Reflection Report

“An assessment of what is in relation to what might or should be and includes feedback designed to reduce the gap”

Group name: **ABSOLUT**

Group Members

- Viktor Albihn
- Betina Andersson
- Christoffer Olsson
- Daniel Rydén
- Emir Zivcic
- Fressia Merino Espinosa
- Gustaf Järgren
- Jan Rasmussen
- Joakim Willard
- Johan Martinson
- Rikard Hellberg
- Shahad Naji

Table of Contents

1.0 Application of scrum

1.1 Roles, teamwork and social contract (relates to D1B)

The whole group had a meeting the very first week to talk about the social contract, the plan for the project and the roles within the group. We picked a scrum master who we saw fitting for the role. This person was able to motivate us and was good at speaking up and helping to raise other people's opinions. Most importantly, we felt that this person was responsible enough to attend all the scrum of scrum meetings, and we all trusted he would communicate our interests with his best efforts.

We had a solid foundation for setting up a good environment to cooperate and build a strong teamwork in. The subgroups knew beforehand what they themselves were capable of and what to expect from each other and thus arguments and dissatisfaction between members were rare to come across. Daily meetings where the majority of members showed up set us up for a good discourse and frequent feedback and no voice was silenced. The social contract we all participated in bringing forth seemed like icing on the cake. As a result, problems were quickly and effectively solved.

One thing we struggled with was that specific information stayed within smaller subgroups instead of being shared with the entire group. This made it hard to know what parts of the project had already been started or completed, as well as knowing what our most urgent problems were since everyone did not know what the others were working on.

In order to do better in the future, we find it can be helpful to pair people up with the ones you usually do not work with. This would make it more natural to spread information outside the subgroups and would hopefully achieve a greater understanding of each other's positions throughout the sprints which would result in less wandering about. However, since the solid subgroups allowed us to achieve a good workflow, this solution was not considered during the actual project.

In the beginning of the process we found that a very small group of members knew a lot about our process, such as how to test on the hardware, setup the server, start the camera etc. So when these people could not attend the meetings, our process stopped. We thought about our group’s truck factor, and realised we did not want to have to depend on just one or two members of the group in order to continue working. Hence, we made an effort to make ourselves more independent of others and in our last sprints we did not mind anyone being late since we could still work on the project without them. We make it a lesson for future projects to make sure not to be dependant of just one “group hero”.

1.2 Used practices (pair programming, stand-up meetings, etc.)

During the project there were daily meetups scheduled from 10-17. Since the group members had different schedules, people attended at different hours. This made it hard to have meetings with all members present, but made it easier for everyone to work a few hours daily. After the demo sessions on Thursdays, all members would sit together and we seized this opportunity to plan the upcoming sprint. This has worked for most of the time.

We would often write code in pairs because we have, during our previous courses, concluded that pair programming works best for us. Therefore, solving problems became easier and the work was more efficient. When we came across a complication, either with our code or the MOPEDs’ hardware, we would first try to ask others in the group for help. Secondly, if no solution could be found within the group we would ask other groups how they solved the problem. Lastly, we utilised tech support to try to get help. However with problems regarding sprint planning, we would discuss it with Jan-Philipp Steghöfer and Håkan Burden. Communication in between almost all involved was made effortless by the given forum of Slack channels.

Unfortunately there were also sprints where very few tasks were completed. A reason for this is that we struggled with the MOPED’s pre-installed software, we were unsure of how we were supposed to continue the work and this slowed us down a bit. By not completing tasks, and getting new tasks that we were not able to slice in a good way, the group became less productive. However, as a result we had to find new ways to work which led us to writing mockups, and make a plan for how we would navigate the MOPED. These things did not completely rely on the MOPED’s functionality. When the sprints did not go our way, this still gave us lot of insight on how to move forward when we came across a problem.

