CHALMERS UNIVERSITY OF TECHNOLOGY

DAT255

M.O.P.E.D.

Reflective Report

Group Name: *Bra-ish*

October 27, 2017

Abstract

The basis of this project involves developing a successful form of a self-driving vehicle, also known as*Mobile Open Platform for Experimental Design of Cyber-Physical Systems* (MOPED). The goal is to present the way the team adapted Scrum (an agile, sprint-based work-flow), what difficulties were faced, as well as how these were handled. The lessons learned from the project are also discussed, not only from work-aspect, but also from the seminars held by numerous lecturers. While developing solutions for the vehicle, the team's way of implementing Scrum changed and evolved, which is highlighted later on, in comparison to the earlier part, where the original implementation of the work-flow is described. By succeeding with the task, the team could describe, as well as discuss the approach used in this report, why it was successful, and finally how it could have been tweaked to become more efficient.

Contents

1 Introduction

According to Transport Analysis, a Swedish government agency responsible for the production of official statistics in the transport and communication sectors, 270 people were killed in accidents on the Swedish roads in 2016[1]. This number has been declining over the years as a result of safer roads and cars, but to take it a step further and eliminate the human factor, many car producers are looking into autonomous driving.

In the transportation sector, manufacturers such as Volvo are developing what is called platooning[2]. Platooning is the ability to have multiple vehicles mimicking each other in a tight line, sending and receiving data to control the vehicles. This is mainly developed for reducing driving fatigue in truck drivers to lower the number of accidents, but also to increase fuel efficiency[3].

During the fall semester of 2017, students at Chalmers Institute of Technology were assigned the task of achieving platooning using small vehicles known as *Mobile Open Platform for Experimental Design of Cyber-Physical Systems*, or MOPED for short. The goal was to learn how to work as a team of 10 people, using an agile work-flow called 'Scrum'.

2 Methods

The following section presents the different methods the team worked with during the project.

2.1 Teamwork and Social Contract

Before the start of the project, a social contract was constructed. The contract contained how the team was supposed to approach the project in terms of communication and respect as well as the desired work culture. The social contract can be seen in appendix A. Expectations were also set on the individual members in regards to how much time and energy should be put into the project.

The team planned to split itself into smaller subgroups that could deal with different areas of the project. This would allow people to fully understand and focus on their respective areas of responsibility. It also took away the need for individual members to understand every single part of the system architecture. Although the team encouraged the members to work together, how and during what time of the day the subgroups worked was decided by the subgroups themselves as long as the social contract was followed.

As per the Scrum method, a Scrum Master was set to dictate the communication between the product owner and the Scrum Team. The team also decided it would be helpful to have a secretary. The secretary's role was to keep track of all the documentation from meetings during the span of the project.

The team would have a continuous discussion about how well the team worked for the whole duration of the project. This would be done by reflecting on the team's work, both on an individual level and together as a team.

2.2 Sprints and Meetings

The project was divided into six equally long sprints, each lasting one week and ending with a mandatory sprint review meeting. Three times a week, short Scrum meetings were held to keep the group up-to-date.

During the *Sprint Review and Retrospective*, the team discussed and evaluated the work that had been completed during the previous sprint. Later the same day, the upcoming sprint was planned. The Review and Retrospective usually occurred prior to lunch, whereas the *Sprint Planning* took place after lunch. This was done to establish a clear division between the two sprints even though they were discussed on the same day.

During the Sprint Review and Retrospective, all members would get the chance to speak about their personal contributions during the latest sprint, both in terms of what was done well and what could have been done better. After that, a group discussion was held discussing the team's overall performance. Changes that could improve the efficiency of the team were then discussed based on the issues that were mentioned previously. If the team agreed on a change, it was documented and implemented into the coming sprint.

2.3 Workflow and Technical Tools

During the Sprint Planning, the team identified what needed to be done next based on our final vision and what already had been completed. The sprint backlog was readjusted into user stories as per the Scrum methodology. To have a clearer understanding of exactly what each user story consisted of, they were later split up in tasks. While a user story was more vague, all the corresponding tasks explained each step that was to be completed to mark the user story as done. The team discussed each task and prioritized each one of them by rating. An *effort* was set on each task, ranging from 1/2 to 100.The team would aim to complete 50 effort each sprint. The tasks were then assigned to a subgroup within the team which consisted of one to four people, depending on the assigned task's expected workload. This was done to maximize production, while not assigning too many people to the same task.

The team used several online tools to aid their approach to Scrum. These include solutions for managing tasks, planning, communicating and version handling. These tools are described in the following subsections.

2.3.1 Trello

The application Trello was used to establish and maintain an overview of and structure behind the project. Trello is most easily described as an online live pin-board, structured into different categories. Trello allowed the team to create and organize for example user stories, tasks and bugs.

When Sprint Planning was conducted, user stories and tasks were moved from the project backlog to the sprint backlog. Once a subgroup began working on one of these tasks, the task was moved to the *In Progress*-column. If a task was finished but had not been tested, it was later moved into the *Ready to Verify*-column. Once the task was tested it was finally moved into the *Done*-column. At the Sprint Review and Retrospective, all members of the team would discuss whether or not a certain task was considered completed or not. Old sprint backlogs were saved on the board so the team could both track their previous work as well as forecast the upcoming week in tasks.

