

Team Reflection W7

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 priority of features changed during this sprint after PO had a meeting with the stakeholder. It turned out to be more important with clarifying graphs and a feature for adding the date for when patients medication was changed and that this is displayed in the graphs as well. In the beginning of the sprint some working hours were spent on finding solutions for how to show answers from previous days in the calendar, but this was something that would not create any value according to the stakeholder, so we stopped doing that. In order to create value for the stakeholder, we will now focus on the things that were asked for on this meeting and wait with features that we are not sure about.

The user story about diary entries might be outside the project scope since it is not deemed important by the customer and it will take a fair bit of development time. Our goal is to continue to make these decisions before we start implementing them as mentioned above.

- **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)**

So far, we have learnt how to improve our teamwork by more effective meetings thanks to having an agenda and now being more familiar with how we work and communicate with each other in a better way. We want to improve these skills even more and will do so by continuing to reflect about what is working and not and keep on listening to each other's needs and thoughts.

The ones who had not worked with the programs we are using for coding have learned to work with them and the ones who did not feel very comfortable with coding feel more confident doing so as well. We want to keep on improving our coding skills and keep doing so by coding even more and trying to solve problems outside our comfort zones.

- **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 are better at breaking down user stories in smaller tasks now than in the beginning, but since it is not very thrilling to do it, it is not done too often. We believe that we would do it more often and easily if it became more of a habit.

We have also added important user stories during this sprint and the PO ordered them according to priority after the meeting with our stakeholder. This gave us a more clear idea of how far we have come in the project and which important features that are yet to implement.

We still have not standardized our way of doing effort estimation but we still had a discussion about how long everything was expected to take. We found that this worked well for us when it comes to delegating resources to different areas of development. There have been some mistakes though, such as the notification system, which we deemed would take less time than they will.

Sometimes it can be hard to make the tasks be of similar effort to implement as when there's APIs that are new to us it's hard to tell how easy to use they are beforehand. It might be a good idea for everyone in the team to think about the complexity of the features that are going to be implemented before the planning meeting.

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

Our PO is involved in testing our user stories both directly and indirectly. Indirectly by involving the customer in the testing of major user stories, this by showing our stakeholder a demo of the app and the specific functionality that is tested. In this way we can get continued feedback. Directly by going through the features that the team deem to be done. This is something that has worked well and this manner of testing will continue.

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

The ones we're using have seemed to work fine so far, and we're keeping them in mind when planning the new sprints every week if there's anything we need to change. Stress is still the main one we keep in mind.

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.**

The clarified definition of roles have seemed to work well, and we'll keep them as they are for now.

- **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)**

In the last sprint some group members spent a little less time than the 20 hours per week for various reasons. This was noticed at the sprint review. The plan is at the end of the final sprint to compensate for this by spending a little more time per sprint. This did not seem to affect our productivity much though since the team has generally become more productive.

Design decisions and product structure

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

At the moment the ability for the app to show notifications is being worked on. There are a lot of considerations to be mindful of when implementing it. For example whether the app should be able to run in what is called Doze mode, which is a power saving mode on new Android phones. If this should be the case it will hurt battery life of the phone. When the notification should be triggered is also something to consider. If the trigger is going to be somewhat exact time wise, meaning it triggers for example once a day at 18:00, the phone is going to be running less efficiently since the operating system optimizations where notifications are handled at the same time are not used. We decided that both of these factors are important enough for the customer value to accept the negative effects described above, since one notification is a reminder of medication. In general, like we have done before, we prioritize features over optimisation.

We have realized that the way the quiz was designed and implemented maybe wasn't the best, we have encountered multiple bugs and implementation problems due to this that have slowed our progress. Even though we have been able to handle these problems and the functionality that is essential for our customer value is still there we plan to try to improve the code in future sprints if time allows.

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

Nothing new here since last week. Github wiki for saving and loading functionality is still planned but not done.

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

Nothing changed since last week.

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

We are continuously peer reviewing each other's code in order to ensure code quality and enforce coding standards. At the sprint planning meeting we try to divide new, unreviewed code from last sprint that will be reviewed during the upcoming sprint. This ensures all code ready for being reviewed actually get reviewed, and clarifies what needs to be worked on more.

Application of Scrum

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

Our PO made a priority list for the user stories and tasks which made it clearer on what we had to focus on.

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

We've kept focusing on the user stories and tasks most valuable to the stake holder. There's now a clear priority list for this which makes it easier for everyone to get a good overview of what to start on next in case we get done with something. Since this list got made a day into our sprint it made us realise that some things we started on were lower in priority than we first had thought, we could then shift that focus to be on something of higher priority instead.

- **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?)**

This week our PO had a meeting with our customer, after the meeting our PO reprioritized some of the user stories. It did not really change our DoD but we will be working after the new prioritations in the future and it changed which things we were working on. We also removed some parts of the original project since it did not add value for the customer and the PO made a plan for the rest of the project from the new prioritization-list.

This prioritization list gives us a clear overview over what has been done so far, and what essential functionality we have yet to implement.

- **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)**

Learning from the most experienced members of the group. They have been very helpful and understanding when things have felt difficult. We believe that our way of working, mixing the more experienced students with others in the group is the best practise to learn and we will continue working like this. External documentation has also been heavily used for some parts of the application to learn.

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

We did not have any guest lectures yet. Some implementation decisions were made from what the group could find online regarding best practices in Java. It is our goal to try to do that in the future. A lot of time was spent reading the Android documentation when it came to the notification system. This is because the system is somewhat of a mess with deprecated features and the documentation is really needed to create something of value.