

Team Reflection W8

Customer Value and Scope

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

At the beginning of this sprint a big reprioritization occurred. Since the project is soon to be done the PO made a list of things that needed to be done this sprint. (At the moment we are on track)

- **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)**

When it comes to how well the app seems to turn out we have been able to implement core functions, as well as add some more features which are useful. There are some more features we had planned to make if we had more time, but were deprioritized due to a lack of time to implement it all. We have succeeded in not having something with lower priority being made instead of the things we actually never had the time to make, which is a success in itself.

We have learned a lot about how it is to work with scrum and also learned more about how to develop our application. It has been quite individual in what area a specific person has improved the most, but we have all learnt a lot in total. It will also be a great experience to finish this project and reflect on how we would have done it if we did it again.

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 continued our earlier way of doing task breakdowns by discussing in our group how long things are expected to take. One problem with this approach is, as we mentioned in our previous reflection, that if someone else has worked on it previously they have an authority on the subject since other team members might not be knowledgeable about that specific feature. We largely forgot to do what we said to do last reflection when it comes to getting an opinion on effort before the planning meeting. It is also hard to judge since the group is quite diverse experience-wise and a feature might take less time or more time for certain members of the group. We believe that this is something that will get more accurate and easier with more experience.

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

Like previous weeks we have continuously shown the app to the external stakeholder where she gave us feedback about how well the app meets her needs. This process is not formalised, but we feel like it is enough for our small app. Before a story is done it also has to fulfill every non functional and functional criteria. It is up to the individual developer to make sure that this is done. We don't have any formalised way of doing this and it has caused some user stories to be classified as done without maybe all of the acceptance criterias being met. This has not caused anything other than minor inconveniences but it is something that would be quite problematic if the team was bigger we believe.

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

We are still using the same KPIs, stress is still the main KPI we look at. At the end of every sprint all team members fill in evaluations on a scale from 1 to 10 of how they feel/performed for each KPI, we reflect on the evaluations and keep the result in mind for the upcoming sprint. We can see a clear progress in the results in the KPI from the first week.

There are some improvements that could've been done to our selection of KPIs. For example in performance we're currently filling in hours spent each week and the amount of tasks finished. The hours might not be as interesting to know as we as a team can't do much about how many hours each individual spends. What could be used here instead could be to fill in how many tasks are planned to get done at the start of a sprint, then to see how many of these planned tasks actually got finished. That way we could get a better basis to reflect on why we didn't reach our goals; were the goals set too high, did we not work enough, were there too few members assigned to a specific group of tasks?

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) There is a survey you can use for evaluating how the team is perceiving the process and if it is used by several teams it will also help you to assess if your team is following a general pattern or not.**

We are satisfied with our social contract and have just made some few changes, nothing new this sprint. At this point it is highly unlikely that any significant changes will be made.

- **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)**

The team members that for various reasons did not spend as much time as planned last week made up for that this week. We can see a clear pattern where our meetings and work are more efficient than in the beginning which shows in us having more done in less time. One problem with keeping track of the time spent on the course is that it can be in conflict with the goal of the sprint. For example if a person has spent 20 hours on the course but a lot of things are left in the sprint backlog. It might also discourage efficiency since we are supposed to sit a certain amount of time with the app, even if you have achieved a lot in a little time. Also, it discourages taking a lot of breaks, which most of the time increases productivity, creativity and reduces stress, which is one of our KPI's. This has not been a major problem for us but it is certainly something that needs to be understood when using this kind of metric.

Design decisions and product structure

- **how your design decisions (e.g., choice of APIs, architecture patterns, behaviour) support customer value**

During this sprint we have created a page where the users themselves can customise the notifications of the program. This is mainly for the doctors. This makes sure that the app is going to be usable by more users in our customer segments since they have quite different needs. Examples of these notifications are times for medications and early signs of a crisis.

- **which technical documentation you use and why (e.g. use cases, interaction diagrams, class diagrams, domain models or component diagrams, text documents)**

Same as earlier, we plan to write down some of our verbal agreements into some text documents so we won't forget them. Such as what to test and other agile practices.

- **how you use and update your documentation throughout the sprints**

We have been bad at updating our UML-diagrams, but we plan to update it as soon as we have time for it.

- **how you ensure code quality and enforce coding standards**

With continuous peer reviewing we ensure the code and enforce coding standards. We also help each other much and if there is something one is unsure of an/some other team member(s) look at the code for ensuring everything to be correct and that the code is nicely designed.

Application of Scrum

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

We continue to work after the POs priority list which works very well. The PO has a clear picture of how our stakeholder wants it which helps a lot in the development of the app.

Also the scrum master seems to be more comfortable in the role which helps especially during meetings due to the fact that there is a clear agenda which keeps the meetings efficient.

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

Like last week.

Continuous focus on what features are more important to our stakeholder. Now we've realised that we won't actually have time to implement all potential features we had planned, so it's good that we've kept our priorities straight in that regard.

- **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?)**

During our last sprint we were not able to finish everything that we thought we would finish. Therefore the PO made a list of features that needed to be done this sprint. This is something that has helped the team focus on the most important parts of the program to a greater extent. This has proven to be useful as we now know we won't have time to implement everything, so the features which won't be included are the ones of lowest prioritization.

- **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)**

Some people in the group feel more comfortable with reading documentation now instead of just watching videos on youtube, which can increase learning because it leaves more room for your own creativity and understanding of the program than following a video.

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

As always there hasn't been any lectures during the project. We have mostly read up on different aspects on our own, like what APIs to use and how they work. There's also a lot of useful documentation for Android Studio which is helpful. Some time was spent reading up on acceptance testing to understand it better.