



Forside IS-304: 2020

Tittel:

Emnekode	IS-304
Emnenavn	Bacheloroppgave i informasjonssystemer
Emneansvarlig:	Hallgeir Nilsen
Veileder	Hallgeir Nilsen
Oppdragsgiver:	Agder Energi

Studenter:

Etternavn	Fornavn
Johnsen	Jarle N
Kittilsen	Christian
Lefdal	Jørgen B
Lindemann	Daniel H
Reisæter	Daniel T
Theissen	Thomas

Jeg/vi bekrefter at vi ikke siterer eller på annen måte bruker andres arbeid uten at dette er oppgitt, og at alle referanser er oppgitt i litteraturlisten.	JA X	NEI —
Kan besvarelsen brukes til undervisningsformål?	JA X	NEI —
Vi bekrefter at alle i gruppa har bidratt til besvarelsen	JA X	NEI —

Foreword

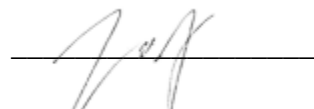
The purpose of this report is to describe and inform the reader of our work process and thoughts throughout this project. The report gives detailed information about our task to develop a Microgrid simulator for Agder Energi. This project is a cooperation between us as students, a project supervisor from UiA and a product owner from Agder Energi. Creating a microgrid simulator that can easily show the benefits of a microgrid, is both an exciting project for us as students and a great tool for a business developer at Agder Energi.

Our team from UIA consists of a group of 6 people from different parts of the country, with different ages and backgrounds. The work has been conducted in an office at Agder Energi most of the time, but after a quarantine due to a global virus, the work was done individually in everyone's home.

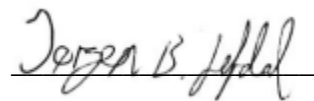
We want to give our thanks and respect to our supervisor Hallgeir Nilsen for help with the project and to our contact at Agder Energi, Erik Åbyholm, and the IT worker there, Thomas Hestenes. We would also like to thank OneCo, Pixii and Mechatronics for great assistance in this project.

Kristiansand 19.05.2020

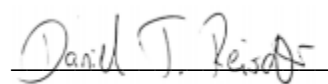
Johnsen, Jarle Nordby



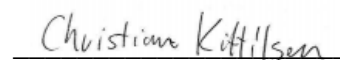
Lefdal, Jørgen Barlund



Reisæter, Daniel Trydal



Kittilsen, Christian



Lindemann, Daniel Helleve



Theissen, Thomas

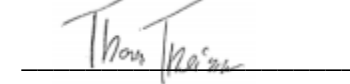


Table of contents

Summary	1
1 Introduction	2
1.1 About Agder Energi	2
1.2 About microgrids	2
1.3 About our project and product	3
2 Central Decisions	4
2.1 Methodology	4
2.2 Defining scope	4
2.2.1 User stories	4
2.2.2 Moscow	5
2.3 Programming choices	6
2.4 Risk analysis	7
3 Running the project	8
3.1 Scrum team	8
3.2 Scrum master	8
3.3 Product backlog	9
3.4 The Sprint	10
3.4.1 Daily Scrum	11
3.4.2 Sprint planning meeting	11
3.4.3 Sprint review and retrospective	11
3.5 Burndown Chart	12
3.6 Quality assurance	13
3.6.1 Research	13
3.6.2 Pair programming	14
3.6.3 Version control	14
3.6.4 Code standards	14
3.6.5 Tests	15
3.6.6 Communication	16
3.6.7 Journal	16
4 The product	17

4.1	Assumptions	17
4.2	Video	17
4.3	Deployment	17
4.4	Security	18
4.5	Documentation	18
5	Reflections	20
5.1	Methodology and project management	20
5.2	Challenges	20
5.2.1	Communication	20
5.2.2	Knowledge	21
5.3	Changes in the project and product	21
5.3.1	Graph	21
5.3.2	Changes to scope	21
5.3.3	How-To	22
5.3.4	Django - Flask	22
6	Statement from Client	23
7	Self-evaluation	24
7.1	Group	24
7.2	Members	24
8	References	27

Summary

We were tasked with creating a microgrid simulator for Agder Energi. In this report we will be providing information about what a microgrid is and how we executed the project to create it. The report is divided into different parts. These include central decisions we made, how we ran the project, the finished product, reflections, statement from client and self-evaluation.

At the start of the project we decided to use Scrum as our methodology for this project. To define the scope, we chose to utilize user stories to find functions and used the MoSCoW method to prioritize them. We had a lot of freedom regarding programming choices as the client did not have any preferences. We also decided to analyze potential risks and create a risk matrix to mitigate harm to our project should anything happen.

When running the project, we used different techniques and methods to make sure that the team worked well together and that we had good control during the whole project. In addition to ensuring good teamwork, these methods helped us control that the product had good quality. By conducting good research, having good communication, using different tests and work techniques we could make sure that the product had as good quality as possible.

The finished product is a working system that is able to simulate how the peaks can be reduced by adding certain components to the microgrid and how it affects costs. By communicating with the product owner, we ensured that the product had all the requirements that were requested.

In review of this project we can see that adding microgrid functionality to a neighborhood can be beneficial for both customers and providers. Making small changes had great impacts on power consumption and we will probably see more systems like these in the future.

1 Introduction

In the following paragraphs we will introduce some important aspects of our project.

In the class IS-304, which is the bachelor thesis project in information systems, the students are supposed to Implement an IT / IS related project, define quality and control progress in a project. (UiA, 2020)

Our project was based on creating a simulator for a Norwegian power company to research if a microgrid would be beneficial for the different stakeholders. A microgrid is a local energy system. Agder Energi is the power company that we worked together with on this project and the idea of the project came from a business developer there. A microgrid is relevant for the future of the power companies and they wanted to research this at an early stage.

1.1 About Agder Energi

Agder Energi AS is a Norwegian energy company involved in hydroelectric power generation, electricity distribution, electricity trading and services for customers in both businesses and the private segment. The company operates in Norway, Scandinavia, and the rest of Europe. They focus on developing hydroelectric power in Norway and investing in new trading solutions for the European energy market. Agder Energi supplies clean energy to the society and is actively involved in developing renewable energy solutions for the future. (Agder Energi, s.a.)

The company is owned by the 25 municipalities of Agder with a total of 54.47% of the shares, and Statkraft Regional Holding AS with 45.53% of the shares (Agder Energi, 2020). The company headquarters is located in Kristiansand and they own a total of 49 hydroelectric power plants in Agder.

1.2 About microgrids

A microgrid is defined by the U.S. Department of Energy Microgrid Exchange Group as: “A *microgrid is a group of interconnected loads and distributed energy resources within clearly defined electrical boundaries that acts as a single controllable entity with respect to the grid. A microgrid can connect and disconnect from the grid to enable it to operate in both grid-connected or island-mode*” (Berkley lab, s.a.). In our project, the statement from the U.S. Department of Energy Microgrid Exchange Group about what a microgrid is, differs somewhat from our project. The main goal of our project is more of a “hybrid” microgrid, which is more focused on cutting costs for both the supplier and consumer by shaving off the tops of electrical peaks, which are the peaks in electrical consumption in a given time period. These peaks are often located in the mornings and evenings of the day as there is more consumption of electricity at these times. An example of an annual peak period is Christmas eve when several households turn on their stove and lights etc. at the same time.

