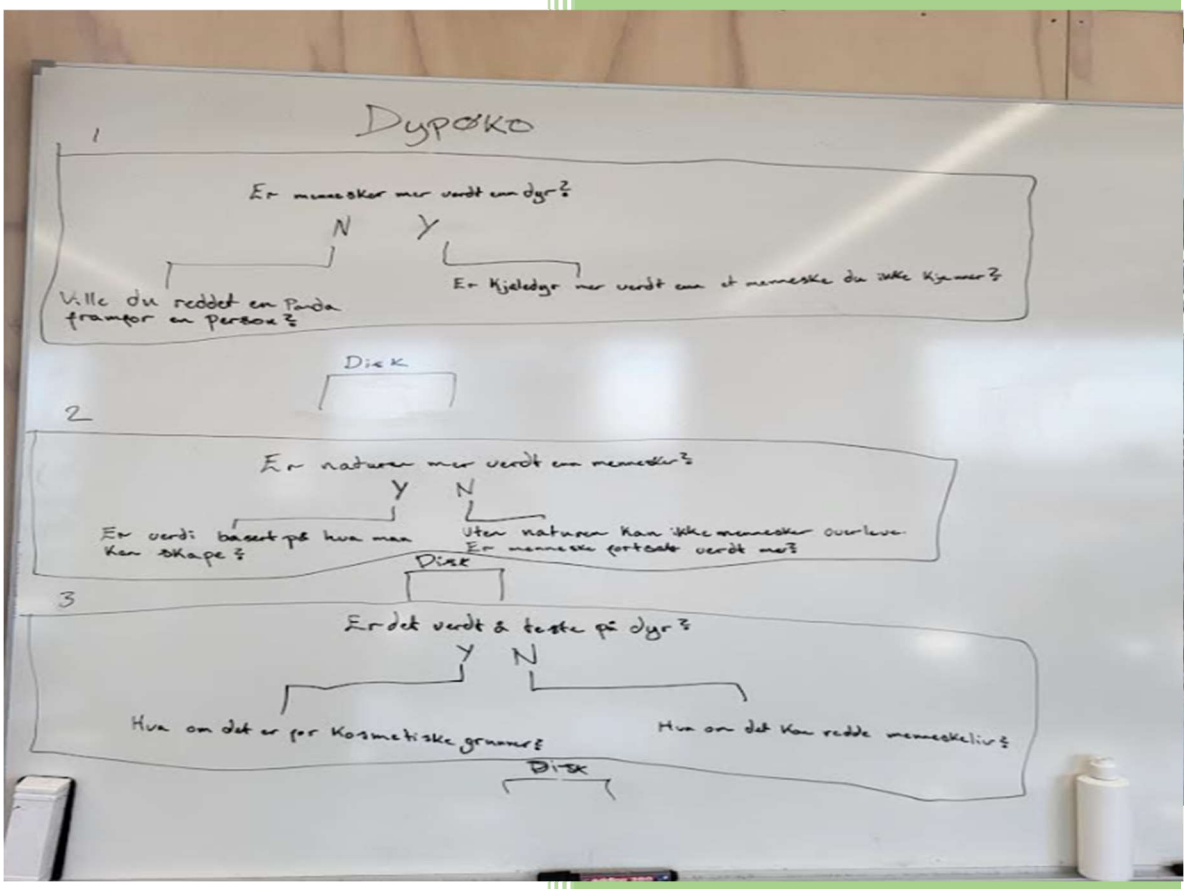
2023

Report for sprint 1



Brussels

02.06.2023

Innhold

‘

SPRINT 1 REPORT BRUSSELS

Group members

Oscar, Jan Roger, Jesper, Vegard, Bjørnar, Peter, Espen, and Joakim

Day 1 Sprint planning (Tuesday 30.05.23)

During this part of the week, we all got together to write up a backlog with the requirements of the product owners, and all user stories our product should include.

