

Customer Value and Scope

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

Our main goal for this sprint was to create a login and make the page responsive as our project owner wanted. With the login came more features that needed to be implemented such as the database (mongoDb) and the server side of the application. This creates a lot of value to our users because they can save favorite exercises and exercise sessions. So they don't have to look up the same exercises every time they start the application.

- 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)

Our goal for this week is to have a database so the user can have his/her own account and save their exercise sessions as well as favorite recipes ect. All These criteria were fulfilled. Some members gained more experience in using mongodb database as well as the communication was much better in sense of cooperation to build the project.

- 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 had 7 user stories at the beginning but 2 of them had the same idea. Some members noticed that and discussed the issue with the group to extract 2 independable user stories from these user stories. Magnus did work with the back-end part of the user story and Deaa did the front-end. The team thinks that it's a very good agile practice to be able to adapt when it's needed. Task breakdown was not perfect this week because we did not have the experience of how hard it is to work with the database. The good thing is we had a meeting at the middle of the week which was a very good opportunity to discuss and fix these issues.

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

It was the same as last sprint, we checked if everything works with the group members first and then we showed the result to the PO to get feedback and see if it fulfills his needs.

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

The KPIs have not been changed this sprint.

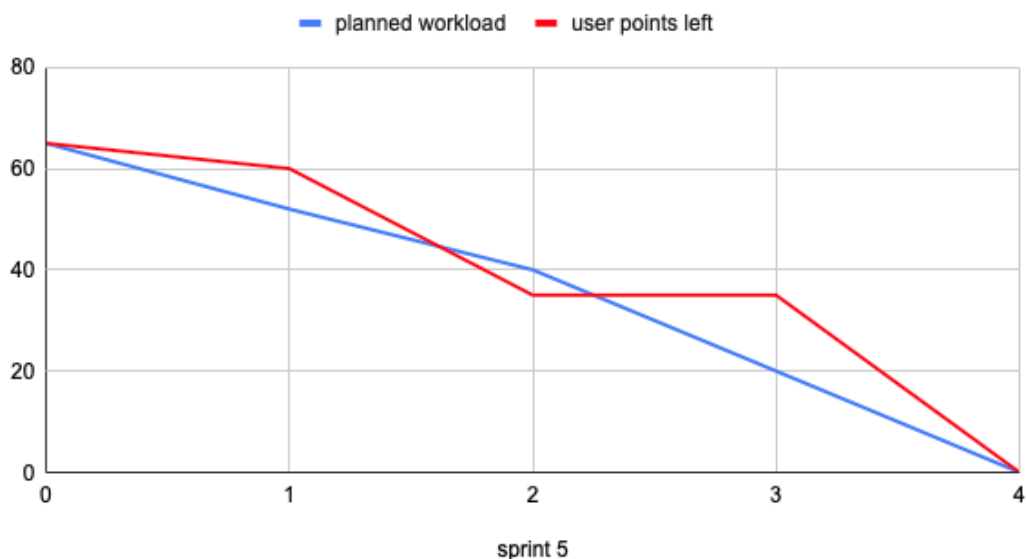
The average time: 8. (1-10) (10 being 20 hours *and more* of efficient work—the % lowered based on the work time available in the week — with regards to holidays or sickness)

The average stress level: 4. (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)

Velocity: **15** (this is the sum of the weight of all user stories.) (last week: 10)

burndown chart sprint 5:

planned workload and user points left



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)

This sprint, as with previous sprint, the group has used the social contract as its foundation for how we want to work as a group. During the sprint planning this monday the group members looked over the social contract to determine if anything needed to be changed or added. The group reached the decision to add a clause about prioritizing meetings on wednesdays. This was implemented during the week and numerous members felt that it elevated the way the group worked. The change will therefore remain, and the Wednesday meeting will be very much prioritized in the future sprints.

- 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)

Compared to last week all members spent more time on the project. Most members were close to spending 20 hours. Having a higher velocity this week (15 compared to 10) is a possible explanation of this. Coming sprints the team would like to continue to spend around 20 hours on the sprint. This will be achieved by having well thought out user stories so that everyone is able to contribute and there are tasks available.

