Team Reflection: Week 2

Team Ginger

Customer Value and Scope

- *The chosen scope of the application under development including the priority of features and for whom you are creating value*
  Keeping a clean environment around us and sustainable society is an important part of human life. With our application the individuals are able to contribute to a clean environment and at the same time inspire others by publishing information on the social media and at the same time inform the different authorities if they need to take any actions. Developing something of value for the environment is the driver for our team. We have decided to prioritize the following overarching goals for application features:

  1. The user should be able to register and keep track of the garbage they have collected.
  2. The user should be able to receive some form of reward for collecting garbage.
  3. The user should be able to show off their garbage they have collected.

- *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)*
  The agile way of working and thinking is very important for all of us. We would like to fully immerse ourselves in each step of the project life cycle and how to break down different complex problems or projects into smaller pieces to be solved step by step. Understanding how different pieces are connected with each other and organizing all dependencies that need to be taken into consideration requires extensive teamwork and contribution from all team members.
  Throughout the week we have been working very well together because everyone puts in about the same time and effort to further develop this project. We learn more by discussing together even though everything is new and we always try to help each other if we have any questions or good suggestions. We support each other to learn Scrum and we strive for working more together and still get a result as if it would come from only one person. It was a great practice to do the Mona Lisa workshop because we could visually see how everyone works differently. Essentially, we would like frictionless and efficient communication and collaboration. How we achieve this is by working together more and learning how every individual works and then combining the work so it blends nicely together.

- *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
How we started writing user stories was writing down the features of the application and then listed them after the priority of what was most important. For the User Stories we used the pattern: “As a <role>, I need <requirement or feature> so that <goal / value / reason>” and added some Acceptance Criteria for each User Story. By writing our user stories it gave us an overview of what kind of application we were developing. We have not yet broken down the user stories into tasks nor have we made an effort estimation because we are not entirely sure on how to do it. This is something that we would like to discuss with our supervisor. We would like to learn more about how to break down user stories into tasks and how we do this is in practice. It is hard to estimate the time effort for each task, because most of us have not worked with this before. But this is also something that comes with experience and we will update and adjust the estimation over time.

- *The three KPIs you use for monitoring your progress and how you use them to improve your process*
  We have chosen the three following KPIs. As we have not completed the first sprint yet we have not used them yet.
  - Target / Actual Time per Sprint
  - Planned features per sprint / developed features per sprint
  - Satisfaction level with project progress per team member 1-5
    (1-under performing, 2-partly performing, 3-meets expectation, 4-exceeds expectation, 5-outstanding)

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)*
  The social contract has been agreed between the team members and we are following the agreement so far.

- *The time you have spent on the course and how it relates to what you delivered (so keep track of your hours so you can describe the current situation)*
  All team members keep track of their own time which will be consolidated when we have started the implementation of the project. The tool we use for the activity planning (Trello) keeps track of the meetings/activities spent during a certain week.

  During this week most of our time has been used for having meetings in which we have decided upon the initial iteration of the project and its accompanying

documentation. Documents designed to aid the meetings and how we should collaborate have been created. We have also made sure all team members' development environment is up and running. As future sprints will look vastly different there is not much to be improved at for the future. But if we were to do this all over again maybe we would have a set of guide-documents to make it run smoother and take less time. We will get better and better with more practice.

Design decisions and product structure

- *How your design decisions (e.g., choice of APIs, architecture patterns, behaviour) support customer value*
  We have created the document "Workflow Proposals" which serves as a template for pitching new workflows or changes to the team's workflow. The proposals will be reviewed at the Weekly Review-meeting and the team gets to decide whether to implement the ideas or not. This will improve the quality of the meetings as they will run smoother and there will be more time for discussions about the application itself.

  We decided to use React Native as our frontend framework, as it works for both iOS and Android to avoid complexity from having parallel code bases. By not having to write the same code for two different devices we will save both time and effort which will result in the customer receiving the application faster and we will have more time to focus on implementing additional features which in turn will improve the overall experience with the app.

  Our goal is to have our code be easily comprehensible and written in such a way that it will be easy to maintain. This will also make the addition of new features run smoother.

- *Which technical documentation you use and why (e.g. use cases, interaction diagrams, class diagrams, domain models or component diagrams, text documents)*
  We have written a Project Specification document describing the overall project and User Stories including the Acceptance Criteria. As the project evolves we will need to include further documentation. For example, what practices to follow with regard to git branching, code style, code architecture, what design system to use and what React Native components to use for certain cases and how. If we do not document some of these practices the team might struggle to stay synchronized and the code base becomes fractured. This could introduce friction in collaboration and communication, which we would like to avoid.

- *How you use and update your documentation throughout the sprints*
  As we have not completed a sprint yet we have no "real" example. But we have decided that if we, at any time during the development process, discover the need

for further documentation, either in the form of updating an old one or creating an entirely new one it will be sorted during the next meeting.

- *How you ensure code quality and enforce coding standards*
  We have two parts to ensure the quality of the code, defining Acceptance criteria and the DoD we will cover the functional aspects of the code while code execution will ensure that we have a well working application. As one of the DoD we will include the code style criteria. We also require testing and code reviews for each user-story before it can be considered done.
  When we start coding next sprint we will try this out and see if there are any difficulties or issues around our chosen methods that need working out. If we do discover some we will discuss it and together decide upon a solution.

Application of Scrum

- *The agile practices you have used and their impact on your work*
  We have had "daily" scrum stand-up meetings where we have shared what we have done and what we are currently doing. This has given the team a greater overall understanding of where in the development process everyone is and how all members' progress will come together in the state of the final product. The daily scrums will need a bit of practice in order for them to flow more naturally, but this will most likely solve itself with time.
  Besides daily scrums we have also created a scrum board which will be used during the sprints. Creating the initial version of it took some time, with creating user stories and such, but we will get faster with each sprint.

- *The sprint review and how it relates to your scope and customer value (Did you have a PO, if yes, who? If not, 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?)*
  As we have not completed a sprint yet we don't know how one would look. This will become clear as we finish our first sprint.

- *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 are using Visual Studio Code as our IDE and git as our version control. We also use Github Project Boards for our Scrum Board. We have read the official documentation and articles of the tools we have used. We have also watched videos and created an instruction-document that supports all team members learning. All documents we are creating will be updated during the whole project life cycle.

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

As we have not had any guest lectures this reflection will only concern the literature. We have read articles as well as videos and information online concerning the agile way of working but we have not yet read any substantial amount of the course literature but we have made good use of the lecture slides. We will read more of the course literature in future sprints.