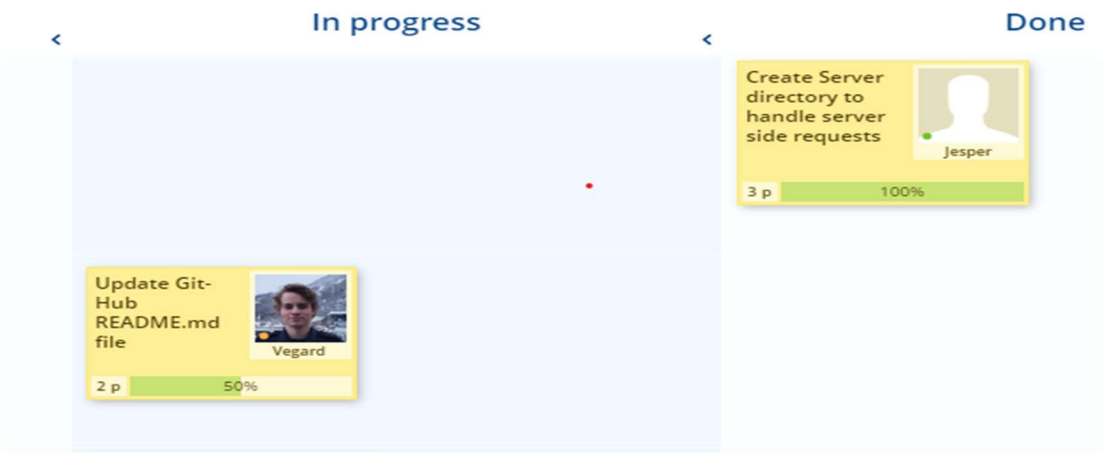
We started the day by setting up epics with the underlying features in our Scrumwise backlog. Afterwards, during the actual sprint planning, we started to map what our goals for this sprint were and started selecting user stories for our first sprint backlog.

Goals for the sprint

During the first sprint planning we all agreed that we needed to be a little conservative and feel out the process of a sprint before we could really be efficient during a sprint week. This does not mean we set our goals too low. We decided to focus on finishing the prototype based on the user feedback from the first user testing. We also wanted to prepare our work environment by collecting all participants in a GitHub repository and deciding on what type of solution to make. We also made a joint promise to prepare during the week, by sharpening the skills we collectively decided would be the most important for designing our product.

Day 2, first official sprint day (Wednesday 31.05.23)

On the first day of the sprint, we all met in a physical space. Here we each divided up who was going to work on the assigned tasks, considering only one person could be assigned to a user story in Scrumwise. It was also during this first day that we acknowledged that we had more than enough to do during this first week. When the first set of user stories were designated to the in-progress bar in Scrumwise, it was a simple matter to plan out the rest of the sprint.



(Image from our Task Board)

It also made it easy to keep track of everyone's area of focus, so when someone finished their task or made some solid progress. The next part of the process was to ask if anyone needed help with their task. If this were not the case, they could start selecting another item from the sprint backlog to put in progress.

Fixing prototype from user reviews

Two of our group members started updating our prototype. We had some great user-review feedback and ANCEP provided us with new color schemes, logo, and texts to help us improve the design. We focused on making a design that could be implemented across the whole solution. One point of improvement was making the "game" more accessible. Adding a large module to top of our previous certainly made the "game" easier to find, and we think it made it obvious what is in focus. Some users thought the old prototype was a bit bland, so we changed out some B&W pictures to color. Users had a hard time reading some of the text boxes and one of our users pointed out that children would need larger fonts. This was an easy fix, and made all pages look more filled out in general. To fit the solution, we had to redo some of the pages to give them more purpose by removing redundant information. Creating a page to prompt users who work in groups to discuss was an idea we had forgotten to implement during the design sprint and was a "must add" once we figured it was missing.