We planned our sprints using Trello, which has been a very helpful tool. We can definitely see that our sprint planning skills have improved the more we worked with it, especially when it comes to creating better tasks that are of value. The improvements we've made concerning tasks is to make them independent of each other and more manageable by lowering the complexity of individual tasks.

By doing this we were trying to reach the following goals for each backlog.

- Anyone can pick a task from trello and work on it without having to complete another task before it.
- Each task brings us closer to fulfill the user stories.
- Each member can see exactly what has to be done when picking a task, which can be done by describing a task in greater detail.
- As a consequence of independent tasks we could almost completely avoid a waterfall scenario where the different subgroups had to wait for another subgroup to complete their task before moving on.

It has helped to think of the INVEST-criteria when creating tasks to make sure they were of value to the stakeholders.

1.3 Time distribution (person / role / tasks etc.)

We used one week long sprints that started and ended on Thursdays. After the demo sessions for the product owner, and a small reflection and support session with our supervisors we started a new sprint. A few members sat down and wrote the tasks for the upcoming sprint. After this, all members could look at the tasks and pick whichever they found interesting. The initial thought was that people could pick tasks they were interested in and thus be more motivated to work and complete the task. While it was well-intentioned, this did not work for us. We ended up with several uncompleted tasks each sprint. Since all members were not participating when writing the tasks, it made it hard for the ones who were to come up with something of value that was interesting for everyone. Furthermore, sometimes people grouped up at very small tasks, whereas the bigger tasks had perhaps one to two people working on it which created an imbalance.

It would perhaps have been more optimal if we would sit down as early as possible in the new sprints and make sure each member gets a task they think they can finish. This would also make it easier to divide the workload equally and really get everyone involved in the project. Once again, a scrum meeting in the beginning of each session is something we could have really benefited of.

1.4 Effort and velocity and task breakdown

We were able to estimate satisfactory values for velocity as well as effort. The LEGO exercise assisted us with estimating better values that were more in accordance with what was realistically possible. However, the sub optimal source code on the hardware hampered our efforts in successfully breaking down the tasks. Clarity in the description of the tasks is of great value considering how effortlessly it can be achieved. We consider it maybe the greatest ROI time-wise to break down the tasks as much as possible and with a clear description.

We got a lot of help and tips from both product owner and supervisors on how to continue moving forward during the time when we felt stuck. It shows that it is important to have a good support system and to ask for help when you have trouble.

1.5 Overall reflection

It was hard to apply scrum to our work process in the beginning, but as the sprints passed by it got easier and scrum has been very helpful. It has been comforting knowing what has to be done, and breaking the task up into very small sections makes it easier to comprehend. It is important that all members are on board and want to work with scrum. Either that or you need someone who almost forces the others to work with scrum. Otherwise it gets hard to implement this framework.

The group worked well and there was not too many disputes. It was good to start with a social contract, this way everyone knew what was expected of them. Also, the fact that all of us took the time and made an effort to meet up every single day, despite having different schedules, made the teamwork easier for everyone. It was also obvious that each team member was willing to contribute.

What we can work on for the future is to make a real effort into creating better tasks, since the lack of these could often make us feel a bit lost in the process. Another thing we think is good to remember for future projects is to have a scrum meeting in the beginning of each work session. We find this can be a good way to get an overall picture of where we are in the process, and it can also be a good opportunity for people to ask for help if they realise they aren’t up to speed.

2. Documentation of sprint retrospectives, 0-1p

2.1 Sprint One:

For sprint one we managed to complete all the tasks we had set up. It took some time before we got started, so even though we completed the tasks, it felt a bit unorganized and stressful. We felt like it was hard to divide the user stories in the product backlog into good tasks. The user stories were a bit vague, and we struggled with writing concrete tasks that we felt confident working with.

For the next sprint we should create more concrete tasks so that all group members feel as if they know what that task is about. We should also divide these tasks among all members.

One negative outcome from having too many meetings is the fact that they turn less efficient. To solve this we need to have more concrete tasks in order to increase our productivity. In order to achieve this we want to have daily scrum meetings that take half an hour tops.

