Team Reflection Course Week 6

Customer Value and Scope

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

The chosen scope still seems feasible. We are half a user story away from MVP. The priority until now has been to prioritize anything that is on our way towards the MVP with prioritization given this being the number of dependencies and our collective understanding of value for the customer. We are creating value for primarily the stakeholder given the context that customers interested in environmentally friendliness being happy leads to the stakeholder being happy. If the stakeholder does not share the sentiment that a choice is best for the customer we discuss and present alternatives according to what the stakeholder says and then let the stakeholder choose direction. This method has been working out well and its use should be continued. We will continue receiving input from the stakeholder and keeping in mind customer values as we prioritize features.

- **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 believe that we have found a good way to use git to synchronize our progress. We use branches for each dev/dev team of which we merge our progress either during the week if there are some dependencies, or a final one at the end of the sprint. Our final product is now in sight, we have some things left to fiddle with, but that alongside our generally better and more efficient application of SCRUM has been an important part of our success. If we get a project that we can be proud of that we made together and that follows the criterias, we believe we have made a successful project :)

- **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**

We had learned that we previously did not use the completely correct standard pattern in our user stories. Some did not have the beginning sentence “as a potential user” or something similar. Our breakdowns of the tasks we felt were okay, but some did feel a bit heavy for what it did eventually entail. In the beginning we underestimated some effort estimations, this was corrected in later sprints however. We feel like we have a much better understanding of how this works now and wish to continue along this path and we will learn more as we try more.

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

Each user story had to be accepted by our simulated PO, the criteria of acceptance was that it seemed reasonable and that it completely followed the acceptance criteria written for each story. The value each user story had to the customer was decided for each story when it was created (user story points). Apart from having the input of each developer, an external stakeholder who would give us feedback on what was to be changed was in place as well. We feel like this worked out quite well and we believe it should and would work for any further projects.

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

We have three different KPI’s, the burn up chart, our team reflection that we make every week and the reflection from the stakeholder that we get each week as well. The burn up chart helps us to reflect over how efficient and effective we are each sprint and gives us the opportunity to reflect over the coming sprints. Our team reflections are really important thus it allows us to think and analyze what can be improved for the future sprints. When you spend a lot of hours each week on the same project day in and day out, then sometimes you can’t see the small details. That is where the reflection from the stakeholder comes. He is basically an eye from the outside to us, where he is reflecting and analyzing our decisions.

Application of Scrum

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

  As mentioned earlier in one of the team reflections. We have alway tried to be equal within the group regarding tasks and roles. We believe that this affects us and our work in a positive way thus everyone gets the opportunity to test and learn about the subject that is up to date. For example, coming closer to the deadline, all of us are aware of how to act as a scrummaster and know the importance of it.

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

  All agile practices we used first week are still in use, and working well. Three meetings a week, startup on Monday, “how are we doing” on Wednesday, and closing the sprint on Fridays. This has proved itself to be a working concept. We have added a couple of KPI’s since week 4. Every week we measure our progress with a burn-up chart to get an idea of how much is left. We get feedback from our stakeholder every week. We have also introduced a team retrospective chart where we discuss what made the team glad, sad and mad this sprint.(A)

The process feels good right now, and if ww keep this up we feel the project will be a success (B). Lets keep it up!(A->B)

- **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?)**

Sprint review has been conducted in the same way as was stated in week 4. We still check if the user stories conform to DoD and the acceptance criteria. During the development of the user stories we sometimes find something that is missing from the backlog and is needed for the completion of the MVP for example the design of the list of vehicles. So during sprint review we also check for features that might have been implicit and make them explicit in the backlog so that we can follow the user stories until completion of the product without missing anything. We have also begun to do our sprint retrospective KPI during the sprint review. We have not found any issues in proceeding in this manner during the sprint review and will therefore only observe, communicate and implement solutions to new issues that may arise for this way of working.

- **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)**

Syncing and merging code within the team has become a much smoother process over the weeks, since all members have developed their understanding. Now every week we are able to merge our branches without hiccups. Also a better understanding of how to limit user stories in an effective way, to enable for concurrent work for all team members, and finishing the stories in one sprint. Google is always an effective tool to learn about coding. But it’s hard to google when you don't know what to search for. As the teams understanding of html, C#, visual studio and the ASP.NET framework deepens, finding new and effective strategies becomes easier.

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

As our product becomes more complete we also gain more knowledge to improve upon our planning and allocation of upcoming tasks. We have gained an idea of how to estimate where for example two user stories may be dependent on each other and communicate and find solutions for this during the planning meeting before beginning development. No new partners have been needed to complete the project however free tools not reliant on upkeep have been used to for example design a button using css. What we have observed as another point of failure is the server which the website is running on. Much of the code used in the product uses the server to compute and then return data to the client. While we never thought this was an issue before we have seen a slight decrease in performance as the product grew bigger. After consideration we found that this was strange since our website is quite small and so we found that the slow nature of the website was due to us never having thought of the difference between debug and release modes of the website. Switching to release mode with optimizations enabled our website was faster than when initially started developing and our problems were solved. This was however an issue that we could not have foreseen without knowledge or research of website development before beginning development and shows, as inferred from the guest lecture, how important either underlying knowledge or a research phase would have been before entering the development phase.