Team reflection Sprint 2

U2

2022-04-22

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
  - This sprint we tried to create thinner vertical user stories. We divided the user stories 1.6 (Connect backend(1.5) and frontend(1.4)), 1.4 (List 10 events), 1.5 (Return 10 events from Tickster API) into three different user stories.

    More specific acceptance criteria is needed. Currently the acceptance criteria is too loosely defined, which makes it difficult to know when the criteria is actually met.

    Task breakdown hasn't been done, which needs to be addressed the next sprint.

    Regarding our effort estimation, we had velocity at 39. We overestimated our velocity a bit this sprint. We estimated the cost of each user story with help of the Fibonacci sequence.
  - We want to have one specific user story for backend, frontend and the connecting instead of three different. This is because they become dependent on each other which is not good.

    Our goal is to have detailed acceptance criteria for every user story so that the team members know exactly what needs to be done and how it should work.

    We need to be better to estimate the velocity of each sprint. The cost of each user story should be suitable for the velocity and the task itself.
  - We plan to discuss how to create better vertical user stories with our TA in the sprint planning in the beginning of the next sprint. We

will look at the INVEST-criteria and make sure that our user stories is Independent, Negotiable, Valuable, Estimatable, Small, and Testable.

We will continue to use the Fibonacci sequence to estimate the cost of user stories, this way worked well for us. Regarding the velocity, we will try to not overestimate and consult with our TA.

- 1.5 The three KPIs you use for monitoring your progress and how you use them to improve your process
  - We use velocity/work completed, burn down chart, code quality and psycho-social level as KPIs. Velocity/work completed is used to look back on what was done in each sprint in relation to committed work. The burn down chart is meant to be used as a tool to monitor the progress during a sprint. As of now this haven't been implemented yet. Code quality was first used this sprint and is a subjective measure of team members opinion on the quality of the produced code. Finally the psycho-social level is used for monitoring the well-being of the team members.

    There hasn't been a systematic use of the KPIs by the team during the previous sprints. This can therefore be improved for the next sprint.
  - We want the usage of the KPIs to be more systematic, and that all KPIs actually are used during the sprint.
  - We have committed to clarify the role of the Scrum master by writing a guide for the team. With this guide the idea is that it becomes clearly defined that the Scrum master is responsible that the KPIs are filled in by the team members.

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)
  - So far, we have spent no time on technical documentation since we are just getting started. This sprint's work has been much about figuring out the frameworks and libraries we will be using. Thus we did not really get to a point where things worked well enough for there to be a point of documenting them until the very end of the sprint.

    The front-end part of the project have not been documented yet. The components have no documentation and it is difficult to understand what responsibility each component has. Good documentation is important, not only for us to understand but also for the client.

  - The ideal scenario is that all code that requires special knowledge of a certain library (such as org.json or jsonpath) should be documented so that a person that is new to it can understand what is happening. We would also like to have a documented project structure for the front- and backend respectively so that all team members have a understanding of where to find what in the project and where new functionalities should be added. Lastly, the team believes that it would be valuable to create a class diagram for the backend to make it easy to evaluate necessary and unnecessary dependencies. By making an effort to document well from early on in the project, we could probable reduce the amount of time a member spend asking for help from the author.

    We would like to have more documentation for the front-end. We should explain each component's responsibility.
  - Since we are still in the beginning of the project, the team thinks it would be a good idea to start making documentation a continuous part of our work now in order to save time as other team members may need to be able to understand how the program works. We think that a good start would be to write a document for the front- and backend respectively which explains how their file structure looks like and what code standards there are. We should also document the code that was written during sprint 2 since it demonstrates how things such as fetching the API and filtering it will work. Due to the simplicity of these methods, they could be a good source of inspiration as new functionality is developed. Lastly, we will add to our definition of done that new code in the backend should be included in the class diagram and any code that may be difficult to understand should be commented.

    To improve our documentation, we need to actually write it. So, we will have in mind to work on the documentation the upcoming sprint.

- 3.4 How you ensure code quality and enforce coding standards
  - We currently don't have any coding standard and the process to ensure code quality is still not clearly defined by the team.
  - We want to have clearly defined coding standards for both frontend and backend that ensures that the produced code follows the same conventions. We also want a clearly defined process for ensuring code quality.
  - We have committed to create a coding standard and to define the process for ensuring code quality during next sprint. By doing this and ensuring that the DoD is aligned with those, we can be more certain that the code we release are meeting our standards.

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)
  - Right now we have done the research on react, typescrigt, spring boot and so on, so that the team could get started with the project. Our best practices for learning is looking at tutorials and discussing amongs the team. We have also done a lot of "learn by doing" where the team has tested a solution, depending on the results we have learned something new or need to do more research.
  - The goal is to have good knowledge about the new tools and technologies where "simple" code is not a obstacle.
  - To achieve this, the team will keep doing research and discuss solutions where we will get better knowledge throughout the project.