2.2 Sprint Two:

Sprint 2 went better than sprint 1, even though there is room for a lot of improvement. We realised that we have a hard time appreciating the effort of the tasks, so we decided to create an excel table where each member fills in how much time they have actually spent on each task per day.

Some people in the group have a heavier workload than the rest, and we want to spread the work more evenly. This could be done at our daily scrum meetings, which we also have to be better at implementing into our working system. We still have to be better at creating more specific tasks since a lot of them are still very abstract.

2.3 Sprint Three:

In sprint 3 we finished the app by adding cruise control. Almost all members have written a plugin which we will test as soon as the MOPED starts working - there is a problem with this. There have been improvements on the server and we have done the logic for speed and brake distance.

We have also created an excel document in which all members can fill in how much time they put on each task at every working session. The biggest priority for the next sprint is to find out what is wrong with the MOPED and fix it so we can test our plugins. We also want to find out how the camera works, what it sends to the server and how we can use this to implement the latitudinal function of the MOPED.

2.4 Sprint Four:

We were not very productive as a group during sprint 4. We had several tasks from sprint 3 that weren't done because we had problems with our MOPED. So we could not test our plugins. During sprint 4 some of us started writing mockups to try to test our plugins. And the others tried to come up with ideas on how to implement latitude driving on the MOPED. Later in the sprint we went on to write code directly to the MOPED instead of writing plugins. So we had to archive or delete many of our tasks. Our estimated total effort for sprint 4 was 167, and our actual total effort was 93, that because we archived a lot of our tasks, and also because we had task-debts that were half done, but we counted their full effort?

2.5 Sprint Five:

We have implemented the adaptive cruise control in the app. We have also written some sort of PID-regulator for the acc. This has not worked, but we now have information on how to fix all the problems that came up.
We have also made progress on the camera. We can now read in all the pictures that the camera takes, analyze them and decide if the MOPED should turn left, right or drive straight. At first we analyzed different colors at opposite sides of the picture, but we realised that the camera did not see the green color as good as the red. We changed this so we now analyze the left side of the picture where we look for red pixels, and then the right side of the picture for red pixels, and then we compare this. This has proved to generate more reliable values. We now have to implement the steering on the MOPED so it reacts the way we want it to.

During sprint 5 we have also continued writing on the report. We feel as if we will be able to get a semi-working prototype. There is mostly debugging left. We are also better at writing tasks, everyone feel like they are involved in the project and we have a team-building activity planned.

2.6 Sprint Six:

Most of the sprint week was spent improving the ACC code, changing the regulator equation and modifying the values of the constants to get the sought after behaviour. We have fixed and debugged the ACC. We implemented a picture analyzing method that split the pictures taken by the camera in half by counting the pixels on the left side and right side individually. This method used a red A4 paper that was attached to the back of the MOPED that was supposed to be followed. The solution iterated through the pixels and incremented a counter for a sufficiently red pixel. The higher counter signaled how the MOPED should turn. The code included if statements that checked if the red pixel count was overwhelmingly bigger on one side than the other, this would in such case tell the MOPED to turn more towards the side that had a larger count.

We are very proud (and a little surprised) that we were able to deliver the finished product we had intended. The MOPED was able to follow the MOPED in front of it, adjust the speed and follow when the MOPED was turning etc. If we could do more on our project we would like to improve user friendliness - so that anyone can easily play with the MOPED by, for example, using the app. We would also like to improve the latitudinal driving. Following the red color works as long as the scenery does not have a lot of red in it.

3. Reflection on the sprint reviews and the sprint retrospectives

It is important to review each sprint. The difference in workload from sprint to sprint, the number of members in the group and the different ways each person works affected the working process. Therefore it is vital to monitor it continuously so the group works efficiently and that no tasks are overlooked.

