Sprint 1

The group started out by defining the intended workflow and the development methods that should be used. Most of us were new to Scrum and much time went into understanding the principles of Scrum and how we should implement it in our team. We also set up our Github and a good base structure for the group to follow. The term “to slice the elephant” was brought up and we decided to apply it to our project. By using this method we could present a fully functioning prototype at the end of every sprint built on backend as well as frontend. All group communication should be performed through Slack.

After a couple of days the group encountered numerous difficulties. Git branches were barely used and most commits were pushed to the master branch. This led to Git conflicts and many bugs with starting the application. Communication between group members was lacking and needed improvement. Bad communication resulted in bad task division and unclear sprint goals. Other than that the group “sliced the elephant” well with a functioning “hello world” web application at the end of the sprint. All teammates were helpful and also used Scrum very effectively.

The development was directed towards giving the product owner a MVP that had both design and functionality. The first task was to display HelloWorld on the web app. It went fairly quick and the development was then directed towards sending an API get request to the PortCDM backend and then display it on the site. After sprint 1 the product on start up would send an API get request to the backend and print it to the console. The intention was to display it but the team did not reach that goal in time. So all in all it was a HelloWorld text on the website, with a table and the correct headers but not with the extracted information.

Sprint 2

In the sprint meeting from Sprint 1 the team used the different decided KPI methods and concluded how the team was doing. A particularly useful KPI was proven to be the retrospective process improvement. From sprint 1 it was obvious that the team lacked in conversation and clear goals everyday. With inspiration from Spotify lecture the team then decided to implement stand-up meetings where everyone in the team answers three short questions about what they have done, what they are going to do and what (if anything) stands in their way of reaching the goal. The team also decided that it was better to meet more often. An effect of this was a more clear communication, everyone had something to do and the git seemed to be working better.

The product during sprint 2 was developed towards the goal of printing the api get response to the web-app. It was proven more difficult than expected but was during this sprint accomplished which was a great relief. The team still had problems with implementing CSS and making continuous API calls.

Sprint 3

The team faced difficulties with the web-app and the development did not go as fluent as wanted to. This was most likely due to knowledge of how the programming language worked. The problems included implementing css, making post-api calls and re-render the web app. It was then concluded that the problems was linked to the fact that the app used server-side rendering.

The product which was earlier based on Node, VueJS and an express router was changed to use Node, webpack, vue and a vue-router. This made the app work with client-side rendering and everything started to work as expected. This caused the development to sky-rocket and the team had new energy in form of developing features for the app. The slicing of the cake also became more eminent and it was obvious that the features the product consisted of/were developed had a clear design and functionality. During sprint 3, functionality for posting and changing the states was developed. This was done by receiving input from the user in the webapp, parsing the information to the correct XML-format and then sending the XML input to the correct API-post request. Other development was done towards displaying the correct ID of a tugboat, getting the table to look as intended and continuously make an API-call so that the table updates every so often.

As for the upcoming weeks of the project we hope to be able to implement the rest of the product and have a fully functioning web app to show. We believe that during the upcoming weeks we will learn atleast as much as we have up until now.