P1 Processanalysis

- B219 -

Aalborg University
Institute of Computer Science

Contents

Chapter 1

Process towards problem issue

1.1 Problem statement

"How do we ensure with a software based solution, that the citizens will be more inclined to use the cars that are already there, thus slowing down the increase of cars in Aalborg, leading to a decrease in car production?"

1.2 Summary of the project

At the beginning of our project, we started mind mapping our chosen topic. This was done through an hv-diagram (5W1H) twice. By using this method, we ensured that all group members were adjusted to the same wavelength so to say, meaning that everyone had acquired the same knowledge in the topic, and we were ready to move on. It also created several questions that was to be researched on, so that our knowledge of the area could develop further[2]. This worked pretty well, as the group got a lot done in a short time. Though at the supervisor meeting, we were told that the topics we had researched on were to broad, and it must be narrowed down. This was a great meeting and eyeopener, as it helped us a lot in specifying certain topics and paving a way that narrowed down to a proper problem statement. Afterwards, we did a lot of research on the SDG 12, cars, CO_2 emission and the connection between them. This helped a lot with expanding our knowledge in the area, leading to a better understanding of where a specific problem is located, that would be solvable. From this data, we made a prototype of a problem statement:

> "How do we prevent citizens of Aalborg buying more cars, with utilizing the cars we already have?"

After this, we made a stakeholder analysis to properly detect and visualize the people that this project will impact and be impacted by[4]. After that, we created a survey to see whether our concept of a solution would actually solve our problem or not, which we received positive and validating feedback on. It was also concluded from the survey that our problem at hand was both authentic and scientific, because the problem was an actual problem in the real world, while also being understandable and solvable[1]. The problem statement was still too broad, as it did not include anything software based, and the problem had a lot of very different approaches. We continued to research, and came up with the final problem statement as shown earlier. The problem analysis still had plenty of holes and the connections between the topics were not that great, but we decided to move on for now and start the creation of a program. The program requirements were discussed in a physical meeting, which took some time because picking out the most important features proved to be slightly subjective, though eventually we were done and all agreed. The creation of the features were distributed amongst pairs of two in the group and we set a deadline. Working pairs proved efficient, because there was not too many inputs from several group members and it was easier to coordinate working times. The deadline was successfully followed and we held a "merge-Marathon day" which was where we put all of our coding into a single program and connected it. As the program was then done and running, we finished up on the program design and implementation. When that was done, we filled in all the holes in the problem analysis, so that it looked like a connecting and recursive process. This was about the entire process of the report boiled down to a couple of lines, and of course there was a lot of minor issues and problems that were not highlighted here, but this is definitely the most important aspects of the process.

1.3   Time management and the working styles

The group work mainly took place on discord, which is described in the tools section. This is because we all agreed, that we work best from home, since the working environment with a desktop computer and multiple screens allowed for a better overview and coordination. The group work also took place at the university a couple of times, mainly because the schedule required us to be there for some other reason, like a lecture. We also met physically when arranging bigger parts of the project, like the work delegation for the problem analysis, program requirements and other big parts. Speaking of the working hours, the general goal was full time, including other lectures from the university. This means that if we had 4 hours of lectures and problem solving, we had 4 hours for the project afterwards that day. This was to prevent overwork and tiredness. We believed that if we work too much, the quality would decrease and we would be not motivated to work as

efficiently as wanted. Another aspect was also the different circadian rhythms of the group members. Because there was a huge difference in when people were efficient at working, we tried to make as flexible working hours as possible. This meant that people work on their part of the project at different times of the day. The deadline would be the same for everyone, but when the group member did their work, it was done individually. This worked out quite well, as the deadlines were held and the quality was beyond acceptable most of the time. It reached a downside in the near ending of the project, as some group members began to not show up and not do their work properly, which proved very problematic. This behaviour was not met with a hostile attitude by the rest of the group, but with a curious mind, asking the group member if anything was going on that they wanted to share. We believed that there is no point getting angry when expectations were not met, as there was most likely a good reason for it. As to why this kind of working structure worked the way it did, we believe it was more a question of the motivation in the individual group member than it was about a group based motivation. This had both its ups and downs as explained. As a general lesson of this way of working, we concluded that it would be better to work in the same hours of the day, adding more physical meetings so that everyone stays motivated and on time.