Our reviews and retrospective sessions were combined during our general meetings after every sprint. It wasn’t planned to have them both together, but it became the best solution due to the limited time we had when all, or a majority, of the members could be present. The meetings worked as long as we had a clear lead and guidance by the Scrum master or some other member. With so many members in the group we found it crucial to have an appointed leading role during the meetings.

Trello was the main tool we used throughout the meetings. We checked every task and discussed with the members in charge of them if the task was completed, the amount of time it took and if our pre-decided KPI was accurate. While the discussion about our achievements went on, a couple of members wrote down and summarized the Sprint Retrospectives.

The way in which we applied Scrum wasn't perfect and there is a lot of room for improvements. One thing we would change, if we could go back in time, is to make a clearer distinction between the review and retrospective times. It would have contributed to the group's organization and communication as a whole.

Just to compare our method to how a big company works, we asked an employee at Ericsson about how they plan their reviews and retrospectives. The employee explained that they only have retrospectives and never actually did reviews, and was a bit puzzled about the difference between the two. For us, it was quite difficult to discern them and we think that is the reason why our group naturally, and without really planning it, merged them into one single session.

In conclusion, these sessions were crucial for the project and should have more importance in the future.

4. Best practices for using new tools and technologies

Many of the tools we had to work with in this project were new to us (e.g. python, Autosar, WirelessIno and Scrum). The first measure we took before we started working with them was to inform ourselves about the tools and technology by reading and researching documentation. After acquiring a basic knowledge we applied a practice colloquially called "Learn by doing". It is a common practice in programming, as some programing languages have some similarities making it possible to learn while actively rewriting and testing the codes.

Another practice we used was to seek guidance from people that have worked with these tools and technologies. It is sometimes a faster method than "learning by doing" since we get specific answers that can be applied directly to solve problems, but it could be less time-effective if the helper isn't available at the moment.

5. Reflection on the relationship between *prototype*, *process* and *stakeholder value:*

For our final presentation we were able to achieve completeness, meaning that the features prioritized in our backlog were implemented and we had a MOPED that was able to follow the MOPED in front of it without crashing. Our priorities were based on the wishes of the product owner which entailed a MOPED to be able to regulate its speed and distance without supervision and maintaining a safe distance while platooning.

In the mobile application we created buttons for starting and turning off CC and ACC, and implemented CC functionality in the application sliders. It would have been ideal to implement functionality for ACC as well, since this would have improved user friendliness. Also, the ALC steering can be made smoother and not as rough as the current prototype. One way to make it better is by making an algorithm for it rather than using hard values.

The stakeholder values were the guidelines we used as a foundation for prioritizing tasks and planning our sprints. When we had prioritized the stakeholder values, we divided them into smaller tasks and made small improvements that we later puzzled together to see if they correspond to the these values.

Our goal for each sprint was to complete tasks that were of value to the product owner, which we could demonstrate at the end of every sprint. First we started with getting acquainted with the MOPEDs and how to write code to them. We also had a task for all members to get familiar with the simulator. The reason why we had such tasks in our sprint is so all members would be more on the same level of knowledge and so that the actual coding and planning would become easier for all of us. We started by trying to come up with ideas on how to begin our work and what more resources we needed or were going to need along the sprints.

When planning each sprint, we would sit down together and estimate what we thought would be possible to achieve in a week, and chose the highest prioritized backlog items according to this. To help with the sprint planning we used Trello as an online pinboard. It made keeping track of tasks in progress and things waiting to be tested, easier which made the whole process clearer. The user stories in the backlog were put in the priority order preferred by the product owner. During the sprint planning we selected a number of items that we thought could be achieved from the top of our backlog and created the necessary tasks to complete them. The initial intention was to set an attainable goal of finding a good balance within the workload without overestimating and underestimating our capacities and time.

One problem we had was that we were not very good at dividing our user stories and tasks as small as we should have. Making the tasks smaller would have made it easier to estimate an effort value that was more accurate to our sprint velocities. It would also make it clearer for the group to know more about what was accomplished and what remainders from an epic or user story that has to be completed. That is one thing that should have gotten more focus and attention when planning the sprint.