Design decisions and product structure

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

This week, alot of design decisions were taken with architecture patterns, apis, databases, authentication and server architecture. The decisions were made to support and create value for a user by offering features that would create value for them, according to the PO. A big part of the login was choosing what database we should use. We chose Mongodb and Express to talk to the database known as a MERN stack. The server uses express because it is quick and easy to set up. It also uses some middleware to support authentication easily. Tha main structure for the backend is a server that receives requests and forwards the request to each respective controller, the controller then uses a model to create or get data from the mongoDb database. The decision to use mongoDB was taken due to its popularity, it's easily scalable and widely used, and thus us using it creates a lot of value not only for the customer but us as well as we have to learn it. It is easy right now to add more functionality. It is easy right now to add more functionality but as more is added, the complexity of it all will grow pretty quick. Thus to make it

even easier to add features and content, one could create microservices out of everything to make it more scalable, although this might be overkill. And this should have been taken into consideration before creating all of it. In the future design decisions should be chosen to maximize customer value but also take into account scalability, reliability and best practices to create value for us as well, either by making it easier to work or making us learn something new. To make it this way, some research needs to be made before making a design decision and a group discussion should be undertaken so that everyone is on the same page and understands why we're choosing the thing we're choosing.

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

Most of our technical documentation created this week was about the server side of our application. How to setup and use mongoDb and connect it to our front end. The documentation exists in the form of texts and diagrams, the text describes the overall structure and architecture while the diagrams are mostly sequence diagrams describing sequences of events that happen and how servers will react to different events. There are also some guides added to describe how everything works. As the complexity of the backend grows it's important to keep track of everything and what it does. So in the future we need to add diagrams and documentation for most of the features added that are complex. To achieve this, we need to keep reminding each other of our DoD and to make sure documentation exists when creating something new or complex.

- how you use and update your documentation throughout the sprints

The main documentation being used generally right now is the DoD and the code style document which are continually updated. We update the use case diagram also when changes occurs on the application so it is clear what new features is added. During this sprint we started with the mongoddb database so it was important to make a database diagram and schema so it is clear what collections and tables there are in the database. We also created a standalone technical documentation document that describes some parts of the application through text and diagrams. This was continually updated during the sprint when features were added to make sure it is easy to understand for people that are going to work on it in the future. We should keep updating our documentation as the project progresses.

- how you ensure code quality and enforce coding standards

With our code style document we ensure all use the same "coding standards" and when a developer pushes his/hers code to github another team member

reviews the code to make sure it looks good. Some developers use Prettier, a code formatter to make sure the code format is the same in all files.

Application of Scrum

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

This week our scrum master was Magnus. The rest of the roles were as usual. This week our scrum master decided to have a longer meeting mid-way through the week to check in with everyone and see how the progress was going. This had a big positive impact on everyone's work.

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

The decision to have a meeting mid-way through the sprint was taken due to the fact that last week the group felt that it would have had a positive effect on last week's sprint. Previous weeks the group had not felt the need to have either daily check-ins or a longer meeting mid-way through the week. We have used agile practices by changing the way the group works throughout the sprint and adapting to different needs that arise from the group as a whole. This practice had a positive impact on the group. Also some adjustments to user stories had to be made during the week. Two user stories were too much alike and with another one there was also a problem. However, by having a clear communication between the parties involved and implementing agile practice it was solved smoothly and the group was still able to deliver said value by the end of the sprint.

- 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 will be talking to the PO after the sprint review--because they were not available before then.

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

Discussing literature (source:

<https://chalmers.instructure.com/courses/18176/files?preview=2059847>)

Facilitating Entrepreneurial Experiences through a Software Engineering Project Course

(page 31, 34, and 36)

According to (source) a recurring “issue” or source of complaint of previous versions of the course was the high level of autonomy, and especially the high level of technical autonomy. This complaint has existed somewhat in our project too, for some more than others--with regards to the technical aspect--and similarly for all when it is about project and management autonomy. We found ourselves often turning towards each other in moments of confusion, and still lacking answers. Sometimes the answer was to contact PO or supervisor. But more often than not, we had to accept some ambiguity and trust that the chaos would feel less so once we got further in. And, in a few ways--with regards to technical know-how, and scrum methodology, this seems to have proven true. This process of moving from chaos to workable order reflects closely to what the Lecturer said in the beginning of the course, that we will reach understanding through the early confusion.