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 main goal of this sprint was to implement the API for the exercises so that all exercises are shown. In addition to this, a goal was to make the page easy to navigate. This was done by having a start page that was clean and contained an overview of the application as such this creates value for the user since using the application becomes easier.

And, importantly, all main pages now exist in some level of implementation. This delivers great value as these are usable pages with clear defined uses.

In coming sprints we want to continue developing the application in terms of easy and clear navigation. This will be achieved by focusing on the design of the application and making sure that the input from the PO is met.

- 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 sprint goal was to finish our user stories, All exercises from the api should be shown. It should be easy to navigate around the page. All these goals were fulfilled and more. We also learned how to fetch data from an api which was a thing that most of the members did not know how to implement. As well as some design experience and how to style the website with best practice. For exempel minimise the navigation when its possible

- 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 6 user stories that were related to the sprint goal. One user story was about showing all exercises for the user and this means getting the data from API. Another was about a landing page which was to help the user to navigate easily. We also had a user story about showing another calculator for the user and about the food recipes.

Our effort estimation influenced how many (and who--those with more or less experience) user stories were “taken” by a group member. Some of the more complex user stories were then co-programmed, while some of the “easiest” ones were taken by single developers with less experience.

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

We checked the results with the group members and decided together if the user stories are correctly implemented and then we had a meeting with the product owner to get feedback and to know if the results fulfil his needs.

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

Even though we already had three KPIs the group felt that something was lacking in terms of tracking process. As such, this week after discussion within the group (and in discussion with the supervisor) we decided to implement a new KPI; using a burndown chart. This decision was taken so that in coming sprints our process becomes more clear.

The group average:

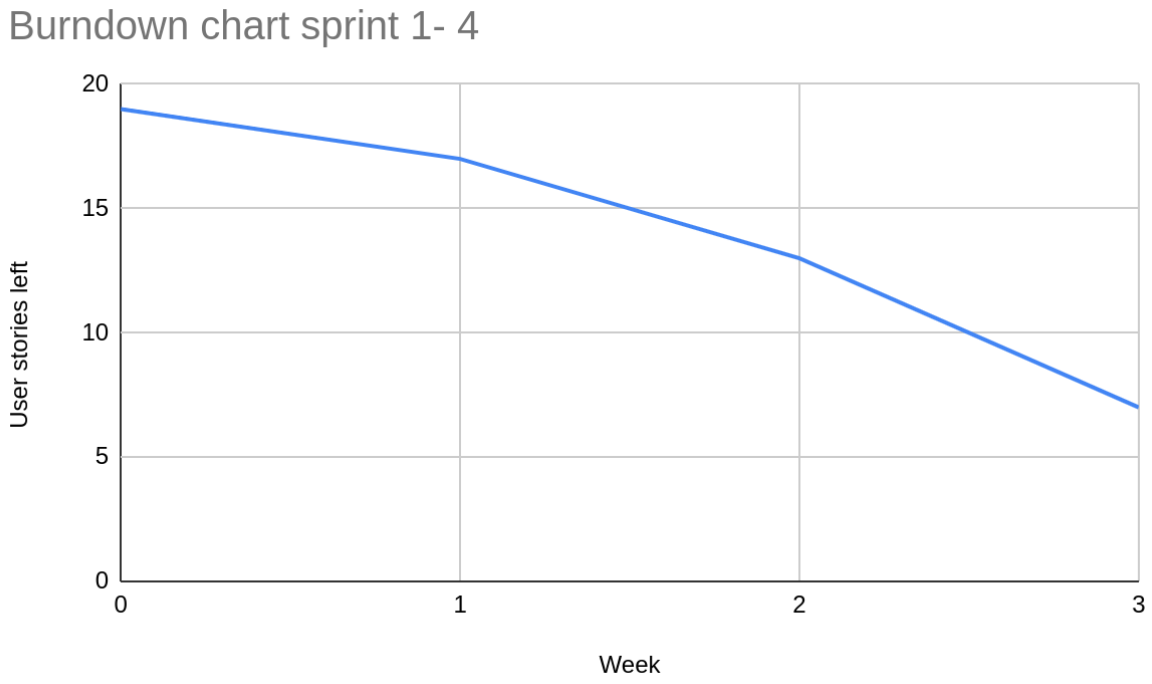
The average time: **7.** (1-10) (10 being 20 hours *and more* of efficient work—the % lowered based on the work time available in the week — with regards to holidays or sickness)

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)

// (We changed what was before “progress measured” to Velocity after feedback from the supervisor.)

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

Burndown chart:



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)

The social contract has remained unchanged still. We want to make sure the social contract is both relevant and fresh in everyone's minds. To do this we will have a mini-meeting inside our next monday meeting to look over and possibly revise our social contract (despite not finding any spontaneous real errors in it yet) at the sort-of mid-way point of the project.

- 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)

15-19 hours ≈ average 16 hours

Our deliveries were good, by our own metrics, and our hours are on the up :)

We would like to spend more hours--if possible--to get even closer to the 20 hour weekly goal. To reach this we will try and make sure there are tasks available after one's user story is done and/or planning the user stories better.

Design decisions and product structure

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

This week we have decided on APIs for both the exercises and the recipes. The parameters taken into consideration when choosing the API for exercises was that it included GIFs, body parts and what equipment was used. This API has then been implemented and supports our customer value. In the coming sprint, we want to make sure that the way the APIs are implemented creates maximum value for the PO. This will be achieved by keeping good communication with the PO.

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

Earlier sprints, there has not been much technical documentation. But this sprint we started creating some use case diagrams to get a better understanding of how users might interact with the application. In the future we might also start using more comments in the code as technical documentation as the project and the complexity of it grows. If for example login is implemented there should be a diagram to see how and where the login is being handled and authenticated, so that one can easily understand all of its components and see possible attack surfaces. To get to this level of documentation we need to continuously create good technical documentation for whatever task we're doing.

- 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. When the complexity and functionality of the project grows we're gonna need more documentation such as use case diagrams, database diagrams and more to ensure it is easy to understand and take in. To get to that level of documentation we need to make sure everything done is documented throughout the project.

- 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

As we have decided on alternating the role of scrum master, this week our new scrum master was Torbjörn. This week one user story was done in pair; Bilal and Ali. Unfortunately the commit was not co-authored but this will be done in the future.

- 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?)

During the sprint review the PO was present. Overall, the PO was happy with the progress. Did have some input regarding design. PO would like us to prioritise a login function and to prioritise the responsiveness of the design. To reach this goal given to us from the PO we will work towards it by choosing and building such user stories, and prioritising them.

- 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)

Learning by doing is the best practice when it comes to learning how to code. So for exempel to learn how to fetch data from an API, it's not enough to just watch videos on youtube. One should try it by him/her-self and try to fix the bugs that show up.

Ask the group members when an issue occurs and no solution is found. It's not a best practice to spend the whole day to find a solution that probably one member can solve in 5 minutes. It's good to search for a solution on the internet but it's not good to overdo it .

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

There was no scrum relevant literature or guest lectures this sprint.

// (and as we have as of yet commented on this task once. We would like to have managed to successfully reflect on this task more. To reach this goal we will ask the supervisor on how to accomplish and reflect on this task!)