

Customer Value and Scope

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

During this week, our scope has been to improve our notifications function for our app, but also providing scheduling functions for the user. A planned implementation from the beginning was to give travel recommendations to the user, but this is still not done and will probably need another sprint to finish.

- 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)

Our main goal is to create user stories of a reasonable size that can be completed each week. Due to our experience both with scrum and with app development our estimates are getting better, which allows us to get closer to completing every user story we set out per week. Sometimes there are unforeseen issues, which result in some user stories not being completely finished. We want to minimize unfinished tasks, and we will do this by being a little bit more pessimistic in our estimates (in other words, assume something goes wrong).

- 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

This week we stayed highly organized and had clear communication about how every chosen user story is progressing. This led to some members getting new tasks at our midweek evaluation. To speed up the progress on more complex user stories. Our effort estimation felt quite accurate this week.

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

We set up acceptance criterias in trello for the user stories. The one/ones working on the user story test themselves if what they're working on passes the criteria. This allows us to collectively agree what a user story should do, making it easier for the one working on it to remember what the goal is. The tests themselves are up to the one working on the user story. The code is then usually tested by other team members who have some time available to double-check that it works on another computer. This speeds up the process, since the one who is writing the code tests

while working on the user story as well. We should however make sure that someone always, rather than usually, double-checks that the user story works as intended. We will do this by telling each other when we feel done with the user story.

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

we visualize how we did during the last sprint and can even see how much we need to accomplish to stay on track. For that we use burn up, burn down and a self designed “tasks per week”

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.

All members worked according to our social contract.

- 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)

We began this week by having a meeting before our supervision, which lasted 3,5 hours. At the meeting, we discussed how any eventual work had been going through the weekend, planning for the upcoming sprint and doing a sprint review for our sprint that we had just finished. In the middle of the week, we had our weekly checkup to see how work is going for everybody in the team. The checkup lasted about an hour. At the end of the week, the team met up to do team reflection, and sprint review of the current sprint. We also went through all the problems that had come up this week. This meeting lasted 5 hours.

Design decisions and product structure

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

We made the choice that our buttons should be user friendly to help new users.

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

We have a UML diagram with some documentation over our classes but otherwise we do not have much technical documentation.

- how you use and update your documentation throughout the sprints

To keep up with our documentation, we try to continuously comment our code as soon as there are methods or some rows of code that are hard to read. This makes it easier for ourselves to keep up with what we have implemented so far. Before we push it to GitHub or merge it together with another branch, we try to be more strict about this.

- how you ensure code quality and enforce coding standards

We only push up things to Main that people see as working functionally when we've shown that it works to others. Our coding standard is mostly that it should function.

Application of Scrum

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

Our scrum master has done their part by booking rooms and communicating with the team making our work smoother. The PO communicates with our stakeholder and this week we reprioritized our user stories and finished the work on changing locations when checking the weather.

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

Our user stories help us divide up the work throughout the week and the meetings work well together to make sure that we are all on the same page. The reflections help us with looking back at what we've done and will eventually help when writing our final report.

- 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?)
- 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)

This Week we collaborated more than the prior weeks so that several people were involved with the same user stories. This way it was easier to learn from each other and get better knowledge about other aspects of the app.

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

This week, we didn't have a guest lecture or any updated literature.