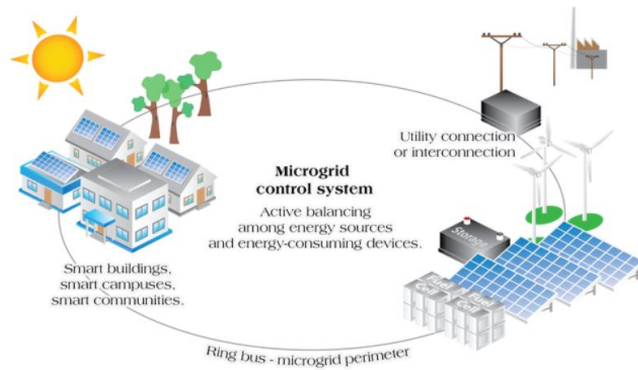


Figure 1: Neosun Energy, what is a microgrid?

A microgrid normally operates while connected to the centralized grid, like it is depicted in figure 1, but it has the ability to break off and operate on its own using local energy generation and storage if needed. This can be helpful in times of crisis like storms, power outage or other reasons (Lantero, 2014). One other great benefit of a microgrid is that it can cut costs. By lowering the power demand from the centralized grid, it gives the consumer lower power bills, and the power company has lower costs of upgrading the local transformer stations. One of the major costs for a company like Agder Energi is upgrading the transformer stations to be able to handle the peaks in power demand.

It is possible to affect this with a microgrid using local batteries for power storage, solar panels and wind turbines for power production, and a program that controls the entire grid.

1.3 About our project and product

The goal of our project is to create a system that can analyze data from a substation in a microgrid and simulate how different components, such as batteries, solar panels and electrical car chargers, affect the flow and usage of power within a microgrid. The main purpose of the system will be to determine if the implementation of these components in a microgrid is beneficial for both Agder Energi and their customers, mainly with regards to costs.

A simulation is an easy and understandable way to visualize how power consumption at a substation is influenced by the changes in a microgrid. The main purpose of the product is to find ways to reduce costs for both Agder Energi and their customers. It is desirable that the total consumption is lowered, but reducing peaks is most important. This is due to the peaks being the primary contributor to higher costs for customers, once new electricity tariffs get introduced, and requiring the microgrid to be upgraded further. In the long run, the goal is that a microgrid can be used to help make neighborhoods and other small areas self-sufficient in terms of power.

2 Central Decisions

In this section of the report, we will highlight central decisions we have made throughout the project. This will include methodology, defining scope, programming choices and risk analysis. We will also mention how we used them and some of our thoughts around this.

2.1 Methodology

During this project we have been working with the use of Scrum. Scrum is an adaptive software development methodology. It uses different processes and techniques so that you can continuously improve the product, the team, and the working environment. (Scrumguides, 2017) Scrum is responsive to a highly changing, dynamic environment when the users might not know exactly what is needed and priorities change frequently. It is a type of social engineering that describes how a team of developers can work together in short mini projects. (Satzinger, Jackson & Burd 2016, p 45)

We chose to use Scrum because it suits our project very well. As stated in Satzinger et.al (2016) it is an effective methodology when the outcome of the project is slightly diffuse and when changes are expected. By using Scrum in this project, we have really gotten to know the Scrum framework. We decided to be very strict and follow the methodology to see how this affected the project. When the covid-19 quarantine regulations started we realized how important Scrum was for our teamwork. We had all the tasks in our backlog, and everyone could see how far we were in the project. By still conducting the daily scrum every day we were also informed on what the others were working on and if they needed any help.

2.2 Defining scope

When defining our scope, we used a couple of methods. We created user stories to define functions for the system, and MoSCoW to categorize and prioritize them. By using these methods, we ensured that the team members worked on the important things first and avoided spending time with functions that we did not prioritize high.

2.2.1 User stories

Satzinger, Jackson and Burd defines a user story like this: “A user story is usually one short sentence in the everyday language of the end user that states what a user does as part of his or her work. In other words, a user story describes a goal the user has when using the system”. (Satzinger, et.al., 2016, p 45)

We write user stories to be able to form a basis of the list of functions that the system needs to carry out. They document the requirements by focusing on *who*, *what*, and *why* for each function. Here is a list of our user stories. In them, a user refers to an employee in Agder Energi:

As a user, I want to be able to upload a dataset from a power station to show a graph of how power is used.

As a user, I want to see a graphic simulation to understand how a microgrid works.

As a user, I want to be able to add batteries to the grid so that it shows how peaks can be lowered.

As a user, I want to add solar panels to a grid in order to see what difference a solar panel does to the microgrid.

As a user, I want to be able to add car chargers to the microgrid to see how this affects the power demand.

As a user, I want to switch on and off all heating cables in the grid in order to see what effect heating cables have on a microgrid.

As a user, I want to add windmills to a grid in order to see what difference windmills do to the microgrid.

Our user stories were based on what the users should be able to achieve when using the system. To create the user stories, we used information gathered from meetings with the product owner along with what we gathered from research. With this information we imagined ourselves in the role as a user and made the user stories.

Making user stories helped us a great deal in the planning process. We experienced that this gave us a better understanding of what we envisioned the product to be. The user stories were vital in creating the scope for the project, but although the scope could change over time, the user stories always stayed the same. Sometimes during the project, we would come up with additional functionality that would be nice to have, or the product owner would expand the requirements, so the scope changed. However, the user stories represented functionality that the product owner wanted from the start and that remained as requirements for the product, so these did not change.

2.2.2 Moscow

In a system development project where time is fixed it is important to plan the prioritization of the different parts of the system in order to keep deadlines and to progress as planned. We used the MoSCoW method to help us do this. (agilebusiness.org, 2018)

After creating the user stories, we used the MoSCoW method to figure out what tasks that needed to be prioritized and what functions that should be worked on first. This task was done in collaboration with the product owner, as after we had used user stories to get functions, we asked him how he would prioritize the tasks we presented. When we worked out the prioritization of the

functions, we sorted them into “must have”, “should have”, “could have” and “will not have this time”.

The use of MoSCoW in our project was very helpful as it helped us visualize the importance of the different features in our project. This made it easier to decide what our next tasks should be. By using the MoSCoW method we also made sure that both the team and the stakeholder had a common understanding of what to prioritize.

2.3 Programming choices

Due to a very few requirements from the customer, we had a lot of freedom in the project. Since we had this freedom when we started to develop the system, we had to make a couple of choices. The product owner was involved in every big decision that we made but trusted our choices. Firstly, we decided to use Python as our main programming language, and Flask for the web-application framework. We chose this because Python is quite powerful, commonly known, well documented and easy to learn. (Rongala, 2015) This made for a good base for us. For front-end coding, we used JavaScript, which helped us connect front-end and back-end asynchronously.