1.4 Knowledge sharing and learning styles

Learning in a group setting is difficult, but very rewarding. Passing on knowledge to other members of the group is not only beneficial to the recipients, but for the individual sharing their knowledge as well. The rewards from sharing information is a deeper understanding of the topic at hand, as you have to be able to explain complex concepts in a simplified way. Another benefit of learning as a group is minimizing the time required to learn, as each individual can learn about different topics and then share the information. This drastically cuts down the amount of work each person has to do. Most of the knowledge we share with each other in our group is explicit knowledge. This is done through links to articles, charts and reading each others work. The quiet knowledge we share is through discussions and sparring with the group as a whole and with the partner in our specific assignment.

When it came to using the learning styles in our project we believed in the midway process analysis, that:

> "the differing learning styles can each tribute to a better structure for the group. When work is being distributed, the tasks can be given to the individual who has the most fitting learning style. This is also true for partner tasks, where both of the individuals involved are compatible

> with each other's learning style. Different learning styles in a group
> will also contribute to more balanced product, as each aspect of a task
> is dealt with by someone with the appropriate approach."

This did not prove to be something useful, as putting the group members in boxes seemed to be very hard and unnatural. Therefore we took a more natural approach, assigning a group member to a topic by making decisions as a group. We discussed who would like to do what, which naturally just found its way. This may subconsciously have been influenced by the individual learning style, but this was not taken in consideration.

1.5   Conflict management

The group agreed as also explained earlier that when something is not as it should be, whether it be an individual or a group wide issue, the problem is met with patience and understanding, instead of hostility and criticism. This ensures that the group and its members were also more inclined to tell, if something was going on or something was not fitting for a member.

1.6   Tools

Because of the flexible working hours we chose to work with, we picked monday as a good application. By using Monday, we made a list of all work that was needed to be done in specific deadlines. This proved to be a good choice, because the group could see who was working with what and how far they were. It also worked as a motivator to get the work done properly and on time. In conjunction with monday, google calendar was used to mark supervisor meetings and non-university related activities, for example secondary working hours or a dentist appointment for a group member. Discord and messenger were used as the main communication tools for day to day coordination, knowledge sharing and general questions, help and social activities. Discord was mainly used when knowledge sharing and doing university related tasks, because of the screen sharing feature and creation of different text channels, so that different topics had its own channel. Messenger was used more when dealing with short term conversations and social life. It was also to check up on each other, to ensure everyone was up to date with the schedule. We used Miro to visualize our ideas and thoughts. This included 5 H's and 1 w diagrams, concept maps, program design and flowcharts. It worked very well as a coordinator, so that all group members had the same vision of what was to be done and how it should work. This application will definitely be recommended for future use, as it is also free and great for various visualization purposes.

Microsoft teams was used for supervisor meetings exclusively. It worked very well, except for technical difficulties here and there, but nothing significant.

GitHub was used to properly coordinate the different assignments that were given amongst the group members. GitHub is an amazing tool to keep track of changes, merging the different branches of code and in general a must have when coding as more than one person. It would be a pain to send everything between each other through discord or messenger every time there is an update, so GitHub really helped in making the coding process flow.

Chapter 2

Supervisor work

We made a supervisor contract document, which aimed to make sure that the supervisor and our own view on the cooperation were in agreement. The supervisor contract contained the different elements of our cooperation with our supervisor, these elements include meetings, communication and documents. We expected everyone in the group to participate in the supervisor meetings. During our supervisor meetings one of us were to take notes and made sure that these notes were circulated in the group after the meeting. We furthermore had an agreement that one of the group members acted as a moderator, who made sure that the meeting followed the agenda for the meeting. The agenda for the next supervisor meeting should at the latest be sent a day before the supervisor meeting. The communication with our supervisor should happen over mail, and they should contain our group ID and be sent cc to the entire group. Emails should be replied to within 24 hours during weekdays, this counts for both the group and the supervisor. All of the groups documents such as worksheets or reports would contain the group ID in the filename and within the document itself. The cooperation with our supervisor went as intended, with weekly meetings, be it physical or online. When we reached the programming phase, online meetings were preferred as screen sharing was very convenient for explaining the features. These meetings mostly went as planned, with a member of the grouping taking notes and the agenda being send out the day before. The only part that consistently did not go as planned was that every member of the group would actively participate in the meetings. At the meetings, especially online ones, there was usually only 1 to 2 active members of the group. Apart from this, our group and the supervisor have followed the agreements in our supervisor contract. The communication with our supervisor have followed the agreements made in our supervisor contract, and our supervisor has been willing to answer any of our questions concerning our project. There has been no problems concerning our supervisor, as the supervisor have followed the expectations mentioned in the supervisor contract.

