

Team reflection Sprint 5

U2

2022-05-16

Customer Value and Scope

- 1.3 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 have been writing our user stories pretty vertical. We involve both frontend, backend and the connection between the components in one user story. We write some frontend user stories in the way that they are not vertical, but that is because they are really thin and independent. Overall, we create our user stories to be independent.

This sprint we created our first DEV story, that involved discussing potential improvements of the categorization so that we can provide a more accurate categorization for the stakeholder.

- Our goal is to have user stories that are vertical and as small as possible. In order for this to work, most user stories need to target increments in both front- and backend. All user stories should be following the standard pattern and be as separate as possible, they should contain concise acceptance criteria and a initial task breakdown. This is in order for us to be able to get a good sense of what the user story actually intends and how we implement it. Consequentially, we are able to make more accurate estimation of the cost of the user story. Having well defined acceptance criteria, we reduce the risks of misunderstanding the purpose and scope of a user story so that no more nor no less than intended is delivered. Just the user story definition itself could be interpreted in multiple ways.

Regarding the DEV story, we discussed how to create value for the stakeholder and we would probably do that more often.

- Regarding how to improve estimation of user stories, a task breakdown could be helpful as a guide, at least during the initial work on the User Story.

We can create a DEV story in the upcoming sprint if it is necessary. We should, even if we don't have a DEV story, always have in mind the value for the stakeholder.

- 1.5 The three KPIs you use for monitoring your progress and how you use them to improve your process
 - We have used a burn down chart, a psycho-social scale and a code-quality scale as KPIs. We don't reflect over, or discuss, the KPIs in the team.
 - We want the usage of the KPIs to be more systematic, and that all KPIs actually are used during the sprint. We want to learn from our KPIs in order to improve our work and ponder the results. Also we would like to use the KPIs more correctly.
 - We got some feedback on the burn down charts this sprint and that we weren't using them correctly and therefore aren't gaining any knowledge from them. Consequently we are going to read up on burn down charts and try to do them correctly next sprint.

Design decisions and product structure

- 3.2 Which technical documentation you use and why (e.g. use cases, interaction diagrams, class diagrams, domain models or component diagrams, text documents)
 - We comment the code where it's relevant to help readers understand the code more easily.
 - We write comments when we make pull requests, to be able to look backwards what has been done.
 - We have written a backend- and frontend guide so new developers can get an overview of the project and how it's structured.
 - We started creating an UML-diagram for the project, but we decided to drop it because the project is quite small and we didn't find it valuable.
 - The ideal situation would be to have very comprehensive technical documentation to make it easy to understand the project. However in relation to the scope of the project, we feel that the technical documentation we currently have is sufficient. More detailed documentation will just consume time which otherwise could be spent on delivering value to the stakeholder.
 - Since we think the current technical documentation is sufficient, we will continue with the current approach.
- 3.4 How you ensure code quality and enforce coding standards

- We are currently ensuring the code quality and coding standards during the peer code review. Team members go through the code in the pull request to make sure that the code meets our standards. We still don't have a written definition of the code standard, but the team have a good understanding of what it is.
- We could have a clearly defined coding standard that ensures that the produced code follows the same conventions. But we have discussed the code standards and we think the process for ensuring code quality is working good since this is a relative small project.
- Since we think it is working good we will continue to do this to ensure code quality.

Application of Scrum

- 4.4 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 teams best practices for learning new tools and technologies are by communicating actively. If there are any questions regarding how a tool/technology is functioning. The group works together to solve it. Most of the issues have been solved by searching for answers on the internet and sharing the knowledge with the group on the meetings/workshops.
 - The goal is to be able to understand the new tools and technologies easily by searching for the unknown online and otherwise discussing the issues with the group.
 - The team thinks that there are no problems learning and using new tools and technologies now that the team is heading into the final sprint.