When designing the web-application we resorted to HTML5, CSS and CSS’s Bootstrap library. It is simple to use and quite powerful for the purposes of this system. These also have good documentation online, which makes it easy for Agder Energi to continue working on the system. (W3org, s.a) To display the graph on the website we used Grafana, as this was clean, easy to use and easy to manage. It was well optimized as well and had no issues displaying thousands of data points.

During the project we decided to have an animation below the graph to explain for the user how a microgrid works to make it simple. First, we looked into using SVG(scalable vector graphics) but landed on using Adobe Animate as the tool to make JavaScript graphics. It was an interesting process because we did not have much experience in that field of work, and it was valuable front-end experience for future work. What we learned from this process was that we should have made more detailed prototypes regarding the animation, since the only prototype we made was a low-fi sketch.

For the server we started by running the system of our own Linux computers. Along with the web-application we needed a database, for this we chose MySQL as we had a lot of experience with it, which saved us time learning something else. It is also a very commonly known database language, so in case Agder Energi would like to change anything in the database, they would have the necessary knowledge to do so.

When we had a good system base, we wanted to get it accessible from anywhere, which is when we gained access to Agder Energi's Azure Cloud. Here we created a virtual Linux machine to run the app, database and Grafana instance.

Our product owner wanted a basic security layer. For this we found it easiest to apply an IP filter to the product, which blocks all unwanted IPs from connecting. This meant that you could only access the app from specific locations, or with a VPN to those locations; one of these being Agder Energi headquarters.

2.4 Risk analysis

Risk analysis is the process of finding and analyzing potential problems that can negatively impact a project. Risk analysis is an important part of a project to avoid or mitigate potential risks. (Rouse, 2020)

For the risk analysis in our project we made a risk matrix. The matrix was too large to be shown here. This is a great way to analyze threats and prepare for them in a project. The first thing we did in the risk matrix was to identify and write down all the possible threats we could think of in the project, and then give each of the threats a consequence and estimate the likelihood of that threat occurring. After we had given a good estimate of the consequence and likelihood rating, we multiplied them together to find the total risk of the threat. We call this the risk factor. The higher the risk factor, the more dangerous the threat is for the project. The main goal with the risk matrix after finding the risk factor for each threat, is to add measures for the threat to reduce the likelihood of the threat occurring in the first place. Then write down what actions you can do to mitigate damage if the threat actually occurs.

One example of a threat in our risk matrix is "incorrect choice of tool or technology". The reason we added this threat was because in this project we used a lot of tools and technologies we had never used before. It was important for us to prepare for such a mistake. On a scale from 1-5 we rated the consequence of this threat a 4, and the likelihood rating 3. After multiplying these numbers, we found out that the risk factor is 12, which is quite high. The measures we added for the threat to not occur was to do thorough research and have a longer planning phase. The actions we added to mitigate damage was to evaluate the time it would take to change tools/technology and estimate the time it would take to make the tool/technology work, if possible.

The risk matrix document was a living document, which means it can always be edited and updated. At the start of each sprint we went through the risk matrix and updated it if it was necessary. Due to the Covid-19 crisis we added a few new risks to the risk matrix that we did not think of before.

3 Running the project

While working on this project we used methods from the Scrum framework to help with task control, time control and general guidance. Quality is also an important aspect in every project, and we discuss how we achieved good quality later in this section.

3.1 Scrum team

To ensure that we had stable progress, quality, and success, we decided to divide our team into three different groups. We had two people on backend, two on frontend and two on project management. Retrieving and understanding information was a big part of the beginning of the project and this is why we chose to have two members in management. One of the members would later on help front end with their work. Since the team consists of people with different skills and personal preference of interest, we chose to divide it this way. This made sure that every team member was working on what they enjoy the most, which in turn ensures better quality because people tend to give more care to things they enjoy. Even though we divided the team, we all worked together and learned from the others to make sure everyone was involved in everything. In addition to these groups we had a product owner who also was a part of the Scrum team. Due to the fact that the product owner was only partly involved in the project, many of his responsibilities fell on the Scrum master. One example of this could be making sure the backlog was updated and sorted correctly.

3.2 Scrum master

The Scrum master is the person in charge of the Scrum elements in the project. His/her job is to make sure everyone is participating in Scrum and are following the procedure. The Scrum master needs to keep everyone focused and know what they are working on. The Scrum master protects the team, so they do not overdo work, or do any work duplication. (Satzinger et.al, 2016, p 317-318)

In our project, Jarle was the designated Scrum master and had an overview of all the parts of the project. A scrum master would in a normal situation not be a part of the team, but as this is a bachelor project with a short duration, we had to use the scrum master in both. He made sure that the different groups(back-end and front-end) had the information they needed to do their work. The daily scrum (will be described later) was also controlled by the Scrum master. If there were any issues keeping the team from doing their job it was the Scrum master's responsibility to ensure that proper help was acquired in order to keep the workflow.

By having a scrum master, the team always had someone to report to, and to be designated tasks from. It made it a lot easier as a team that only one person had to keep control of all the tasks, making the project clearer. However, sometimes the scrum master was not given proper feedback

on what the different team members were doing, resulting in the scrum master not having a total overview of the different members. As a team member, receiving information on the go from the scrum master, while doing something else was a very effective way of working, e.g.: someone was working on back-end tasks, and needed information about a solar panel within the next hour. The scrum master could find that and give it to the other member, resulting in very effective work as they did not have to stop their work in order to find that information.

3.3 Product backlog

The product backlog is a prioritized list of tasks that need to be completed during the project. It is usually supplied by the product owner. At the start of a project the project backlog is usually quite vague and not very detailed. This is because the product backlog is at its most effective in its agile form, where entries are rewritten, reprioritized and sometimes removed from the backlog completely. (Evens, 2014, s.26-28)

To control our backlog, we chose to use Trello. Trello is an application that lets you organize your project by maintaining cards and lists of tasks. You can add deadlines, comments and assign members as responsible for completing each task (Trello, 2017). In the beginning the backlog was created by going through the MoSCoW analysis and the user stories. The team discussed which tasks needed to be added to the backlog to complete the different user stories and their functions. At the start, the tasks were a bit vague, but as we started working, we decided to make these a bit more detailed. When we got changes in scope, got new ideas or requirements, we added tasks to the backlog so they could be pulled into a later sprint.