Chapter 3

Project Management

Code management

Code management was a high priority for the whole group, as this was something that that we did not want to have any issues with. One of the first things we did when we started the coding process was to learn Git and GitHub. Git is a software for tracking changes in any set of files, which means it is a version control system. GitHub us a provider of internet hosting for software development and version control using Git. It offers the distributed version control and source code management functionality of Git. It provides access control and several collaboration features such as bug tracking, feature requests, task management, continuous interactions.

We started off by discussing the program requirements, which features should there be and so on. Once these features had been discussed, we discussed the design of the program, this included the flowchart of the program. As soon as both the program requirements and program design had been settled, the group discussed who should code what. The group agreed that every larger subject of the program would have 2 group members responsible for it, and in these larger subjects there would be smaller subject to be programmed. This worked well since you would have a sparring partner to help with the programming and if you had any issues.

We made a repository on GitHub called "P1-project" that every one in the group had access to. In this repository we made a rule on the branch called "main", that would contain the main program which would be the final product, that no matter what could we commit a change to the main branch. This means that we would have to make another branch and work on that, and whenever that was good, 2 group members would have to approve the code before it could be pulled together with the main branch. This is simply to prevent having wrongly made code in the main file. If this would happen whenever anyone makes a new branch that code

will be wrongly made in there, which would ultimately end up in every group member having a bug in their program.

Git combined with GitHub has enabled us to create this projects code without any major flaws. This type of code management worked incredibly great for this group, and is definitely something we will bring in the future.

Time management

During this project we used Monday as a time scheduling platform, it was a good platform to control that we uphold the deadlines set by all of us. It was also used as a tool for delegation, because we could put in the problem analysis as a main item, then we could add sub items and then delegate each subsection to members of the group. This setup can be seen in figures 6.1, 6.2 and 6.3 in the appendix. Throughout the whole process the members of the group could mark whether they was 'started' on the item they should do, or they were working on it or if they were done with it. This helped a lot since everybody could log in to Monday and see what the status was with the all the sub items, that way those who were done with the assignment could offer their help to the rest of the group. We have been using the SCRUM method as our time schedule and the website we have been using is Monday. We have also been using Google Calendar, but that is for private stuff like if you know that there is some day you are unable to come to school or group work we put our private appointments in the calendar so all the members of the group will be able to see it. Although the tasks were assigned dates for the most part, the actual deadlines were usually orally agreed upon and a bit flexible.

The actual scheduling

As for the actual scheduling in relation to the deadlines we had on our report and our code, we did not finish on the due to date, since we had more to do than we realised. We were using the SCRUM method and the website Monday.com which had a 30-day trial where we had the Gantt chart, but since we did not pay for it we can only use what is seen in the pictures at the start. We have been trying to prioritize some of the subjects to form a kind of Gantt chart in the main table of our time schedule. Though using the SCRUM method we agreed to have SCRUM meetings once a week, but we did not hold a single SCRUM meeting

Chapter 4

Meta reflection

As a different group than the one from P0, we have different experiences and abilities that we acquired during P0, which has helped our project and our way of thinking and cooperating for P1. We have made good use HV models and concept maps now that we have a better understanding of these methods through P0. This also means that the project has not felt as overwhelming as P0.

The group contract is also different from P0, now that we know the major aspects of teamwork, but also the more subtle aspects of cooperation. As for the major things, it would be scheduling and meeting times, but for the more subtle things, it would aspects as knowledge sharing and social life. This ensures that we all can talk comfortably and that we can have an academic discussion to broaden the horizons of the entire group.

An issue that has been in the group since beginning was missing members of the team, which was a problem, but was solved more subtle. The missing group member is still required to be able to solve the problems that come up in the P1 process and other topics for that sake. This also makes sure that even though the group member has been missing for a particular session, the group member is still as competent as any other, and does not have any holes in their ability to contribute and work.

Our group work has worked well, thanks to good task delegation through Monday, discord or physical meetings. Using GitHub and Overleaf also made it easy for everyone to follow the creation of the program and report, so that the members working on the report still knew what was happening to the program and vice versa. As for the supervisor, we also have acquired a better understanding of how to use this help. The supervisor is very good to have when the team is in doubt of how something should be handled, which could be everything from the project structure to general terms and choice of words. The supervisor has also been very good at bringing forth issues the group itself did not see, which has been very useful in reflecting on our topic and process of our project.

