

# Team reflection week 7

## Customer Value and Scope

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

This week we have had a lot of focus on prioritising. We have done an overview in the result page, which was highly prioritized by our stakeholder. We have also taken a big step in the direction to be able to load in data and make it possible for our stakeholder to change text in their evaluations. Json has been implemented for this purpose. Our stakeholder has talked much about how it would be nice to be able to change the evaluations, and therefore was this implementation a big step in this direction. A lot of work on UI has been done, and some fixes that our stakeholders wanted, also very good priority and much value for them! We have also explored solutions for mobile compatibility. We await the response from the stakeholders in order to decide upon the alternatives.

- **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)**

We have had a well functioning teamwork and we are starting to feel comfortable with scrum and our roles which improves the success of this project. This week however the merging of our code was a bit tricky and time consuming. We have added code reviews for this sprint which has improved the code quality. Learning outcomes has been positive as a result of switching programming partners and being challenged in various aspects. We have achieved great changes in backend functionality which is satisfying.

- **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.**

The quality of the user stories varied between the groups. However this did not affect the outcomes to our knowledge. We solved this issue through open communication within the team. Last week we said that we would try to become better at commenting in the code. By defining a “code review” we had a structured framework to fall back on which has made the code more understandable in general.

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

The “code review” has also helped to apply the framework of our “definition of done” which in turn has improved the code quality in general. This also provides a higher

value for the stakeholders if they would decide to further develop the code in the future.

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

Did we complete all our planned user stories in time (0-100%)? 100%

1. How satisfied are we with this week's work (0-10)?  $60.5/70 = 86\%$
2. What has my stress level been this week (0-10)?  $20/70 = 29\%$
3. Have I contributed and been listened to in this sprint (0-10)?  $63/70 = 90\%$
4. How happy have I felt in this sprint (0-10)?  $58.5/70 = 84\%$

Last week we had 64% in KPI 3, but this week we are up on 90%! We tried to have smaller programming groups, which we believe helped a lot. (also no exams this week). Almost all other KPIs were improved this week. We hope to continue this path.

## 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)**

No changes made since last week. No conflicts have risen.

- **The time you have spent on the course and how it relates to what you delivered.**

This week all groups felt that time spent was sustainable although it was greater than previous sprints. We feel that we have made a lot this week, the backend works and the UI looks great. We look forward to showing this to Hast on monday.

## Design decisions and product structure

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

Last week we implemented providers architecture patterns for state management. We needed something to manage our states in the application so we read the Flutter documentation and saw that this is the appropriate way to do it. These helped us remove call back functions, helped further improve the backend and reduced coupling in our project structure!

This week we have made a lot of changes to our back end. We have added Json to provide easier insertion of data for our stakeholder. This enables them to maintain

the code easier and creates value for them. It also enables them to create more questionnaires based on this framework. Additionally the JSON implementation makes it easier to export data.

We also precached some images to enable faster load times and update our SDK version so it is null safe.

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

### **How you use and update your documentation throughout the sprints**

Currently we have comments in the code and we use figma to create visual examples of the application.

This week we have provided more comments through our present code and also in our older code to make it more understandable and easier to maintain. We have also added a convention for writing file names which we have followed this week.

We have also started some technical documentation where we describe the design and decisions made while developing the application. Currently no diagrams but we have some images that describe the flow within the application.

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

Currently we enforce comments on public classes and functions, as the app isn't particularly complex we have yet to implement any unit testing. We wish to have tests for the application to be sustainable in the future, so as development continues we will try to test it.

We have added a column on our scrum-board that is called test-run last week, where we place the user stories that are "finished" but haven't been run on another computer and looked through by someone else. We have worked more with this aspect this week and it has improved code quality.

We have started code reviewing to ensure quality and some standards, this has been working so-so as we haven't specified who will do it. The code review document helps when you are writing the code so it is useful.

## **Application of Scrum**

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

We have four roles in our group

Scrum Master - Responsible for keeping the meetings running smooth. The main "talker" when we have meetings with external parties as well for internal meetings.

Product Owner - Develop the user stories together with the group and make sure that the work continues. After the changes made this week the role feels relevant again. The product owner demonstrates the code on Mondays. To have someone responsible for demonstrating the code makes sure that someone takes on responsibility for the demonstration and that value is added each week.

Secretary - Keeps protocol of the meetings.

BSSARE & time keeper - Bullshitstopper makes sure we stay on track and on time. This has helped us greatly in keeping meetings on track. So that everyone can have a role we have added a second BSS role, this works well to make sure everyone is busy.

We want to reach a point where all roles are clear and contribute clearly and we feel like we have done this.

We can improve even further by continuously evaluating what was good and what was redundant in the roles during our sprints.

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

We have used the agile practices “sprints”, “user stories”, among others to work in a way that increments the total work in a moderate way and enables us to show something at the end of each sprint.

We want to reach a point where we have a steady work flow and get a lot of things done.

We can reach this point by writing crisp user-stories that have clear criterias and purpose. Each Monday we talk with our stakeholders about the user stories which enable us to prioritize some of them for the sprint.

The user stories have been increasingly more specific and value generating.

- **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?)**

Yes! We rotate PO every week. As the PO in this project is also a developer most of the feedback comes from the sprint review with the stakeholders. We do however reprioritise the user stories according to their wants and needs.

- **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)**

About half of our project group are not as used to working with the preferred tools for the project, so we split the sprint up in groups of those used to the tools and those that are not so used to them.

We want everyone in the group to feel comfortable with the tools.

We can reach this point by supporting each other with great sources on how things work, or through verbal collaboration.

We will try to split the work within the coding groups better so that all members can contribute equally.

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

We agree with literature that states that crisp user stories really helps the project move along in an apparent way. This brings a lot of joy to the group members.

We use a lot of the Flutter documentation to implement features in our project and to learn more.