### **Customer Value and Scope**

* the chosen scope of the application under development including the priority of features and for whom you are creating value
  + We continue to create value for the customer Made By Waves. The scope of the application continues to be basically the same as the first week. We have prioritized the feature of toggling between weeks and handling crucial data in backend. This sprint has not created that much direct value for the customer but enables handling data when developing more features in the app.
* 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)
  + This week goals were connected to adding intractability to our software. As well as connecting the front end to the backend.
  + The main learning outcome for the group is to get better at webdevelopement as well as working with a customer and listening to his demands
  + We did not succeed completely with adding all the interactabiity as we wanted. Same thing with connecting the back end. It is almost finished but not completely. The reason for not reaching all our set goals were mostly because we had limited time during the sprint. This were because many members had other school things and travel that took time.
  + Next week more time will be spent on the sprint to work towards having sprints that actually fit more content and are not too short. This will be done by accepting work on weekends. This will hopefully make our teamwork better since individuals have more options to plan their own development time.
* 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
  + More structured this week. This is something we want to continue upon in the following weeks.
  + We tried to use more of a “slicing the cake” approach, in the next sprint we may improve upon this by trying to separate backend and frontend concerns.
  + We have tried to generalize user stories to be able to have different milestones of when it is seen as done.
* the three KPIs you use for monitoring your progress and how you use them to improve your process
  + Our first KPI is velocity, which means the amount of “points” we finished each week. We determine points for a user story together during our start up meeting each sprint. This week our velocity was 8 points, however we said that our velocity should be 12 points during the startup meeting. This means that we set up a to big sprint velocity this week and has to do the remaining points next sprint.
  + Our second KPI is costumer s urvey and since we scored 18 out of 18 points last week, we had to talk to our PO and evaluate the scoring system to give us more input. We said that the PO should be more critical to our work and give us more response. This week we got full score on the survey once again, however we got a constructive comment this time.
  + Our last KPI is code coverage, this is all the code that has been tested and therefore completed. This week this is still 0, since no code is actually tested. However, this is something we will start working with next week as we have come far enough with the coding to run tests.

### **Social Contract and Effort**

* your [social contract (Links to an external site.)](https://www.scrum.org/resources/blog/how-make-social-contract-and-build-better-teams), 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 not yet been changed but will be updated next week due to people wanting to work on weekends. Another thing might be to add more detailed info about on how oversleeping relates to fika punishments.
    - Right now we have no standardization for when we merge into master branch. Later we want to have a standard for when and how to do merging. Therefore we will add a part saying: When one is done with an acceptance criteria a merge to main branch is done.
* 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)
  + This week time has been a scarce resource. This have affected what we have been able to deliver. Our goal last week was to increase our velocity. However we did not because of this.
  + We have still managed to be more effective with our time in the frontend group since everyone has been working on their own instead of together.

### **Design decisions and product structure**

* how your design decisions (e.g., choice of APIs, architecture patterns, behaviour) support customer value
  + Started using flexbox which gives a dynamic UI-experience to the user. We wish to uphold this to ensure that future work isn’t too restricted.
* which technical documentation you use and why (e.g. use cases, interaction diagrams, class diagrams, domain models or component diagrams, text documents)
  + Current situation: We are using use cases and clear commit messages to get a good overview of the current situation
  + Where we want to be: The frontend team wants to have a better overview of how to use the backend API
  + How we will get there: Therefore we want to create a text document for API-specification to make the use of backend API easier and decreasing the production time until we have a interactive product.
* how you use and update your documentation throughout the sprints
  + We try to work with commenting in especially in backend.
  + By coding and commenting the code, especially in the backend a documentation is put that is easily controllable and readable as the team members code. This helps with understanding the code since not everyone is coding together. Another thing would be good merge comments.
* how you ensure code quality and enforce coding standards
  + We make sure everyone know the definition difference between fish and fishes
  + By following common coding standards the code gets more unified and easy to read for everyone. Merge requests are also one thing that helps ensure coding quality because the code gets reviewed before merging.
  + Merge requests
  + We started making feature branches in frontend

### **Application of Scrum**

* the roles you have used within the team and their impact on your work
  + Where we are now: Right now we have a scrum master, front end team and backend team
  + Where we want to be: We would like to have a dedicated PO in the team who has contact with the customer.
  + How we will get there: We will take a decision at the next sprint startup and set an official PO
* the agile practices you have used and their impact on your work
  + Sprint, scrum board, sprint start up, sprint review
  + Where we are: We are using sprint, scrum board, sprint startup and sprint review
  + Where we want to be: We would like to integrate a reward to motivate the team to work hard in the end of each sprint.
  + How we will get there: We will discuss what kind of rewards could be possible and work well with most of the team working remote. This could then be implemented into the team contract.
* 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?)
  + Our PO and Cusomter uses a survey to communicate what value we create for him each week. So far his feed back has not lead to reprioritization of scope or userstories. He expressed that he likes the way our project is moving and therefore can keep moving in the same direction. However he expressed that we need to focus a bit more on design. To make sure we develop a design that he likes he will deliver a mockup that suits him. From that we will later try to develop the future design.
* 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)
  + In frontend we have started constructing more components for the application which taught us to better make use of version control by dividing up the project into feature branches that we are working on. By being stricter with future use of feature branches we will ensure a better separation of concern.
  + The backend team has been using Postman to test API-endpoint functionality. In the future we wish to be able to investigate and test more complex situations.