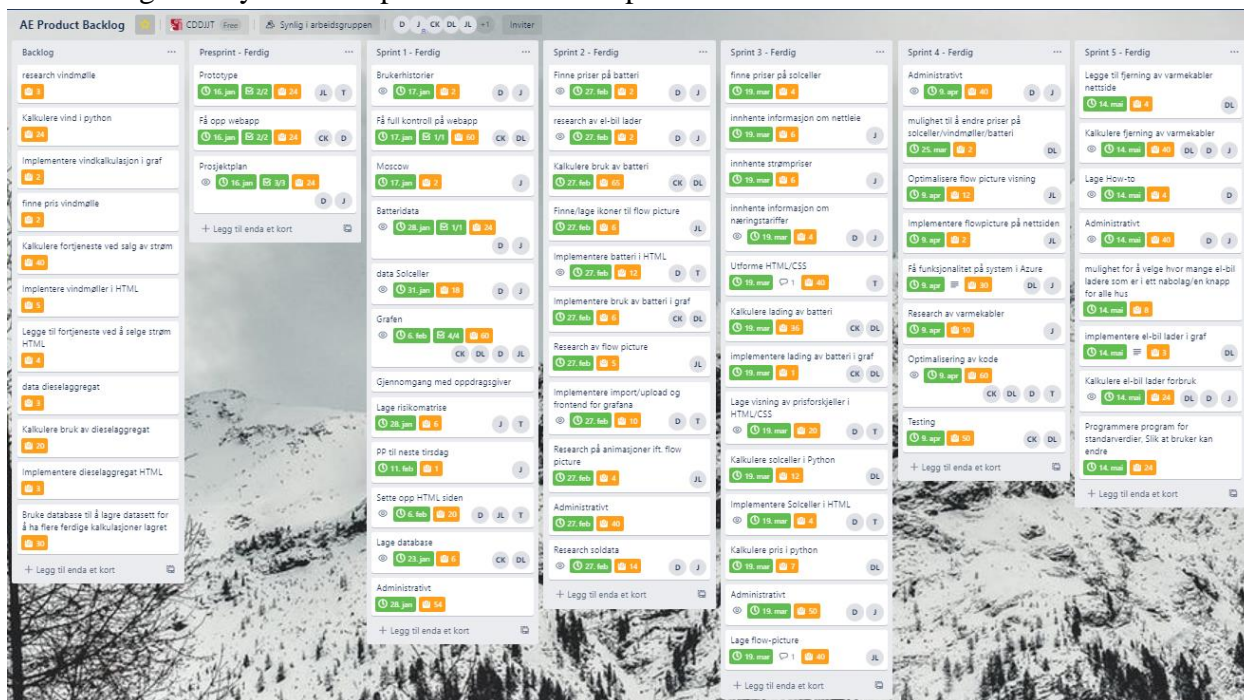


Figure 2: Product Backlog

By dividing our product backlog into more specific tasks, we got a lot more time control on the different tasks. Early in the project the tasks were more vaguely defined, and we found it hard to estimate how many hours were expected to complete them. This could increase the potential for a failed sprint. (Verwijs, 2017)

One example of this can be the task; “Add solar data functionality to the program”. We divided this into; retrieve data regarding solar panels functionality, find prices on solar panels, find sun data, calculate solar panel power production, and add solar data to program and graph.

3.4 The Sprint

Sprints are the “main” event in Scrum. It contains a given time period of one month or less with different elements. It consists of Sprint planning, Daily scrums, Sprint review and the development work.(Scrumguides, 2017)

The project was divided into six sprints that lasted for three weeks each. Different milestones were set that should be successfully completed after each sprint. In the sprint planning meeting the team pulled tasks from the product backlog and added them to the sprint backlog. In some sprints we managed to complete the tasks earlier than expected and it was decided that we would pull some additional tasks from the product backlog.

The different milestones in our project were:

Presprint: Finish prototype, backlog, and work method.

Sprint 1: Html page, web-app ready, database ready.

Sprint 2: Show power consumption in graph and have battery functionality in the program.

Sprint 3: Charging of battery in graph, solar panel functionality, pricing information.

Sprint 4: Flow picture on web page, deploying solution online.

Sprint 5: Smart heating and car charger functionality.

As we divided our sprints into 3 weeks it was easy to figure out how many tasks, we were able to complete in that time since the tasks were well described. Having 3-week sprints worked well for us to avoid spending too much time planning and discussing. We experienced that it was smarter to have a bit longer sprints (considered having 2 weeks sprints) to get more work done. Having longer sprints does have some downsides. In one of our sprints we did the tasks faster than we expected and if we follow the Scrum framework strictly, we are not supposed to pull new tasks during the sprint. As we wanted to avoid downtime in the project and we work in an agile way, we decided to pull some new tasks from the product backlog and add them to the sprint.

3.4.1 Daily Scrum

Daily Scrum is an activity where all team members meet to discuss and synchronize the work they are doing. This is done at the start of every day in the sprint and is usually time-boxed to around 15 minutes. Members of the team answer three questions:

- What have you done since the last daily Scrum?
- What will you do by the next daily Scrum?
- What kept you or is keeping you from completing your work?

(Satzinger et.al, 2016, p 317-318)

At Agder Energi every workday started with a Daily Scrum meeting. The scrum meeting was led by the Scrum master and every team member told the team what they were working on that day and if there were any challenges with their task. After the global pandemic of coronavirus, the work communication and daily scrum was conducted via a voice meeting on the communication platform “Discord” every morning.

By having a daily standup meeting every day, we ensured that every team member knew what the others were working on and how far along they had come. This was also very helpful for the Scrum master to keep a good overview of everything and make sure that the team had the information that they needed.

3.4.2 Sprint planning meeting

At the beginning of each sprint the team gathered, sometimes without the product owner, for a sprint planning meeting. It was decided what tasks that should be pulled from the product backlog and added to the sprint based on their prioritization. All the tasks in the backlog had estimated hours, so we pulled the number of hours that we expected that we could complete during the sprint.

3.4.3 Sprint review and retrospective

The sprint review and retrospective were sometimes conducted together with the sprint planning meeting. In these meetings, the team reflected on what worked well and what did not in the previous sprint and discussed how it could be improved in the next sprint. We reviewed the work that had been done and decided if it met the requirements. If the product owner were present, he would decide if the completed work was accepted or not.

The upsides we experienced from having these meetings was that we could learn from our previous work and improve upon it. If one of our work methods had proven to not work well, we could find a better way to do it for the next sprint. Another upside is that by having the product owner review and accept our work, it ensures that what we implement into the product is what the client wants.

3.5 Burndown Chart

The Scrum Burndown Chart is a visual measurement tool that shows the completed work per day against the projected rate of completion for the current project release. Its purpose is to show that the project is on the track to deliver the expected solution within the desired schedule. (Evens, 2014, s.33-34)

To easily make a burndown chart we figured that using Excel was the best way to do it. There are a lot of programs that can be used for this, but Excel has a lot of possibilities and we knew the program. All the tasks were added to an Excel sheet and we made a graph that showed estimated hours for the project, actual hours spent and tasks remaining. It also showed how many tasks were completed in each sprint.