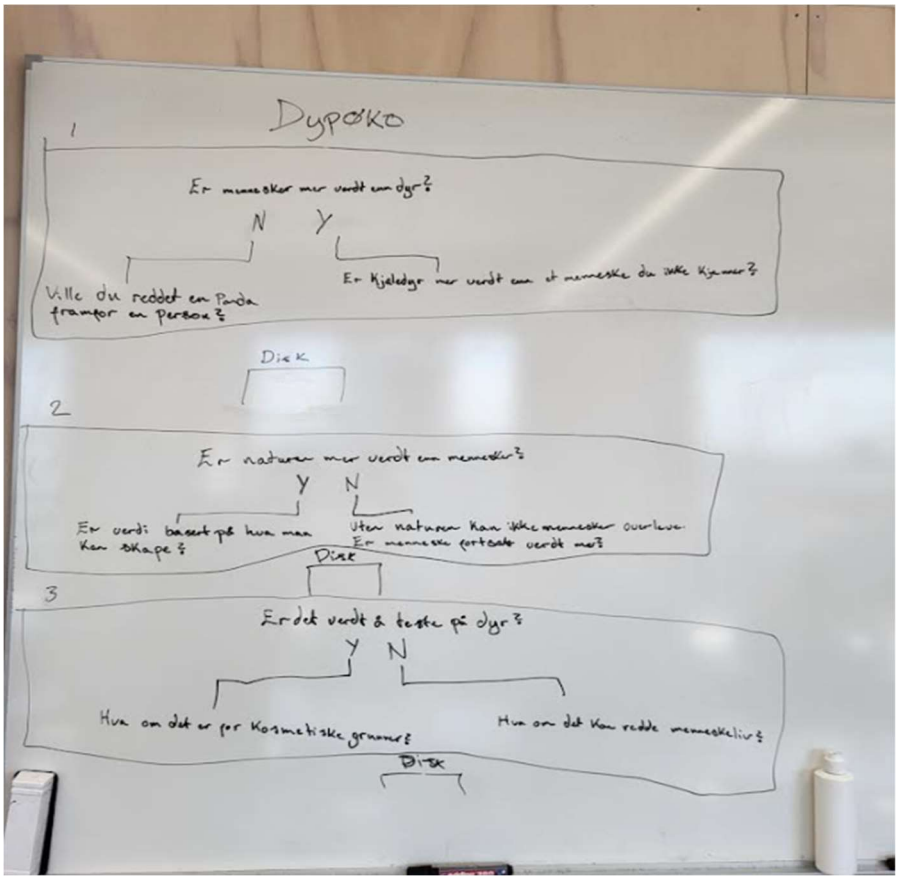
Research for our dilemmas

On day two, two of the members paired up and researched what categories we wanted to implement into our Ethical game (Dilemma game). We wanted this to closely align with some of Arne Næss's opinions, and philosophy. Though the categories did not have to directly align with Arne Næss's opinions, the questions needed to have some sort of relevancy to his work, or what he had openly spoken about. The research was relatively easy to complete, and we decided to use three categories for our Ethical game. The categories were "Ecology", "Minimalism", and "War & Conflict".

Creating dilemmas

When the categories were decided, we could move onto defining the questions for each category. This process was harder and proved to be more time-consuming than expected. This is the part of our project where we had to use our creativity, but also try to cater the questions to our audience.

In terms of the overall questions, we want to implement into our game, we initially planned for three introductory questions, and the questions are plain text with a given dilemma. When the user has finished the introduction, they are given three other dilemmas, but these are visually presented, meaning the users will be able to see the dilemmas visually and decide what the best option is. Visually presenting the dilemma and making sense of the questions given to the user turned out to be a tedious process in terms of not trying to influence the user, because there are no wrong answers. We did not manage to complete the dilemmas on day two and had to continue this process the next day.



(How we drew up our ideas for the questions on a white board on Day 2)

Learning React

Some of the group members did not have experience with React, so we set aside time to learn and brush up on our skills. We searched for videos and instructional materials to gain a better understanding of how React works.

