Team reflection week 7 - RCA CT-100

Customer Value and Scope

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

During our latest meeting with our stakeholder Linda, we were discussing the possibility of adding mobile support for the web application “läskuppen”. Although this is something that we would like to do and would add value for the stakeholder, it most likely won’t fit in the remaining time frame of the project. We also talked about broadening the scope by adding tests. This most likely will be done with a framework, and will add value to the stakeholder as the code and final product will be more robust and free of unwanted bugs.

- **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**

We started logging our hours spent on the project connected to a specific task or user story. We did this to ensure that our estimations on the trello board were somewhat correct and if not change the way we estimate and do it better next sprint. This would in the end result give us the “tool” to deliver the maximum amount of created value to our stakeholder, because we can finish all the tasks we take on. We’ve also changed the way we work with the trello board using more columns and always assigning tasks to individuals. This makes it easier to directly see who is working on what and no misunderstandings and people working on the same thing not knowing of each other. Linda, our stakeholder, is also very important to all of these parts, she tells us every meeting what we should prioritize and that is crucial to the value we can bring her.

Design decisions and product structure

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

Previously, we asked questions more related to functionality and features of the site. Because this was seen as the most valuable aspect in order to create value for the customer. Now however, when most of the primary functions have been

implemented we are looking more towards the design aspect. This means that we ask questions about how she wants the page to look, what colors it should have and overall design. She has put much of the design decisions in our hands and gives us much freedom.

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

Right now we don't have that much technical documentation about the project.

We want to have a good technical documentation so that everyone can understand and explore our code.

We plan on updating our current document with more information about the project so that it is easier to follow our progression by making a complete readme-file in the repo.

- **how you use and update your documentation throughout the sprints**

As previously mentioned, we have meetings three times per week where we discuss our progress and path forward. Here we take notes and upload these documents to GitHub. Major decisions are discussed and documented during our meetings.

- **how you ensure code quality and enforce coding standards**

Right now we are focusing this current sprint on code quality and styling. We have implemented close to all of the customers requests about functionality and now we want to clean up the code and also make it look nice. We have structured most of the backend and ensured it's more easy to understand and maintain. We are discussing more about design decisions and how to work with components.

We are still using Git Hub for pull requests when merging and different branches for each sprint.

We want to make automated tests but we are not there yet.

We will get there by looking up information about a framework that is used for testing in React and discuss if we have the time and knowledge to implement it.

Application of Scrum

- **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)**

We use the official documentation and online articles and videos to solve problems and learn new things. We also try to learn from each other as we all have different knowledge about different tools and technologies. We develop these skills by working with the technologies and solving problems using the things mentioned above.