Customer Value and Scope

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

Polishing the code and removing eventual bugs. Ultimately merging everything into main.

- 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 wanted a good working app based on what we had accomplished this far

- 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 did not work on user stories this last print but polished the existing code and functionality for the presentation

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

we simulated on the chosen model.

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

We did use the same KPIs as before

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.

A new one we could add in another project would be to come up with a plan for merging to become more organized. This is the last sprint so we did not add this to the social contract however.

- 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)

Design decisions and product structure

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

No more design decisions this last sprint

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

We completed the Javadoc documentation on our classes and methods.

- how you use and update your documentation throughout the sprints

- how you ensure code quality and enforce coding standards

Application of Scrum

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

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

This week the agile practices did not have a large impact on our work. We did not have any user stories left to work with and there was not much focus on the work but rather the presentation. The stakeholder was unavailable but we still had meetings for the sprint.

- 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?)

Since developing already done, PO wasn’t involved.

- 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)

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

Not very relevant for this sprint specifically, but looking back we can see that estimations did get easier and easier, just like for Matthias.