Day 3 Second sprint day (Thursday 01.06.23)

We started the day with our daily scrum meeting. Here we addressed the importance of communicating if a task was unclear, or someone needed assistance on anything. Regarding the skill set of the group, someone took it upon themselves to act as a mentor for the other team members on various tasks. This was important to educate more members and solidify the t-shape of our group's skill set. This was also the day when our burnout graph was beginning to look like the optimal graph.

Continuing the creation of our dilemmas

After finishing the introductory questions on day 2 we were eager to continue with the dilemmas. Though we did not spend too much time on the questions this day, we managed to create some, but had some other obligations as well, so we restricted our time spent on this task. We ended up creating some dilemmas for our game, but we did not manage to complete the task.

Fixing prototype

After we had agreed on a design and color palette the previous day, we started implementing the design. We standardized the prototype by adding the same backgrounds, headers, and color palette on every site. Afterwards we played around a bit by adding contrast elements and other small fine adjustments to make the prototype seem livelier and more professional. We agreed to use a plugin to convert the Figma prototype into HTML and CSS code so we could use it as a template for how we wanted to develop the site and saving time on frontend development. The plugin we used was sensitive when it came to converting the code, so design elements outside the “screen” in Figma would also be included in the source code. Therefore, we also had to use some time to make the Figma design friendly for conversion by making sure everything lined up and worked properly.

Basic React App and Azure Deployment

Much of day 3 went to creating a basic react app and deploying it to Azure webservers. The React app itself only consisted of an empty html page, but setting up a React app, installing dependencies and deploying it to Azure took time and had to be researched by the group.

We faced some bugs, due to incompatible versions across plugins and Node versions, so some time went to resolve these bugs.

Some group members had worked with React and Azure in previous courses, while others had never used it, so the day also went to setting up the correct developer environment and learning React.

Last day plus retrospective (Friday 02.06.23)

Finishing dilemmas

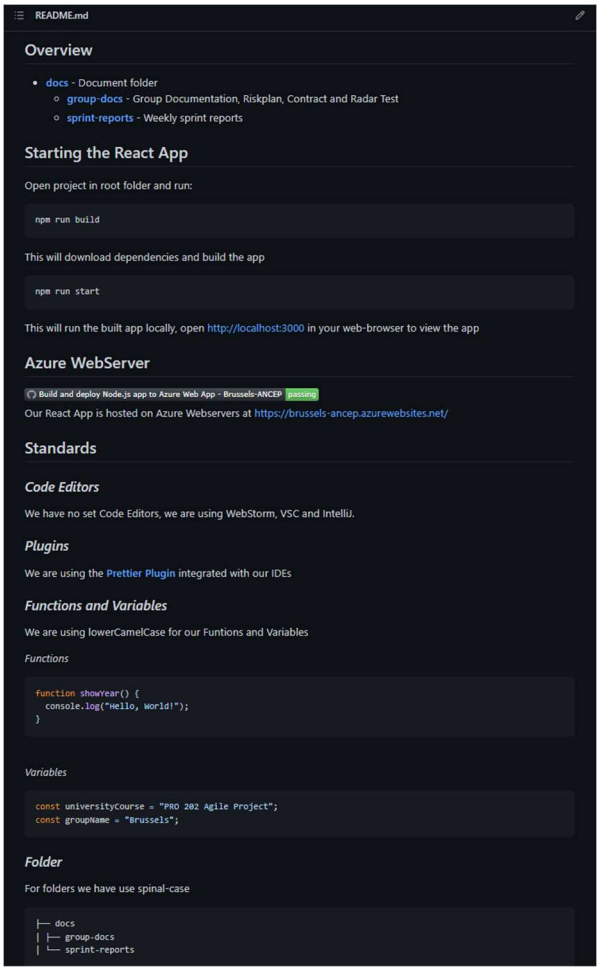
We wanted to have as much time as possible focusing on the functionality of our product next sprint, so we decided on finishing the dilemmas on Friday. We were struggling a little bit with our creativity, but working together as a group we created suitable dilemmas for all our categories. That said we did agree on creating a new backlog item for reformulating some of the dilemmas to be more understandable.