Chapter 5

Conclusion

One thing that for sure should be kept and brought with us to our next project is our code management. This worked incredibly well in the group, and is definitely something that the group reflects on when talking about the things in the group that went well.

The group contract is something that should be extended. Since the groups we came from in P0 did not have any kind of issues with the group members, the conflict management on the group contract was not that comprehensive. The thing we wrote if we had any conflicts with the group was that "everybody should be friends". This is of course not optimal, and is something that should be changed the next time. The group needed some clear guidelines for what to do if there was any conflicts that should be handled. Here we could have taken use of the conflict stairs (konflikttrappen)[3]. The aim would always be to stay on the lowest level. Even though the group did not have any conflicts that was brought up, it was not be a bad idea to have it on the group contract.

Another thing that was too broad and loose in the group was the consequence of being late. The consequence was to bring cake if you did not arrive on time without any notice to the groups members. Reflecting over the P1-period this was not an optimal solution. The first couple of times the respective group members did bring cake, but it kept repeating without the consequence being upheld. This is of course because bringing cake is not a harsh enough consequence, and being late would in the end be perceived as a normal thing than something that should be done, and could be overcome by just bringing cake to the group members. All in all, a prevention or consequence system should have been added, so that group members are motivated to stay on time, instead of it being tedious.

In terms of regulations, the amount of individual working periods might have been too much. There have been times where people did not check their inbox whether it be mail, messenger or discord which lead to a misunderstanding in the group. This would be in terms of whether we were working from home or not or

if we even were to meet at all. Including more meetings as a group might have motivated the group members to check their inboxes more frequently. That way we could also have done more check ups and have an asking round, where people could ask the others about issues they have with current work. A solution could also have been implementing/taking use of scrum, so that we would coordinate on at least a weekly basis if not daily, which also would have allowed for announcing meetings and deadlines and assuring everyone heard it.

Bibliography

[1] Lasse Krejberg Møller Casper Knudsen. *Problemdesign*. `https://www.moodle.aau.dk/pluginfile.php/2389091/mod_resource/content/1/Problemdesign%20workshop%20-%20slides%20-%20PBL-Dat-sw-dv.pdf`.

[2] Anna Markman Henrik Worm Routhe Sofie Otto. *PBL*. `https://www.moodle.aau.dk/pluginfile.php/2367472/mod_resource/content/1/PBL%202%20PPT.pdf`.

[3] Henrik Worm Routhe. *Projektdesign*. `https://www.moodle.aau.dk/pluginfile.php/2328969/mod_resource/content/5/5%20-%20Projektdesign%20SWDAT%202021.pdf`.

[4] Systime. *4.2 Interessantanalyse*. `https://projektarbejdet.systime.dk/index.php?id=185`.

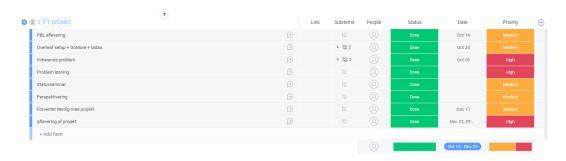# Chapter 6

Appendix

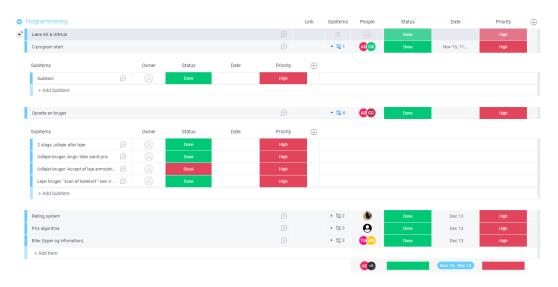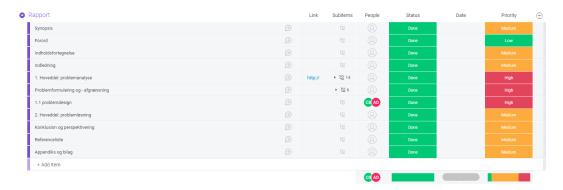**Figure 6.1:** Main table of our Monday.

Figure 6.2: Programming table.

Figure 6.3: What main subjects should be done in our rapport.