**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 scope of the application have changed since the customer communicated how he wanted us to prioritize our features differently. This means that we will not include the section for goals and metrics in our mvp. Instead we will focus on including drag and drop as an important feature in the task board. This was something that the customer thougt was very relevant for the user experience. He also wanted us to polish the design och the program to be more minimalistic and clean so that is also a part of ous scope now.
* 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)
  + This has not changed that much. However our stake holder has expressed that he wants us to focus foremostly on a few features and make sure these are running really smoothly. Our goal therefore is now to create a really good software that our stakeholder is pleased with. Overall our pesonal goals is the become better software developers.
* 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 have kept writing useful use rstories mirroring the needs of the PO. This enables us to work forwards a product that will create value for our PO and does not wear off to a faulty application.
  + We have kept the same standard on user stories, putting the POs opinion as the main value adding orientation.
  + Keep using acceptance
  + We have kept using acceptance criteria for the code we write, where this ensures god quality coding with less bugs and the right formation to create value for the PO.
* your acceptance tests, such as how they were performed, with whom, and which value they provided for you and the other stakeholders
  + The backend and frontend test are used for regression prevention.
  + The frontend’s components have gained more tests in the form of acceptance tests.
  + The backend has gotten more feature documentation.
  + Value created is that the development of the application is going more smoothly and safely, also these tests can be used as a documentation complement for how the code is supposed to work and how to interact with the different modules and components. Which ensures correct functionality and may improve future velocity which in turn brings value to both our customer and ourselves.
* the three KPIs you use for monitoring your progress and how you use them to improve your process
* Where we are now: We are using our 3 APIs below to monitor our progress. This week we had the following results.
* Velocity
  + 11
  + Where we are now: We did not finish more than one user story but got most of the remaining ones partially done. So we will mot likely be able to finish ad debug our complete MVP next week.
  + Where we want to be: It would be preferable to plan our sprints better so that we can pick fewer user stories and make sure they are finished by the end of each sprint instead of having a couple of them “half-done”
  + HOW we get there: practice more on giving points to tasks/user stories and being more flexible so that one person can put in extra work to be done on a specific feature to get I finished in the end of a sprint and maybe work less intensively next sprint.
* Customer satisfaction maxed out again..
  + NOW: super happy
  + THEN: more critique
  + HOW: better questions and no friends as customers
* Code Coverage (85% lines)
  + Where we are now: We have a higher code coverage helping us negate regression of our application.
  + Where we want to be: For the next week we want to be in line with the velocity estimation of our burndown chart to finish the MVP. We also want to get feedback on the final product from our customer.
  + How we will get there: We will get there by setting up clear tasks for next weeks sprint, making sure to have a few days for potential bug fixing before the sprint ends.

**Social Contract and Effort**

* 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 have started analyzing time spent in relation to velocity with a couple of excel sheets. We have the time spent in total and by group members from data collected over the past weeks, this is an estimate over how much time and effort each of the project members have worked on the project. We separated time spent in meetings and time spent coding. This separation is quite vague since some coding has been done during meeting times. Still, this gives some overview of how we use our time in the project. Overall, the time spent is similar and our conclusion is that everyone has put time and effort into the project to make it a final product. Some meetings have not been super focused so we could probably lower meeting times. Finally, roughly the same time has been spent on meetings and coding so far.

**Design decisions and product structure**

* The main design decisions for our product has been made with the agile methodology in mind. Developing a feature in every sprint and having that feature as the main focus adding depth and complexity to code when needed for future features.
* Customer communicated that it is important that the design of the product is minimalistic

**Application of Scrum**

* the roles you have used within the team and their impact on your work.
  + Just like previous weeks we have had a scrum master keeping our overall progress tracked and making sure we are keeping pace up.  He not been very present this week so other members have helped with creating meetings and such.
  + The roles regarding frontend and backend Is kept just like before.  This is because we want to not disrupt the workflow by changing the workplaces, i.e. backend spending most time on logic and Java plus SQL while the frontend has spent most time on GUI which is programmed in React.
* the agile practices you have used and their impact on your work
* 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?)
  + One of the group members act like PO just like before. By contact with our customer using zoom we decided to focus even more on refining the software to make it look good and easy to use. This was quite alike what was communicated last week, because of this no major change was made in the writing of user stories.
* 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)
  + MAYBE not this time