Standards and GitHub

We agreed as a group to use the Prettier plugin. Prettier works similarly across all our IDEs and will be the coding standard we will use going forwards.

As for Function and variable naming conventions, we decided to go for lower CamelCase, and directories are named with spinal-case.

Coding standards, how to start the react app and the link to the Azure website are in our GitHub readme.md file.



README.md

Overview

- docs - Document folder
  - group-docs - Group Documentation, Riskplan, Contract and Radar Test
  - sprint-reports - Weekly sprint reports

Starting the React App

Open project in root folder and run:

```
npm run build
```

This will download dependencies and build the app

```
npm run start
```

This will run the built app locally, open http://localhost:3000 in your web-browser to view the app

Azure WebServer

Build and deploy Node.js app to Azure Web App - Brussels-ANCEP passing

Our React App is hosted on Azure Webservers at https://brussels-ancep.azurewebsites.net/

Standards

Code Editors

We have no set Code Editors, we are using WebStorm, VSC and IntelliJ.

Plugins

We are using the Prettier Plugin integrated with our IDEs

Functions and Variables

We are using lowerCamelCase for our Funtions and Variables

Functions

```
function showYear() {
  console.log("Hello, World!");
}
```

Variables

```
const universityCourse = "PRO 202 Agile Project";
const groupName = "Brussels";
```

Folder

For folders we have use spinal-case

```
├─ docs
| ├─ group-docs
| └─ sprint-reports
```

(Readme.md from our GitHub repository)

Sprint Review

We finished the Figma prototype based on the user feedback we received during the design sprint. We managed to set up a simple webserver, running on Azure. The website itself only consists of an empty html page, but any change to our GitHub page will automatically deploy to the website.

We managed to finish the questions for our dilemma game, the remaining part for the questions is to adjust the "language" of the questions, meaning the questions need some fine tuning. This will be a feature on our next sprint.

Since React is new to some of the team members, we decided to include training for React as a feature on our Scrum Board. This proved to be necessary and was a productive part of our sprint this week (Sprint 1).

We planned to post an empty html site to azure Webserver, with Express. We managed to complete this task, with a good margin.

We set up our GitHub site and had to have some additional training for this, since some of the team members are not used to using GitHub this way.

We managed to define the acceptance criteria for what we are supposed to deliver next week, meaning we defined what the bare minimum of a module should be.

We managed to define our standards for our coding. This included subjects like naming conventions, layout of the code etc.

Next week, we will start working on the Frontend part of our website. We discussed as a team and concluded that this should be done in pairs or in groups of 3 on each page. This will divide our experience better, and it will also be a learning experience for those who are not used to using React.

Regarding the backlog, we are going to add a task for polishing our questions for our dilemmas, this will be carried out next week (Sprint 2).

Overall, we managed to achieve our goals and are ready for the next sprint week.

Sprint Retrospective

What worked

We all have an understanding that meeting physically will both increase our productivity and will allow for better communication. So, meeting physically on campus has proved to be valuable. This is something that worked really well within our group. We have an understanding that we start at 10:00 PM every day, so we don't have to communicate this every day.

In terms of reporting absence, we created our own Discord channel for this specifically, this makes it easy for everyone to both report their absence, but also to keep track of who is and will not be available on a given day. It is worth mentioning that the absence of our team

members has neither been a problem in terms of finishing our first sprint, nor has the absence of any team members been significant.

We have mostly been working in pairs, this concerns almost every task we have done. This has proved to be valuable in terms of creativity and productivity, but also a good way of sharing our experience. Meaning, some of the team members might be more capable when it comes to programming, and others may be more capable when it comes to designing. So, we have made sure beforehand that we try to share our experience as much as possible.

Our verbal communication has been on point, we make sure that everyone is up to date when it comes to deadlines, deliveries, but also requirements in terms of what we are trying to produce, meaning that we do not deviate from our initial project idea.

