

Team reflection

Group Alphaville

2022-05-06

Members present: Edvin Brogeland, David Boman, Valdemar Vålvik, Linus Lundgren, Felix Edholm, Martin Wahrén, Patrik Olsson, Lukas Jigberg

Customer Value and Scope

We do not think there has been any major changes regarding Customer Value and Scope since the last time we discussed it, and therefore we will not discuss it this week.

Social Contract and Effort

Social Contract and Effort will not be discussed this week since we discussed it last week and there has not been any significant changes since then.

Design decisions and product structure

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

This sprint we developed a new design model for the application architecture. We decided to remake it because our previous design didn't make use of android's standard architecture components. Our new design is more specifically adapted to use components made by android studio, which supports customer value because it makes further development and expansion easier (which means we can deliver features/value to the user faster).

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

For our new design model we used UML class diagrams and arrows to represent dependencies between these classes. We used UML because it is easy to translate the created diagram into actual code, and having an overview of the application structure improves the general understanding of how the application functions, which means that each developer can implement their changes with this bigger picture in mind.

- how you use and update your documentation throughout the sprints

These diagrams are not set in stone however, as they can be updated if problems occur during implementation. The end goal is to have a diagram which represents the structure of the finished product. Getting there necessarily means implementing all features described in the diagram, so developing the design is a constantly ongoing process.

- how you ensure code quality and enforce coding standards

Using these diagrams makes sure that each developer knows in what context they are programming in. For example, when creating a new class or feature, the developer can use the diagram to understand the class/feature itself as well as how it relates to classes and features around it. This ensures higher quality code as it will be developed as part of a greater context, rather than having individual parts which may or may not work with each other.

Application of Scrum

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

Martin has continued being the scrum master. In his role he is responsible for planning and leading the meetings. He creates the documents for the meetings that need them as well as an agenda for each meeting. We wish this to continue in the same way as we feel that it adds structure to our meetings and discussions.

Lukas has also continued being the product owner and has continuous contact with our stakeholder making sure that our features are to his liking as well as asking for what brings him (the stakeholder) value. This helps us greatly in our decision making and planning and we wish this continuous contact to continue.

Both Lukas and Martin are also part of the development team, with the rest of the members, where all of us work on user stories, partake in meetings and do our part each sprint to try to deliver a minimal viable product.

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

We have used planning poker to make our estimations. This has been a useful practice, even though the estimations haven't been perfect. Furthermore, we have followed the scrum sprint-structure and divided work over the sprints. This has led to a clear division of work, and eased the weekly planning.

- 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 have our sprint reviews each Friday, an opportunity for us all to give our opinion on the passed sprint. The discussion and review is helpful because it makes the whole team aware of what worked well and what didn't as well as helps us reprioritize some user stories for coming sprints. Depending on what comes up regarding what didn't go so well we definitely change our way of working. For instance, when a user story turned out to be too big and not well estimated we try to make sure that this does not happen again by highlighting this during the planning when another user story looks to become too big.

- 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)

The scrum board has been in use throughout this project. In the beginning this was something that we did not have much experience with but throughout the sprints, through the experience of using it, we have learned of the importance of the scrum board and how it helps our work. We have also developed the scrumboard and added columns over time.

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

We have tried to follow the way of working described by Håkan during his lectures. Much like the literature describes, we have tried to utilize the “slicing the cake”-way of making user stories, in order to make sure every user story provides value to the end user. We have also tried to work towards making a “Minimum Shippable Product” in the sprints, to prevent the risk of not having anything to show to the stakeholder.