The task effort estimation was hard for the group to grasp and get a proper understanding of. Hours was the agreed effort estimation unit for the tasks, but we came to the conclusion that time estimation made it harder to evaluate the difficulty of each task. There would be tasks assigned a low effort value despite being complicated and important for the progress of the project, while other tasks would be less important and have a much higher effort value. For example if two people were working on a task that we thought would take three hours, it would an the effort value of six hours, which it did not deserve. This consequently made it harder for us to recognize which tasks actually needed more time and energy. Therefore, we came to the conclusion that having an estimated size without deciding the unit would have made it easier to determine the effort to assign each task.

6. Relating your own process to literature and guest lectures

Talking to other people who work with scrum on a daily basis has been quite rewarding. They mentioned that they struggled with the same things as us - like the fact that you really need one or two group members who really push the others to use scrum. Without this it is hard to be good at it since it takes time to get used to the work process. Knowing they had the same struggles made us feel like we were on the right track. They also talked a lot about the pros of using scrum and the agile work process, such as it leading to always delivering something of value, and that it is easier for the product owners to make changes and them truly getting the product that they want. Knowing that these people who uses scrum in their daily life saw a lot of benefits, acted as a carrot for us to continue trying when we struggled the most. Also, just finding out that scrum is something companies actually use was extremely motivating.

By talking to people outside of school who work with agile development, we have learned that it is a process which allows you to constantly deliver something of value to the product owner. Using the agile triangle instead of a waterfall process allows the product owner to be more involved in the project, giving more valuable feedback and thus realise what kind of product they truly want. To describe the pros, a scrum master from *VCRS* gave us the example that the pros of this can be to avoid straying too far from the desired product through misinterpretation on our side. If the product owner is not pleased with the function then it will be easy to reevaluate and reprogram until the next sprint. Hence, we can see the benefits of scrum.

One lecture that was helpful was when we talked about how the get from a product owner's head to a product backlog to a sprint backlog. Here we learned about the importance of having tasks that are like cake slices, so that each task gives something of value to the product owner. We got to do a small exercise with this and afterwards we tried really hard to implement it when working with everything our product owner wanted. The film that Kenneth, our product owner, showed us made it easier to understand that he had an idea of what he wanted, and he was not bothered about all small details. As an example, Kenneth does not care about if we navigate using green or red pixels. Understanding this made it easier for us to be creative and realize that what Kenneth wants is somewhat our decision - as long as he finds what we deliver to be of value.

7. Evaluation of D1 - D4

7.1 D1 - LEGO exercise and Social Contract

After the Lego exercise we developed three strategies that have helped us with the project's process. "Divide and conquer" was a strategy we exerted constantly through the whole project. It could be improved by, for example, making sure every member has a task to take care of before ending the sprint review/retrospective sessions. The social contract we developed also helped forging our group and increased team spirit a lot. One more thing we learnt from the Lego exercise was to ask the product owner what he actually wants to make sure that we are doing the right thing. At scrum of scrum meetings the product owner was asked about important details of the project, we tried to compromise with him and agree on what should be expected and what we can offer. The last strategy we had in our first hand in was to not over- or underestimate the task velocity. We worked hard on that point and tried to be more precise when estimating our task velocities, but there is always room for improvement.

7.2 D2 - Product Backlog

Our first draft of the product backlog was not very concrete, and did not handle all aspects of the people who would be involved in the project. For example, we had only thought of what the product owner would consider, and not the product user. Fortunately, we got some response on this backlog and were able to improve it. We are pleased with how the product backlog looks now, and think that we were able to include everything that was of value. We would not change anything without input from the product owner. It has been helpful to revisit the product backlog when we were unsure on how to continue our process.

The product backlog also contains our definition of done. We could have worked more with this, as it was mostly something we checked off in the end of our project.