We used the template from our lectures to assign story points, this proved to work well with our own measurement of time.

What **did not** work so well

Our plan was to meet up every day at 10am at Urtegata. Multiple people have been showing up late, so we agreed to be more punctual. Especially because we start the day with daily scrum, and we need everyone. It is worth mentioning that this has not been a big problem, but it is just a small adjustment.

After this sprint it became obvious that we needed better sprint planning. We want to make sprint 2 more structured and accomplish it by splitting the task in smaller user-stories with better descriptions. This will make it easier for everyone to see what is going on and what needs to be done.

We realized that we could have done a better job of mapping our individual skillsets and experiences. This could have made the assignment of tasks easier and would also make it clear what skills we need to work on.

We assigned all the tasks before the sprint began in Scrumwise. It is not possible to assign a task to more than one person and we mostly work in pairs. This left an imbalance to assigned story points, where people who did the same amount of work had no assigned story points. This could lead to jealousy/spite and from now on we will not assign task in Scrumwise, but rather assign task verbally in our sprint planning.

Additional documentation

Timesheet

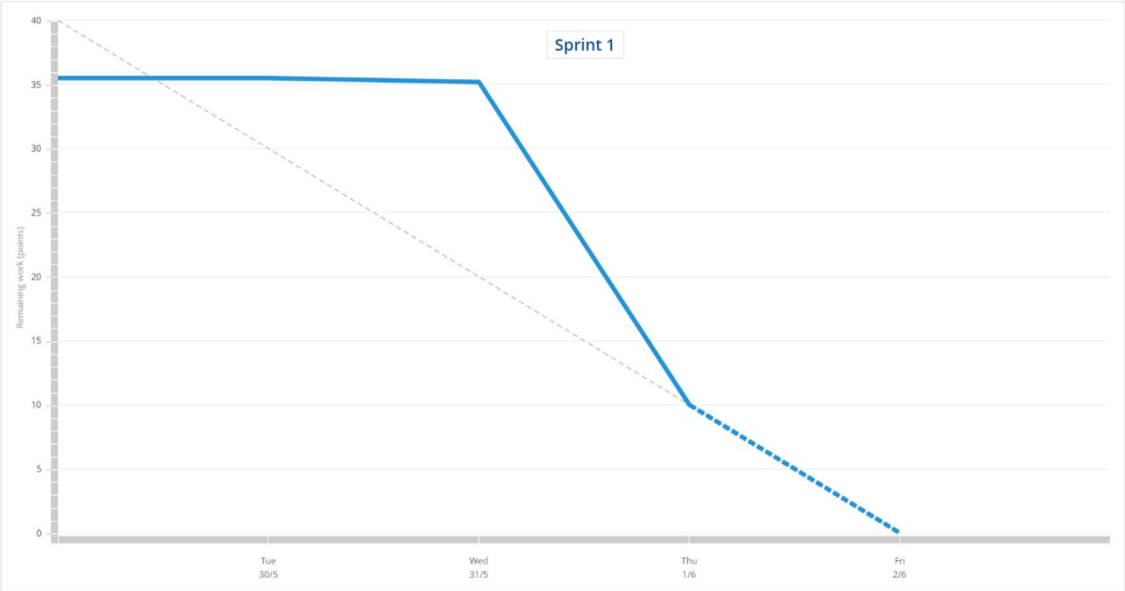
Used by Everybody When Sprint 1 Total 141.3 hours