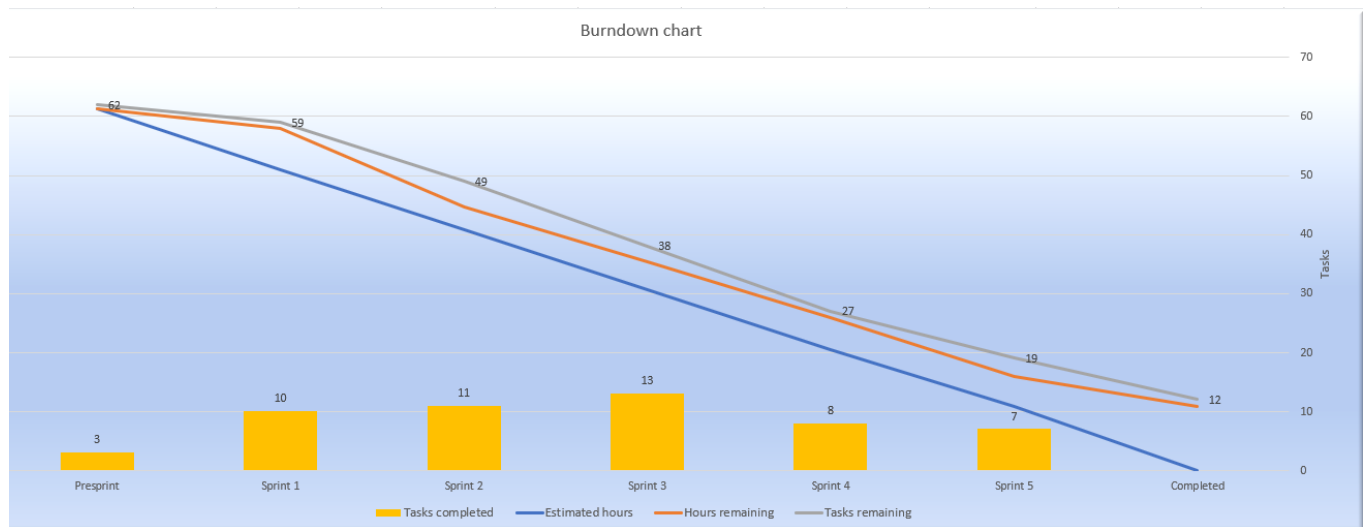


Figure 3: Burndown Chart

The way the burndown chart helped us was mainly that it gave us a very clear visual understanding on how much was remaining. As you can see in the burndown chart above, very early we miscalculated the time schedule a bit. This is because we did not take the winter break into consideration. This miscalculation followed the project all the way to the end in terms of always being a little behind the schedule. Because of this miscalculation, we did not have time to finish every task that we had in the backlog. However, we did finish all the tasks that we decided was a must have in Moscow as well as the should haves. In the chart you can also see that we did not spend as many hours as expected, which is due to a lot of hours used on writing the different reports for our courses. We did not add these tasks to the backlog as they were not part of the microgrid project, therefore, all the hours were not shown there.

3.6 Quality assurance

Quality in our product means that the product meets all requirements, the calculations are right, and correctness is high. These are 3 values of quality defined by Myron I. Peskin & James J. Hart (1996)

Meeting all the requirements is done by keeping the perspective of the customer in mind. In our project, the customer was our project owner and all the requirements were built in cooperation with the project owner and reviewed at the sprint reviews.

To ensure that the program calculated things properly, we used excel and calculated things manually while checking the values we received against the values given by the system.

To define quality by its means of “correctness” involves three different terms that we had in mind when we did front-end, back and are a part of how our code standards are. These terms are:

Completeness: This means that the system conforms to the user’s specifications and must fulfil the entire requirements of the user. This is primarily focused on design and planning of the design.

Consistency: A system must have uniform designs, consistent implementation techniques, and complete documentation. These must be consistent throughout the project or the system will fail to perform as required. For the system, that means that front-end and back-end are done in a standardized form and can be replicated easily. This is a quality that we had to keep in mind if Agder Energi would want to do improvements to the system in the future.

Traceability: To determine correctness, the user’s requirements must be totally traceable throughout a system. For this project this is by having good version control, testing and documentation in the system.

(Peskins & Hart, 1996, p. 74)

To be able to assure the desired quality for our product, we implemented certain measures. In the following paragraphs we will explain how we utilized different activities and guidelines for this purpose. These include research, pair programming, version control, code standards, tests, and communication.

3.6.1 Research

To ensure that our product was correct and that the calculations were the right calculations we had to do some research into current solutions and projects done within the microgrid field. We looked

at different projects and the results of those projects, to see if we could learn from their experience in Microgrids.

In addition, we talked to local experts on how elements of solar and battery storage would be in the implementation to a microgrid in Kristiansand, both in function and cost. We talked to Oneco Solutions; a business located in Kristiansand that has a lot of experience with solar panels. We also talked to Mechatronics (MIL) in Grimstad to collect information for how a battery would work in a microgrid.

3.6.2 Pair programming

Pair programming is when two people code together. This means that one person writes the code, while the other looks over and helps check for logical errors and provide input on solutions. This method increases quality without a penalty to time. (Extremeprogramming, s.a.)

While at Agder Energi the backend team used pair programming in order to achieve fast progress while maintaining a great level of quality. It also ensured that they had the same understanding of the system.

After we were isolated at home, we used screen sharing to achieve pair programming. To work at the same workstation over the internet is possible, but not to the same extent. We have come to realize that when the observer (the person not controlling the machine), would bring much less input, due to increased sources of distraction; compared to face-to-face pair-programming, when you can see when your observer is doing something else or is not paying attention.

3.6.3 Version control

For version control in the project we used GitHub for the code, and Microsoft teams for other files. Version control is important to be able to backtrack if a version turns out to be ineffective. It also helps cooperation between team members, as everyone can have the same version without much hassle.

3.6.4 Code standards

When developing the system there have been several aspects in focus when it comes to code standards. We have tried to be very careful to avoid too much coupling and code duplication, however some has been left in due to difficulty with refactoring it. We feel that cohesion has been kept to a good standard. One way that we have ensured good code standards is that every time we make a new version, one or more people look over the code to make sure it follows all the criteria we set as good code.

Some of the criteria were that it should be easy to read, and all the functions and variables had good and descriptive names, as well as the code being well structured.

3.6.5 Tests

To ensure quality of the end product there are several things that should be considered. First of all, some description of what we considered was good quality had to be made. By looking at our user stories and the acceptance criteria we had there, we had a vision of what the end product should be able to do. If these criteria were met, we could say that our product held high quality.

In order to ascertain whether or not our user stories' acceptance criteria are feasible to maintain, we conducted acceptance tests of every user story when we marked them as complete. The status depends on the ability to perform and complete the acceptance tests. The tests were performed by running the program and performing the actions in the acceptance tests.

One example of a user story and how tests were conducted is regarding solar panels.

User story: As a user, I want to add solar panels to a grid in order to see what difference a solar panel does to the microgrid.
Priority and Importance: High. To see if a microgrid is cost efficient the program should show how power demand from the public grid is lowered.
Acceptance criteria: Be able to add solar panels in the microgrid.
Acceptance test: <ol style="list-style-type: none">1. Add solar panels to the grid2. The graph should show when and how much power the panels generate.3. Calculate the total power generated by solar panels.

