract and Effort

- your [social contract](#) , i.e., the rules that define how you work together as a team, how it influenced your work, and how it evolved during the project (this means, of course, you should create one in the first week and continuously update it when the need arrives)

We made no changes to the social contract--or had a need to pull up or bring up the social contract points of conduct.

- the time you have spent on the course and how it relates to what you delivered (so keep track of your hours so you can describe the current situation)

The group time is on average below 20 hours but we did not have the whole week for this sprint due to easter holidays, we had only three working days. We finished some user stories to get a general sense of how the application will feel and look. Last sprint we worked with “basic” user stories so some members gained more technical skills. This sprint we did more complicated user stories like BMI calculator.

We would like to increase our hours and efficiency. We will reach this by tracking our hours better, finding our footing in the project and the technical newness, and focusing on our user stories and tasks yet even more.

Design decisions and product structure

- how your design decisions (e.g., choice of APIs, architecture patterns, behavior) support customer value

Right now the design decisions remain the same as last sprint and mostly relate to the choice of APIs. There should probably be some architecture patterns implemented as well in the future, but seeing as this is the first time for many of us working in both React and TypeScript, we have had no discussion about this yet. To make such design decisions in future sprints we should continue working, gaining more experience, and see where code can be refactored to create more value for the customer.

- which technical documentation you use and why (e.g. use cases, interaction diagrams, class diagrams, domain models or component diagrams, text documents)

Right now, there is no technical documentation such as diagrams, except for comments in the code. There should be some technical documentation such as diagrams showing interactions or how the api calls work. To get to the point where such technical documentation exists, we need to implement the api to get a good feeling of how it works and how it's used. An interaction diagram does kindof exist in the form of a figma.

- how you use and update your documentation throughout the sprints

The readme has been updated to a work-in-progress readme and is shaping up to be a helpful document for the developers and the users—and a good source of documentation. The goal is for the readme to be finished in time for the product “launch” at the end of the project. This will be achieved by iterative updates to the readme and constant discussions and revisions of it.

Other documentation that we have are the continually updated DoD and codestyle documents.

- how you ensure code quality and enforce coding standards

We have used the git function request to review before being able to push to main. So every contributor has to create a pull request and someone else in the group has to review their code before they can merge their branch with main. This approach is a best practice in order to avoid lots of unnecessary merge conflicts.

Application of Scrum

- the roles you have used within the team and their impact on your work

This week we changed who was the scrum master—which was the plan, to have it as a rotating role. This week it was: Deaa Khankan.

Some of the group members worked together and did some pair-programming. This was experienced as an efficient and educational way of working. This approach had a positive impact on our work as we finished the expected user-stories as well as individuals were able to exchange skills and knowledge with each other. The group members who did pair-programming this week were: Ali and Torbjörn and Magnus, Hedda and Bilal.

Other roles include the product owner—this role has remained with the same person.

the agile practices you have used and their impact on your work

Some members worked extra on Easter holiday so we had some time to work on more user stories that were not included in the sprint at first. So I thought that since we have more time we can change the sprint plan and include more user stories. I did some research and did see that it's not recommended to change the sprint plan. We did a group meeting and decided to not change the sprint. It's good agile practice to work on new tasks when a member is done with theirs and help other members and I see these characteristics in our group.

- the sprint review and how it relates to your scope and customer value (Did you have a PO, if yes, who?, if no, how did you carry out the review? Did the review result in a re-prioritisation of user stories? How did the reviews relate to your DoD? Did the feedback change your way of working?)

We had a meeting with the product owner and it was valuable. We got some feedback about our website like which calculators should be on the website and how good the interaction is with it. The prioritization of the user stories did not change but new user stories were created. We did a demo of the webpage to show each other what we have done. Also tips of what we had learned during the sprint that could be good for the others in the group to know.

- best practices for learning and using new tools and technologies (IDEs, version control, scrum boards etc.; do not only describe which tools you used but focus on how you developed the expertise to use them)

Some members gained more technical experience since we did more user stories related to the application, like how to use GIT, specifically gits branch protection rules to protect the main branch, react and typescript skills. We all used SCRUM to work this week which increased our proficiency in this way of working.

- relation to literature and guest lectures (how do your reflections relate to what others have to say?)

No relevant literature or guest lecture related to scrum this week.