Tuesday 30/5	Jan Roger	3 hours		In the backlog item "Sprint planing"
Tuesday 30/5	Bjørnar	3 hours	Sprintplaning	In the backlog item "Sprint planing"
Tuesday 30/5	Jesper	3 hours		In the backlog item "Sprint planing"
Tuesday 30/5	Joakim	3 hours		In the backlog item "Sprint planing"
Tuesday 30/5	Peter	3 hours	Sprint planning	In the backlog item "Sprint planing"
Tuesday 30/5	Bjørnar	0.5 hours	https://github.com/dpox...	In the backlog item "Create github repo"
Wednesday 31/5	Jan Roger	4 hours	Defined questions for the...	In the backlog item "Define questions"
Wednesday 31/5	Oscar	4 hours	Defining good questions...	In the backlog item "Define questions"
Wednesday 31/5	Joakim	4 hours		In the backlog item "Refresh your react skills"
Wednesday 31/5	Espen	4 hours		In the backlog item "Refresh your react skills"
Wednesday 31/5	Jesper	2 hours	React repetition	In the backlog item "Code react"
Wednesday 31/5	Jesper	4 hours		In the backlog item "Code react"
Wednesday 31/5	Joakim	2 hours		In the backlog item "Code react"
Wednesday 31/5	Jan Roger	2 hours	Researched topics for the...	In the backlog item "Research for categories/questions"
Wednesday 31/5	Oscar	2 hours	Research arne...	In the backlog item "Research for categories/questions"
Wednesday 31/5	Peter	1 hour	Helped with defining...	In the backlog item "Research for categories/questions"
Wednesday 31/5	Peter	4 hours	Changed the design of...	In the backlog item "Change prototype from user feedback"
Wednesday 31/5	Bjørnar	2 hours	Started changing old...	In the backlog item "Change prototype from user feedback"
Wednesday 31/5	Peter	1 hour	Changed color schemes t...	In the backlog item "Change color scheme and logo"
Wednesday 31/5	Bjørnar	3 hours	After my meeting w Sofie	In the backlog item "Change color scheme and logo"
Wednesday 31/5	Vegard	2 hours	Reading notes on Github...	In the backlog item "Create simple webserver"
Wednesday 31/5	Vegard	2 hours	Research react setup	In the backlog item "Create Client directory to handle client side requests"
Thursday 1/6	Jan Roger	1 hour	Worked on Defining...	In the backlog item "Define questions"
Thursday 1/6	Oscar	1 hour		In the backlog item "Define questions"
Thursday 1/6	Espen	3 hours		In the backlog item "Sprint planing"
Thursday 1/6	Oscar	3 hours	Sprint planning	In the backlog item "Sprint planing"
Thursday 1/6	Vegard	1 hour	Updated readme.md file...	In the backlog item "Update GitHub README.md file"

Export all

(Our timesheet, but we could not fit the whole timesheet into one image)

Burndown chart



(Burndown chart after our first sprint)

Product backlog

Change prototype from user feedback		3 points	Done	100%
Change color scheme and logo		2 points	Done	100%
Add a backlog item				
Get a site up and running	Epic	12 points	Done	100%
Create github repo	Other	1 point	Done	100%
Create simple webserver		3 points	Done	100%
Create Client directory to handle client side requests		3 points	Done	100%
Create Server directory to handle server side requests		3 points	Done	100%
Deploy to azure		2 points	Done	100%
Add a backlog item				
Learning react	Epic	5 points	Done	100%
Refresh your react skills	Other	2 points	Done	100%
Code react		3 points	Done	100%
Add a backlog item				
Update GitHub README.md file		2 points	Done	100%
Create a Frontpage	Epic	6 points	Ready for sprint	
Import Prototype Frontpage to React		3 points	Ready for sprint	
Create header		1 point	Ready for sprint	
Create routing from frontpage		2 points	Ready for sprint	
Add a backlog item				
Create Archive	Epic	7 points	Ready for sprint	
Import prototype Archive to React		2 points	Ready for sprint	
Find suitable text/info	Other	1 point	Ready for sprint	
Create frontend		2 points	Ready for sprint	
Add Routing		2 points	Ready for sprint	
Add a backlog item				
"Etisk Spill" Backend	Epic	12 points	Ready for sprint	
Create a yes-no react component		5 points	Ready for sprint	
Setup a backend database, to store userinput		3 points	Ready for sprint	
Retrieve "Etisk spill" data from Database		4 points	Ready for sprint	

(Our product backlog after sprint 1)