We conducted the acceptance tests and made sure that everything worked as intended. When the test was finished, we marked the user story as complete and continued with the others.

This proved quite helpful as we could easily ascertain whether or not a function we intended to implement was implemented successfully. These tests were quick and simple to perform, and we were happy with the results.

In addition to the acceptance tests we manually tested each function we implemented into the system to ensure that they did what we expected them to do. We did this by manipulating the data directly and ensuring that the values we received in return were the ones we expected.

This manual testing, while a bit tedious at times, has proved very helpful. It allowed us to carefully select parts we thought could have errors and test them thoroughly. This has been of great benefit to us throughout the project.

A common way to test a system is to use unit-testing. However, we opted to not use this method. This is because we were developing a web-application and we struggled with how we would structure these tests in order to fully utilize them. Instead we were thorough in our manual testing and when running the acceptance tests. By conducting the tests, we can ensure that the system has good quality.

3.6.6 Communication

Having good and frequent communication was important to ensure quality of the product and to ensure that we did not put too many man hours in working on things that were not important.

An example of this is that we had little to no experience in electricity and the science behind solar panels. This led to several hours trying to figure out how much power was generated by solar panels every hour in different areas. After consulting our project manager hours later, we decided to use standard values taken from a solar farm in Lillesand which was given to us by Oneco Solutions.

The communication between the team has been very good, but like any team, communication can stagnate. This especially happens when someone has been working on a specific task for a while that they are not able to finish. This can sometimes lead to a passive and defensive communication which can lead to irritation between some of the team members. However, this often resolved itself over a few hours.

3.6.7 Journal

During the project, the scrum master wrote a journal that included the different discussions, assumptions, and choices that we made every day. This was a great tool that we used later in the project to remind ourselves of why we made certain decisions. It also helped us a lot in writing this report.

4 The product

Our final product is a working system that gives a good indication on how much microgrid solutions would impact a neighborhood and how this would affect costs. The product includes most of the functions that were requested by the stakeholders and is a web-based system that can be accessed from anywhere with valid IP addresses.

4.1 Assumptions

Before, and during the project, there were some assumptions we had to make to ensure that the program worked as intended in the limited time that we had on the project. These were made in collaboration with the product owner and helped us reach our goals. Here we list the different assumptions that we made:

- There is a system/AI that controls the entire microgrid
- Battery prices are calculated from values given by Pixii AS
- Solar panel prices are calculated from values given by Eidsiva Energy
- Batteries are set to 0.5C (C-rate) and main range from 10-90%
- Solar panel generation is based on numbers given by Oneco from a project in Lillesand, Aust Agder.
- Estimated time to charge car batteries are based on data given from one of the team members' personal electric car.
- Average mileage for an electric car per year in Norway is retrieved from ssb. (ssb, 2020)
- Estimated power consumption for heating is calculated to 300w per hour. (Kragerø Installasjon, s.a)

4.2 Video

A video of the product is attached to this report - see canvas. (Confidential - Not to be used for teaching purposes)

4.3 Deployment

To deliver the project in a safe and secure way it was decided that we would use Microsoft Azure to set up the server where the program was running. Agder Energi used this for other projects and it was decided that our project should be included here as well. Azure is a cloud computing platform that provides a range of cloud services, including compute, analytics, storage, and networking. (Rouse, 2020)

We had never used Azure before, and we used some time to understand how it should be set up and what solutions were best for our project. When we figured out the best way to use this for our project, it suited us very well. It had all the features that we needed and with some help from an IT manager at Agder Energi, we made a program that was reliable and safe.

4.4 Security

To ensure the safety of the program we realized we did not have the experience that was expected. We reached out to the IT department at Agder Energi as we wanted to make sure safety was prioritized and taken care of properly. In hindsight we are glad we made an inquiry to the IT department early to avoid spending too much time figuring out what solutions were the best for us. As mentioned in (programming choices), after a brief chat with the IT department we settled on adding an IP filter to the azure web application. This allowed us to pick which IP sources would be able to connect to and use our application. It was simple to achieve, and we deemed it good enough seeing as our system does not hold sensitive data.

4.5 Documentation

Documentation is very important in our case because we could use whatever kind of tool or technology to complete our project. If we chose to use a programming language or special technology in our project that Agder Energi is not familiar with, documentation is important when we deliver our project to them. Documentation can be divided into system and user documentation.

System documentation: A description of system requirements, architecture, and construction details. This is information for the developers and technical personnel who will maintain and maybe upgrade the program. (Satzinger et al., 2016, p 456)

User documentation: A description of how to interact with and use the system. Information for the end users of the system. It describes startup and shutdown, commands, and functions. (Satzinger et al., 2016, p 456)

At the end of the project we gathered all the documentation that we had and created both a system documentation and a user documentation.

For the system documentation we described how the system was built up and how to implement this on e.g. a new server. The methods and functions we had in the code were well explained through comments so that they could easily be understood by someone later. All of the code was added to a GitHub repository where IT staff at Agder Energi had access. As the IT department at Agder Energi was involved in the setup of Azure, we did not have to document too much on how this was set up.

User documentation was created to make it easy for the users to fully use the program. The documentation was created as a how-to guide that described step-by-step how to perform different actions within the program. This was formulated and illustrated in a way that ensured that anyone could be able to use the system without problems. This was made in PDF format that is accessible from the system.

5 Reflections

The purpose of this part is to discuss our experiences working with this project and what we have learned from it. We will reflect on three main topics. These are regarding methodology, project management, challenges and changes to the project and product.

5.1 Methodology and project management

We consider our choice of methodology a success. Both because the outcome of the project was as expected and that the project work went well.

We discussed if we would use any other methodologies. But we chose scrum since this suited our project well. The different steering committee meetings we had scheduled with the supervisor from UIA could also act as a sprint review for us. By having these meetings at the end of every sprint, we could review the sprint that we have had, show the changes to our system, and get the opinions from the project supervisor and project owner.

With Scrum we got a quick project start and got a lot of progress done in the earlier sprints due to following the framework well. Since we knew the framework from earlier projects, we did not have to spend a lot of time learning the different aspects of it. This may not have been the case if we chose to use another methodology.

An element of Scrum that helped us during the challenge of Covid-19 and working from home, was that there are roles in Scrum. This made it easier for us with the coordination of the team. This could have been more difficult if we had chosen a methodology that did not have such strict roles management.

5.2 Challenges

In a project such as this, challenges will inevitably arise. It is important that these are dealt with in a good and effective way to avoid setbacks. Below, we will discuss what challenges we experienced and how we dealt with them during our project.

