Agile software project management  
DAT257

Team MS Stena Nautica

Oscar Thureborn toscar@student.chalmers.se

Max Petersson pemax@student.chalmers.se

David Hermansson davherm@student.chalmers.se

Dara Khadjehnouri dara@student.chalmers.se

Karin Perstorper karperst@student.chalmers.se

Philip Hellberg phihel@student.chalmers.se

Teachers

Jan-Philipp Steghöfer

Håkan Burden

# Customer Value and Scope

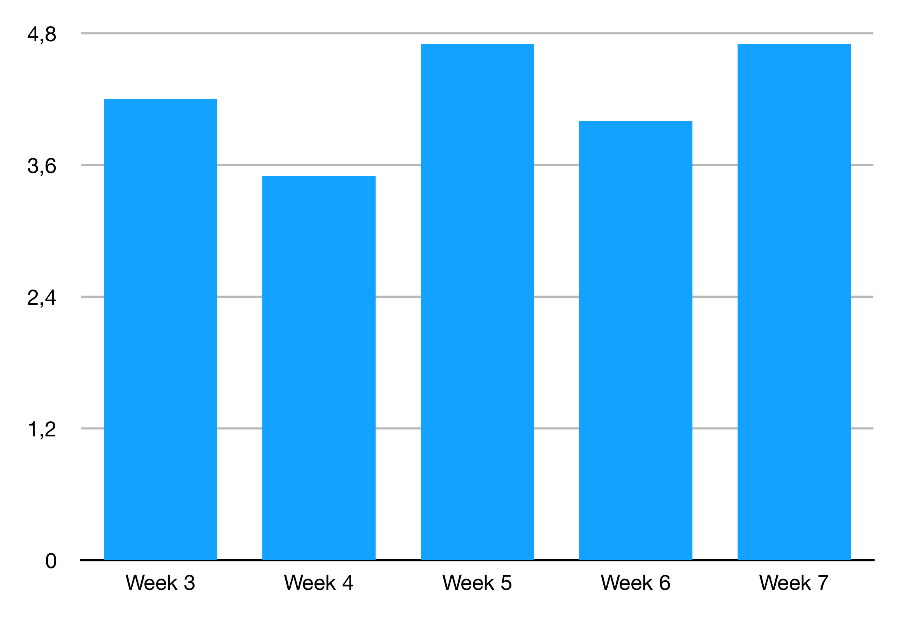
**Present(A):**

Coming up with a suiting product was something that we all felt was quite hard. We realized quite early that it had to be something graspable and reasonably small. However, we left a narrow window for future expansion open, without damaging the core product.

When dividing the product into epics and further into user stories we received great help from our product owners with different priorities. After the first couple of meetings with the product owner we realized that we could save time and increase quality by not writing anything too specific before the meeting but rather having an open mind and adjusting to their desires.

Writing user stories and dividing them into tasks was something that everyone mentioned at least once that they felt was difficult. However, after a few weeks we managed to produce tasks and user stories at a much higher quality and rate.

KPI:

We decided to monitor three specific KPI’s. The first thing we decided to monitor was ESI (Employer Satisfaction Index) to see how the group felt. Everyone was asked to pick a number between 1 and 5 (5 being ”very good” and 1 being ”very poor”) every week with the focus on how they felt in the group. The result of this KPI can be seen in the graph below. 

The second and the third thing we wanted to monitor both fall under the work-flow category. We had Work-In-Progress-limits (WIP) on different parts of the process. The first thing was the main ”In Progress” category and the second one was ”Code Review”. The goal with this was to make sure that we did not have too many things in progress at the same time and also to make sure that code-reviewing was done. We monitored these by having a number in the two different columns on our scrum-board. This changed over time. We felt that this became very handy in the end when we knew that we had to clean up, summarize and prepare for the final delivery. We lowered the limits greatly the last sprint.

**Future(B):**

Be more efficient in the start of the project.

**How will we get there(A->B):**

Meet with the product owner as often and early as possible. Saves a lot of time and steers the project in the right direction. Don’t focus on possible expansions of the product at all in the beginning.

# Social Contract and Scope

**Present(A):**

Our first social contract mainly described when, where, and for how long we would meet up. It also stated how working in groups was encouraged, which led to getting the majority of our work done at planned meetings. The contract was then revised to more specifically declare how each task should be named and branched in git, as well as technicalities regarding languages and compiler version for deliveries.

The contract was only revised in the first two sprints since that’s when all confusing questions were brought up in the group. The contract served perfectly as a group-decided guideline for the remaining sprints, and the group felt that the contract increased efficiency throughout the course even though it took a lot of time to refine.

**Future(B):**

Reduce amount of revisions of social contract, and general time spent on it.

**How will we get there(A->B):**

Be even more specific and cover all possible questions at the beginning. Make sure all people involved know exactly how to work from the start.

# Design decisions and product structure

**Present(A):**

One of the biggest design decisions involved ignoring a big part of our backend, since it was supposed to be based on NeTEx/SIRI data which we would never have access to during this course. We could’ve implemented support for that kind of data, but we decided to use the most basic mockup data we could think of, since the delivered product would’ve used mockup no matter which data structure we supported. This was something that produced equal value but saved an immense amount of time because there wasn’t any need for complex diagrams or models for our backend.

The small amount of documentation we had was updated in each sprint, although mainly to stay relevant and keep up with our coding standards. We ensured code quality per default due to our later use of peer programming.

**Future(B):**

We’d like to know more about where to put our focus regarding customer value, to maximize value per effort.

**How will we get there(A->B):**

Ask our product owner as to whether frontend or backend is favorized in value and simply focus on what is possible in that domain.

# Application of Scrum

**Present(A):**

We have only had one official role throughout the project, the scrum master. The scrum master focused first hand on organization and documentation but also partook in coding sessions. This meant that everyone else could focus more on work tasks.

The group had a scrum board on Trello with a prioritized product backlog, which were updated weekly after our meetings with our product owner. This made it easier to plan our sprints and deliver value to our product owner.  
We weekly updated our velocity depending on our resources available (how many were sick). This helped a lot when we chose how many user stories we should include in the current sprint.  
We tried to make each user story and task as narrow and vertical as possible. This meant that each increment of our product produced value in each layer, and every one of us got a better understanding of the product design as a whole.  
We started of the first sprint by working separately on tasks but migrated towards group-coding sessions with co-programming. This made sure that we got help faster if needed and we could produce work faster.

Each week we met with the product owner and showed the result of our sprint. We took feedback from them on our current progress and also discussed what user stories we should focus on in the upcoming sprint. This meant that we sometimes had to update the priority in our product backlog.  
Each week we also got feedback from the teachers which impacted our way of working. We started to focus more and more on making sure we delivered as much value to the customer each sprint. That way we could get more feedback from them and they were happy to see progress.

As mentioned above we used a scrum board on Trello. Initially we didn’t really understand how to write good Epics and break them down in to good user stories and tasks and where to put them on the scrum board. As the weeks progressed, we got feedback and applied it on our scum board layout and how we designed our user stories. Each of our user stories gradually expand our product and it is clear to the product owner what value it will bring to the product.

**Future(B):**

We want to deliver more value to the customer during a sprint and we want to keep better track of the team’s progress during the sprint.

**How will we get there(A->B):**

We want to increase the sprint time. One week is very short and we found it hard to deliver substantial value in each of the tree layers (UI, business and data) during that time span.  
We would also like to start with daily stand up meetings. This was not possible for us to perform since we could not meet up every day.