7.3 D3 - Halftime Evaluation

It was helpful writing a halftime evaluation. Our evaluation described very well where we were in our process, what things had gone wrong as well as some suggestions on what we had to do in order to be where we wanted to be. Unfortunately, we sort of said these things and then we just forgot them. At the time we did our half time evaluation, we started to make progress on the prototype. After being stuck for three sprints or so, we now had more concrete plans for how to get the ACC and ALC working, so this distracted us from restructuring our work process. This could be a reason for why we did not actively try to take our own advice to heart.

The most valuable outcome from making a halftime evaluation was that everyone had to say exactly what they had accomplished and what more they had to get done. This would then give insight to all members on where we were in the project, and it became easier to help each other out. Now we knew what the struggles were and where we could contribute, which was something we struggled with earlier. One thing we mentioned in the evaluation was that we wanted to improve the way we use scrum. A few of the group members tried to improve this while others found that other aspects were more urgent to improve. Perhaps having a “scrum fascist” would have motivated us to get on the same page. I suppose, at this time we did not really see all the benefits of using scrum, those insights came sometime around sprint six. Hopefully, we will not neglect scrum improvements in the future, but we think it is understandable that we did it back at sprint three and four.

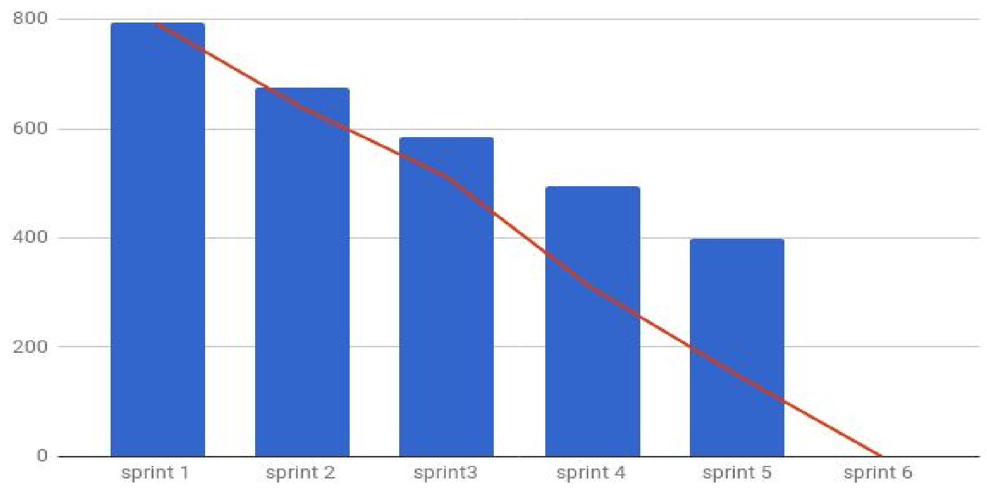
7.4 D4 - Prototype

Our presentation went well. At first we had some hiccups with the hardware, our first MOPED could only drive at full speed and our second one had a huge preference for turning right, but eventually we got that fixed with some minor adjustments in the code. It followed most if not all criterias for our user stories and stakeholder values. There is room for improvement, especially in that it acts a bit ‘drunk’, but that could be said for almost any project.

8. KPI charts

One thing that has been helpful was the KPIs. By taking tasks, that are otherwise quite abstract, and making them measurable, it is easy to see if something is working or not. As shown below in the burndown chart, we did not achieve all tasks in each sprint, so we had multiple tasks moved from one sprint to another. However we worked very hard in the last sprint and completed what we started.

As an example, in our burn down chart, we could see that for sprint 4 we were not able to deliver as much as we had wanted. Once we had this information, it was easier to evaluate what had gone wrong, and what we could do to improve our result for the future sprint.



Burn down chart:

- x-axis: the sprints
- y-axis: our remaining effort
- The red line is estimation, and the blue represents actual remaining effort