5.2.1 Communication

During the span of the project we became accustomed to working together in the same room and having very open and fluid communication. This, of course, changed when we had to isolate due to Covid-19. After isolation we resorted to voice communication over the internet, which forced us to be better at communicating, but of course is more challenging than actually meeting in person. The communication with the product owner went from face to face communication to voice chat

and email. This was of course a bit more challenging, but we made sure we sent requests on email in good time, and scheduled voice meetings ahead of time to make sure everyone was available. One positive side of this was that we learned to be more structured in our meetings and plan them well.

5.2.2 Knowledge

When we first started working on the project, there was a lot of work to be done regarding the basics of electricity. This was some of a challenge for us as we had little to no experience on the topic and it was difficult to understand the different terms related to the topic. Since we divided our team into different groups, we had a team that could focus on learning more about power generation, power consumption etc., which worked well for this project.

5.3 Changes in the project and product

Throughout the project we did some changes due to change in requirements and scope or because we found a better way to solve a problem. Since we worked in an agile way, we were open for, and expected, changes.

5.3.1 Graph

To create a graph that showed how the power consumption was during a whole year for a transformer proved to be quite difficult for us in the beginning. We had never done anything like this before and after some research we figured out that creating this in JavaScript would be the best way. After working on the JavaScript graph for some time, with a lot of challenges, we saw that another project group was using a program called Grafana for a similar solution. We decided to research Grafana and found that this worked perfectly for our system as well.

If we had done more research on different ways of displaying a graph, we could probably have saved time early in the project. Our wish was to create the solution ourselves, but there are many solutions already created that could help different projects progress faster.

5.3.2 Changes to scope

In our planning phase, the team decided on the different components that one was supposed to be able to add to the microgrid. This included solar panels, batteries, wind turbines, car chargers and diesel generators. As the project developed, we figured out, in collaboration with the product owner, that adding wind turbines and diesel generators would be too much work compared to the effect they would have on the simulator. Adding these could have been done if we had more time, but it was decided that we should focus on making other functions better instead.

Getting hands-on experience with changes in an agile project has been very valuable for us. We learned that unexpected changes could happen anytime, even well into the project, and it is

important to be able to adapt. We realized that we could have done some changes sooner and others we could have avoided altogether as they had minimal impact.

5.3.3 How-To

This is the first project we have had where we created something for someone else. One thing that we did not think of in the beginning was creating documentation for the users of the system. In the fourth sprint this issue was addressed by the product owner and it was decided that we should create a how-to guide. This was added to the scope, and a task was added to the backlog. When creating systems for others we have to make sure they want to use the program, so we have to tell them how.

5.3.4 Django - Flask

When deciding what framework to use for our solution, we initially started working with Django. As none of us had used Django prior to this project, we decided to keep our eyes out for other possibilities as well. We noticed after a while that Django was a very large framework which had a lot of in-built classes and functions and realized that it would take a lot of time to familiarize ourselves with these. Therefore, we decided to look for another solution and, shortly after, we found Flask. This seemed like a promising framework as it was not as extensive as Django. This saved us some time even if we had to make the classes and functions ourselves. (Mindfire solutions, 2018)

6 Statement from Client

Beskrivelse:

Mikrogrid er et geografisk avgrenset elektrisk distribusjonsnett, som kan driftes helt eller delvis uten å være tilknyttet kraftnettet. Smart styring av forbrukslaster i et mikrogrid vil være en viktig bidragsyter i å løse de utfordringene kraftnettet står ovenfor med elektrifisering av samfunnet og overgangen fra fossil til fornybar energi. Oppgaven gikk ut på å lage en simulator for et mikrogrid, hvor man som bruker har mulighet til å simulere hvordan forbrukslaster vil endre seg ved å legge til ulike komponenter i mikrogridet. Hovedmålet med simulatoren er å se i hvilken grad bruk av batterier og solceller vil redusere behov for energi fra kraftnettet, og redusere belastningen på trafostasjonen. Slike løsninger vil gagne både nettselskaper og forbruker i form av reduserte kostander. Dette er et svært komplekst område, da man er nødt til å forstå flere ulike teknologier for å klare å lage en slik løsning.

Leveranse:

Gruppen har levert en fungerende prototype av en simulator som i aller høyeste grad tilfredsstiller våre krav og ønsker. De viktigste funksjoner og aspekter ved løsningen:

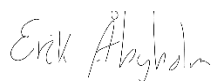
- Mulighet til å legge til et batteri av egendefinert størrelse
- Mulighet til å legge til egendefinert antall solceller
- Automatisk utregning av reduserte kostnader det medfører
- Løsningen er integrert i Agder Energi sin IKT struktur, og tilfredsstiller de sikkerhetskrav som stilles.
- Intuitivt og brukervennlig grensesnitt
- Grafisk fremstilling av forbruk og endringer

Agder Energi er meget fornøyd med leveransen, og er imponert over hva studentene har fått til. Energi og teknologien er et komplekst område, hvor studentene har vært nødt til å forstå hvordan det fungerer for å lage en slik simulator. De har selv satt seg inn i hvordan kraftnettet, batterier og solceller fungerer, og anvendt denne kunnskapen i utvikling av løsningen og de analysene som gjøres i den.

Prosjektarbeidet

Studentene har gjennom dette arbeid vist at de evner å jobbe selvstendig og tar selv ansvar for fremgang. De har selv innhente mye kunnskap, og flinke til å si ifra hvis de har behov for bistand. Prosjektarbeidet har vært preget av struktur og profesjonalitet, men tydelige rollefordelinger innad i gruppen. Det har ført til at kommunikasjon mellom Agder Energi og grupper svært enkel.

Agder Energi ønsker studentene lykke til videre i studier og arbeidsliv, og håper på flere slike studentgrupper i fremtiden



21.05.2020

Erik Åbyholm (Det måtte bli med digital underskriv pga. hjemmekontor)

Forretningsutvikler Agder Energi

7 Self-evaluation

Evaluating our teamwork has been an ongoing process through the whole project. There are several things that we have learned that we will bring into future projects.

7.1 Group

We are satisfied with the way we divided the team, and everybody feels that they have contributed to the project. As we have different skills, this was a perfect way for us to utilize every member. Seeing as we worked so closely, we also got involved in each other's tasks and got the help we needed inside the team.

We as a group have been working together for the last 3 years, so a lot of the problems that arise during project work have been discovered in earlier projects and been fixed and we are always in the process of making our group work more effective.

One of the biggest changes that impacted our work routines was that we could not work together in person but had to chat over discord from different locations. This was a big change for us, and it took some time to get back to the good efficiency that we had earlier. Following the scrum framework helped us a lot with the challenges since we had daily meetings, reviews, and a backlog to follow. Even though we were three different teams, everyone was up to date on the project at all times.

Working with a scrum master made sure that communication with our product owner, external companies that we needed information from, and within the team was streamlined. Everyone was updated and had the information that they needed at all times.

