Team Reflection W4

Customer Value and Scope

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

  As we had nothing at the start of the sprint we started with the essential base for building our application on, such as views and data saving, and from there we can build the individual parts in the upcoming sprint. We create value for our external stakeholder and her patients. To prioritise features the product owner decided which features were the most important and created the most value. This is something that worked well and we planned to continue doing.

- **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 want to learn/get better at the Android framework, but also get better at learning software related tools, for example reading documentation, in general. A lot of time this week was spent on getting familiar with the work environment as it is new to most group members, we plan to work together to learn new things.

  Another goal of the group is also to get into the agile mindset and get comfortable with agile processes. This is going to be achieved by continuing to follow the agile practices we at the moment follow but with some improvements when needed.

- **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've started with some main categories of user stories and have started to break them down into smaller one, but this still needs to be worked on. We also need more acceptance criterias. The same goes for tasks. We also need to get better at having everyone involved in marking tasks and moving them to the right lists to make it easier for other group members to get an overview of what other members are working on and what is not currently being worked on. We did a very informal effort estimation of the tasks where we had some discussion about them but it might be a good idea to implement something like planning poker to get a more structured way of doing it.

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

  So far we've not done a lot of work here, we need more of a plan on how it should be done and have a meeting about it so that we can make a joint decision and so everyone is on the same page on how it will be handled in the future.

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

  We have discussed how this can be done, but still lack a clear way of tracking progress aside from task cards in Trello being moved to different lists depending on their status. We have had discussion about what parameters we want to measure, for example well being and productivity but since we have not had our first sprint review yet it has not been a priority. As long as it is done before the review.

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 need to regulate our communication in our social contract more because there has been some confusion related to when meetings are. We are going to have a discussion about giving the scrum master the final say on meeting times, as well as being responsible for making sure the meeting date and time are written down in Slack, and keeping track of which members that can and cannot attend the upcoming meetings.

- **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)**

  Our goal was to spend 20 hours a week on the project. This was largely fulfilled although our productivity could be improved due to some meetings being inefficient time wise and the start up time associated with using new tools. To make sure we have more efficient meetings in the future we will need to have an agenda.

Design decisions and product structure

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

  After a meeting with our stakeholder we got instructions about things to keep in mind about the design, such as not using some specific colours and certain images in the design. We had that in mind while redesigning our mockup. This will be used to create our GUI, and works as a good reference point for everyone to work towards so we can be sure we have the same idea about what the final product will look like.

We tried to do a design that can be easy to navigate, the stakeholder told us that this is very important so we had that in mind while doing the design. To ensure this we have created an app without too many nested pages which will make the app simpler to use.

Our code is architectured with a controller-view pattern. We have an idea of wanting to break it down to a model-view-controller pattern. To find out what is best we plan to search for more information on how to break it down in android studio since it is theoretically possible to use MVC in android but quite complicated. The reason why is that the GUI part of Android is heavily intertwined with the controller part.

At the time of writing we have not chosen an API for graph representation but it is something we want to have done at the beginning of the next sprint because data representation is an essential part of our program. It is also important due to the fact that a lot of implementation design decisions will have to be based on the choice of graphing api. The API should be simple to use and also look graphically pleasing.

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

Our documentation at the moment consists mostly of comments in our implemented classes. It seems to work well but it might be a good idea to have documentation in an external tool or document when the program grows larger. We drew a class diagram at the end of the sprint because it helped us to visualise dependencies and since our previous lack of a class diagram had already caused some trouble concerning how different parts of our program should interact.

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

If something is changed we should of course update the documentation. At the time of writing it is only the comments in code.It is also our goal to keep our UML class diagram updated when new classes are added. To do this we have a shared lucidchart project. The idea behind commentating our code is to make it easy for another one one in the team to use functionality that they themselves have not written. We also use the author tag to make sure if someone does not understand something they can contact the one who wrote the code for help.

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

We ensure code quality in the master branch by working on separate branches in GitHub and push new changes in suitable parts so that we easily go back to a working code if something goes wrong.

All our code should also be peer reviewed to ensure that the code is of reasonable quality. This although something is something we can be more systematic about instead of doing the peer review at the very end of the sprint. It would perhaps be better to do it after every commit instead.

All code we write should be tested, it is in our DoD, either through for example JUNIT or other means if unit testing is not efficient. This ensures that there are fewer bugs but also that the code we write is testable which is not a guarantee for good code, but certainly an indicator of it. This is something we plan to continue doing for every sprint.

Application of Scrum

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

  The beginning of our sprint was quite chaotic which made it quite difficult for the team to adapt into their roles. The roles of scrum master and product owner were also decided quite late. This resulted in some confusion early on which made some of the agile processes such as meetings and creating user stories/breaking them down suffer. This is something that surely will improve with experience and we also plan to make the roles responsibilities clearer in the next sprints.

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

  Our work with Trello has been a part of working agile. By having user stories and breaking them down into smaller, well defined tasks it is easier to divide the workload within the group and make sure that everyone can have a better overview of what everyone else is working on.

  However it feels like the numbering system we've made for tasks are a bit confusing right now, that's something that can be worked on.

  We have had daily scrum meetings to keep the team up to date when it comes to development and other related tasks. It also creates a channel for discussion about design choices. This is something that has worked well and increased our productivity. One way to maximize the usefulness of these meetings is perhaps to have a clear agenda that is followed. During our meetings discussions about things we all group members are not working on has been a bit of a problem since it takes up quite a bit of time. This also something that might be solved by an agenda.

- **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?)**

  While writing this team reflection we have not had any sprint review, because on this week's Monday meeting we decided to have a sprint review every Monday. So our first sprint review will be this Monday the 19th of April.

- **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)**

  For the students from I we got help from our friends from IT for learning Android Studios and thereby could practise on our own to receive knowledge on how to work with it.

  We made a semi-final design of our app in Figma and learned that tool as we were working on it.

  We have watched many tutorials online and read a big part of the documentation on the new tools we are using.We have also actively discussed and asked for help from the team members that had more experience using these tools.

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

  This week we had no guest lectures, so we tried to find literature in other ways, including videos about how Android studios work on youtube and help from the group members that have knowledge in the program.

  Some of the group read a blog post about using MVC in android. It is still up for debate if we will use it since it is quite hard to implement in Android.