Team Reflection Sprint 4 Group 17

Customer Value and Scope

When we were in an early phase, we discussed what we wanted to personally create and achieve, and we were unanimous about delivering something that could be of use even after the project was done. A usual theme with project work is that after the project is done, all work stops, and the project is discarded. And we did not want this to happen again. At the start people were merely focused on learning about Agile processes. The code was a fun ‘side project’ that would help us explore Scrum and other Agile methods. But as the project progressed, some focused shifted to learning about our code. People felt it was rewarding creating and learning about the new technologies that we are using. This was also a way to motivate our use of Scrum since people wanted to work and learn. Because we felt a drive to continue working on the product, we quickly saw flaws in our Scrum workflow and tried to solve them as soon as we could. Our excitement about developing the product resulted in a better way of working with Scrum.

Last week we developed a good structure for the creation of our user stories, but the Sprint Planning were very long and exhausting. This week we focused on streamlining the way we worked during that time. By dividing the group into smaller subgroups, we divided the process, so it became more time efficient since we did not discuss everything with everyone. Instead the subgroups did different aspects relevant to the process and when that was finished, we regrouped and quickly discussed what we had done and if any other members had input. After the Sprint Review we felt that this was a very effective method that delivered the same quality User Stories and Tasks as the old method while taking significantly shorter time.

Our KPIs have remained the same as last week because they fit well and provide quality measurements in our agile processes. Regarding the surveys there have only been slight changes overall, which web believe to be reasonable since the changes made have not been too critical to the processes.

Although, this sprint we can see quite an improvement in velocity. We think this is because now we are up and running and more people can get work done without the help of others, and even if some still require help the progress is much faster now. People did not report a feeling of being more burdened with work which indicates that the choice we made to increase our velocity and rather highball it than lowball it

gave results. Therefore, we will continue to aim for at least the amount of points in sprint 4.

Regarding the burndown we can see that our work is consistent throughout the week. The chart seems to have some small problems now, like ideal burndown and hours worked doesn't start at the same time. We can also see that some tasks were completed immediately. This is due to some debt and having a User Story that was already completed on creation.

Social Contract and Effort

Through some assumptions about the MVP and what we have delivered so far, we can see that we are a little more than halfway there. Considering the setup time and the learning curve of the technologies, we think this is quite good progress.

Design decisions and product structure

We have chosen to use PostgresQL since it is widely used and has good integration support with other libraries. This is good because if the customer wants to continue development in the future there is a good chance that those developers, if not us, are familiar with Postgres. Postgres is also a very tried and tested database that allows for scalability if the customer changes increases the demands on the application in the future. Furthermore, every person in the team were aware and have used Postgres, making the decision easier since we have not received any requests from our external part.

Considering the backend we have chosen to use Node together with GraphQL libraries. There was a discussion about using Rust, a quite difficult language, but this idea was scrapped and we picked NodeJS since web development experience was limited within the team and using JS for both the frontend and backend reduced the strain for newcomers at learning new languages. This increases the productivity for the customer and by using JS we have utilized a very common language that should bring ease for possible future development.

For the frontend we picked React since it considered by many to be the best language for web frontend now and it was in JS. Frontend languages offer different

pros and cons and the decision was purely based on what we thought would be the best supported in the future and what would be relatively easy to develop in.

Our external partner did not have any request regarding the language used since they didn't have great insight in what would be the best choice. This means we had great freedom in picking language that we thought would provide a good development environment for our group and that would result in a codebase that was relatively futureproof and could be expanded upon.

This has not been mentioned in precious reflections but going into the project we made a text document presenting which programs we use, a description of their function with corresponding information regarding implementation and different tooltips. This technical documentation was for unexperienced programmers vital for success since it became a go-to place when questions or confusions about the applications emerged. It became a short-circuit to understand the application as a whole, instead of having to google everything. It also served as a description for the project overall.

In addition to this we constructed a use case in the early stages of the project. The ambition was to update these in the end of each sprint to keep track on the progress. However, this has been forgotten and not been updated at all. On the other hand, looking back at it we do not feel as it gave meaningful insights, and therefore our time and effort could maybe have been better focused elsewhere.

Application of Scrum

We are continuing with Trello as our Scrum board since it has proven effective through a trial and error strategy. We have tested different practices with different processes and have now found a standardized way of working. Every member of the group is well invested in how it works and therefore we will continue this practice.

Regarding version control we use github and will continue doing so. Our only problem so far has been that people do not take time and review pull requests, which people must do to merge to Master according to our github rules. We have decided to solve this by implementing a rule that the first approximately 15 minutes of every planned coding session you must first look through open pull request and merge them if no problems appear. Late reviews can lead to process debt if they do not get merged during the entire sprint since the task transfers to the next sprint and can also lead to extra work when someone must resolve merge conflicts.

We have developed our way of having the daily standups which differs from the literature and how they mean it is suggested to be performed. We and many others believe that a daily standup is a great thing to get the meeting started and for the wellbeing of the team, since you get an opportunity to reflect on how you feel and make others aware too. If you are having the worst day ever maybe a situation can be handled differently. However, this is what we find of utmost importance regarding the standup, the well-being part. Other talkatives, such as what you have worked on this week and what has been done, we found not as suitable to our way of working. We work very closely and have a good understanding of the different tasks people work on, so therefore the announcement of what you have performed in the week does not bring much value and we have therefore decided not to discuss that in the standup.