7.2 Members

Jarle N. Johnsen

In this project I have acted as the Scrum master for the team. This has been a very exciting task that I have learned a lot from. I have learned even more about the elements of Scrum and learned a lot about planning. To make sure the project went well I had to plan the different sprints, meetings with internal and external resources and how to achieve the information that we needed. I made sure the team had the information they needed to do their work and gathered this by researching the topic or talking to people/companies that had knowledge about the subject. I helped the front-end team with ideas and design decisions in the program and the back-end with calculations on e.g. solar data, car charger data, etc. I had to make sure that everyone knew where we were in the project and kept them updated. I also had contact with our project owner and discussed changes

and informed him of the choices we made. Looking back at the project I am satisfied with my contribution and I am proud of what we managed to develop.

Christian Kittilsen

In the beginning of the project, I was assigned to the back-end team. We decided to write the program in Python. My group members learned some Python the previous semester, but I was exchanging to Korea at the time, so I had to learn it from scratch. Because of this I had more of a supporting role in the back-end team while Daniel was in charge. Later, I also helped with problem solving and decision making on other parts of the project as well. About halfway through the project, most of my efforts went to writing on the project report, but I still helped with the back-end whenever my help was needed. I have had a lot of fun with this project and I have enjoyed working for Agder Energi. It has been very interesting to work on something more significant and realistic than our usual assignments. While working on this project, I acquired new knowledge and learned new skills that I will bring with me to further studies and work. These include how to manage a project and what challenges may arise, how important teamwork and communication is in a team environment, and how to write code in Python.

Jørgen B. Lefdal

In the project I mostly worked as a front-end developer for our system. We started with the planning of the system where I made the prototypes with input from the others while they gathered data on how much batteries and solar panels. After we distributed our group into more distinctive work areas, I worked with the front-end. When the overall baseline interface for the system was agreed upon as a group and the project owner was satisfied, I made the transition into working with the animation. The animation explains how a microgrid works simply for the user. The animation was more intensive and took longer than I first assumed but was interesting to learn about. I had never done anything of that kind before, but the task was very interesting and aggravating at the same time, the learning curve was steep. I wanted to work more with the back-end during this project, but as a group it would have been wasteful to assign more people to work back-end and we had to prioritize to work with the group's strengths to ensure that we would achieve the absolute best result. Looking back at the project, I am proud of what I contributed to the group, and I am satisfied with the product that we produced.

Daniel H. Lindemann

In this project I have acted as the main full-stack developer of our system, with my main focus on back-end, and helping the front-end team when they needed it. During this process I have gained a lot of knowledge about different techniques and technologies for developing applications. I have also contributed to other aspects of the project, such as helping gather information on different aspects of microgrid components or helping with project management. It has been a very

educational process, and I got valuable insight into how cloud solutions like Azure work. It has also been a pleasure working with Agder Energi during this project and getting some insight into how projects are done in the business world. I have during this project learned that I enjoy project work in general, but mainly programming and project management.

Daniel T. Reisæter

For the project, my function was being in the management team, which meant a part of the information collecting, and meetings with other businesses regarding information together with the Scrum master. In addition, I was also in charge of keeping the backlog up to date, as well as managing hours on all the tasks that were in the backlog. I was also assisting the front-end team when there were no new project planning tasks to be done. I assisted in this by either finding information about how we wanted to create the web app visually, as well as how it can be done from a technical point of view. During the project, I have learned how it is to be a part of a bigger project, especially the planning part of it. I enjoy the project planning part of a project more than the programming part, as I feel like I have more control on this aspect of a project, as well as that I find programming harder than management.

Thomas Theissen

During this project I have learned much and received a lot of new experiences when it comes to project work. During our bachelor project I have learned how projects work in bigger companies. I have definitely learned a lot that I will take with me in my future job career. I have mainly been working as a front-end developer in the project. All the way from the early planning phase where I created a low fidelity prototype with a pen and paper and later creating a high fidelity prototype using UXPin to creating the front-end part of the website using HTML, CSS and some JavaScript. I wish we started using react for the front-end development, because the learning outcome of using react would be very nice to have. I have also been working on risk management in the project, which I am very happy with. I have gotten a bigger insight of how risks can impact a project.

8 References

Agder Energi, (s.a.) Vår virksomhet. Collected January 29th from: <https://www.ae.no/var-virksomhet/>

Agder Energi. (2020, February 17). Our owners. Collected from <https://www.ae.no/en/about-agder-energi/organisation-and-management/our-owners/>

Agilebusiness.org. (04/10-2018) MoSCoW prioritization. Collected from: <https://www.agilebusiness.org/content/moscow-prioritisation>

Berkley lab (s.a). Microgrid Definitions. Collected February 6th from: <https://building-microgrid.lbl.gov/microgrid-definitions>

Evens, J.(2014) Scrum Revealed. United States of America.
International Scrum InstituteTM

SCRUMstudy. (2013, December 16) Time-boxing in Scrum. Collected from: <http://blog.scrumstudy.com/time-boxing-in-scrum/>

Extremeprogramming (s.a) Pair programming. Collected 04.05.20 from <http://www.extremeprogramming.org/rules/pair.html>

Kragerø Installasjon (s.a) Hvordan kan jeg redusere strømforbruket? Collected 05.05.20 from: <https://www.xn--kragerinstallasjon-l4b.no/enok/>

Lantero, A. (2014, June 17). How microgrids work. Collected from: <https://www.energy.gov/articles/how-microgrids-work>

Mindfire solutions.(2018, May 3). Flask vs Django. Collected from: <http://www.mindfiresolutions.com/blog/2018/05/flask-vs-django/>

Peskin, M.I. and Hart, J.J. (1996), *Measuring the quality of computer systems development*, Benchmarking for Quality Management & Technology. pp. 68-84.

Rongala, A. (2015, March 12). Benefits of Python over Other

Programming Languages. Collected from:

<https://www.invensis.net/blog/it/benefits-of-python-over-other-programming-languages/>

Rouse, M. (2020, April) Microsoft Azure. Collected from:

<https://searchcloudcomputing.techtarget.com/definition/Windows-Azure>

Scrumguides. (2017). The scrum guide. Collected from: <https://www.scrumguides.org/scrum-guide.html>

Satzinger, J., Jackson, R & Burd, S. (2015) System analysis and design in a changing world. United States of America. Cengage Learning

SSB (2020, March 24) Kjørelenger. Collected from:

<https://www.ssb.no/transport-og-reiseliv/statistikker/klreg>

Trello (2017, October 16) What is Trello? Collected from:

<https://help.trello.com/article/708-what-is-trello>

Uia, (2020) Bacheloroppgave i informasjonssystemer. Collected 13.05.2020 from

<https://www.uia.no/studieplaner/topic/IS-304-1>

Verwijs, C. (2017) 10 powerful strategies for breaking down Product Backlog Items in Scrum.

Collected from: <https://medium.com/the-liberators/10-powerful-strategies-for-breaking-down-user-stories-in-scrum-with-cheatsheet-2cd9aac7d0eb>

W3org (s.a) HTML & CSS. Collected 01.05.20 from:

<https://www.w3.org/standards/webdesign/htmlcss.html>