2.3.2 PlanITpoker

A website called PlanITPoker was used to simplify the assignment of effort to the user stories and tasks there were to be worked on in the coming sprint. PlanITPoker allowed the team members to individually assign a task with an effort, and once all members had chosen the results were shown. Thereafter a short discussion was held after which an effort was set on user story or task in question. The effort was then promptly written into Trello. After using PlanITPoker, all user stories and tasks in Trello would be assigned with an effort that was discussed and argued for.

2.3.3 Communication

The team used Facebook Messenger and Slack as the main means of communication. All issues, plans or other serious inquiries were to be handled on Slack, however daily and more casual talk was held on Messenger. Ultimately, most of the team used Messenger more than Slack; this might be due to not all team members receiving notifications from the latter. Meetings and workshops were scheduled in a shared Google Calendar so that all members could refer to this when planning.

2.3.4 Repository and Git-workflow

To enable version control, and allow a structured way of organizing files, the team created a GitHub-repository. The team's best practices were not strict in the beginning and it was decided that the small groups could decide how they would implement Git in their part of development. The general practice was to branch when you performed large changes to the existing code. When it came to code review it was decided to do as much development as possible in pairs, but that all members could push code to the master branch freely. No code review system was implemented due to such small groups working on each subsystem.

2.3.5 Pylint Python Code Checker

Pylint is a tool used to identify and correct bugs in the Python code constructed during the project. It is a simple and useful tool to structure and reduce the amount of unnecessary code. Since a goal was set to minimize and track the number of bugs per KLOC (Kilo Lines Of Code), Pylint was of great assistance.

Pylint also enforced a coding standard and looked for bad practice in coding, known as "code smells. While checking the errors and code smells it also recommended suggestions about how specific blocks can be refactorized. As well as helping with refactorization it gives a sense of the code's overall complexity and rates it with a number up to 10.

The goal was to have as perfect code as possible according to Pylint, aiming for a rating of 10/10.

2.3.6 Findbugs

The team decided to use FindBugs as a tool for evaluating the code quality and to find mistakes in the code. The team used it on the java code and used the results given from FindBugs to track their progress throughout development.

2.3.7 Simulator

During the sprint planning prior to the fifth sprint, the group decided to start the development of a simulator. The main goal with the simulator was to have a platform on which the group could test the server and MOPED despite the different hardware problems with the MOPED. Development started during sprint five, but was put on hold in the next sprint due to all group members fully focusing on finishing the code for the real system in time for the demonstration. The development of the simulator eventually ended during sprint six.

2.4 Time distribution

The group addressed time distribution by letting each member choose when they wanted to work. However, all members agreed on delivering 20 hours of work each week. Working more or less than 20 hours was something that should be avoided, in addition to working on weekends. All of these regulations were recorded in the social contract, as mentioned above.

3 Results

The following section presents the results of the project, both with regard to the Scrum-methodology and the general working process, as well as the final project.

3.1 KPI

Through the project, the group worked with three different KPI (Key Performance Indicators). The KPI:s that were to be used throughout development were decided when planning the first sprint. The team decided to track the team members' happiness, bugs per lines of code as well as velocity (effort completed per sprint).

3.1.1 Team Member Happiness

Over the course of the project, all group members were to note their current happiness on a scale from one to ten. This was to be documented on a shared Excel sheet. These evaluations were made in conjunction with the daily Scrum meetings. Happiness was not directly connected to the project, but rather how each member was feeling that particular day. It should be noted that the average happiness value rose in correlation to making higher amounts of progress in the project, as seen in appendix B.

3.1.2 Bugs per Line of Code

When developing the system, bugs were introduced in the code. The team decided to track the number of bugs divided by the KLOC (Kilo Lines Of Code). The bugs were tracked using two kinds of software: FindBugs, which finds bugs in Java, and Pylint, which finds bugs in Python.

The tracking gave the team an insight into the current code quality as well as an indicator if the team focused more on quality or quantity.

3.1.3 Velocity

Velocity represents how much work the team estimates that they will complete during the course of a sprint. The group's goal was to deliver 50 velocity in each sprint, which translates to ten people working ten hours per week. The set velocity value was used during the sprint planning, to choose which user stories the group would work on in the following sprint. This meant that the selected user stories should have a total value close to 50. The burndown-chart of completed effort per sprint can be observed in appendix B, under effort.

3.2 Scrum

The most significant change as to how the team implemented the Scrum methodology during the course of the project was that in the later stages, collaboration and sitting in the same room and working together, even pair programming, was done daily. This led to work being done more efficiently, due to it being much easier to help each other with different problems. If questions arose about a certain part of the code, it could be answered immediately by the person that wrote it. It also created a broader understanding of the system as a whole for all team members present. The team could deliver more to the project owner as a result of this.

At first, the user stories created during sprint planning were very basic and technical, and left little room for personal interpretation, which was discussed during the second sprint retrospective meeting (see appendix C.1). This meant that the user stories were horizontally sliced, which is not desirable. The team soon discovered why it was not desirable since it led to subgroups not having anything to do, as hardware problems restricted work at times. Further on in the project, the way in which user stories were written changed, and they became completely vertically sliced leave more room for interpretation. The change was due to the team change in perception; instead of looking at what was completed and what could be built from it, the team remembered the vision and the desired end result. A vertically sliced user story slices through all architectural layers and allows for all pieces to be built, integrated and tested.[4] This led to greater creativity when problems arose, and a greater sense of completion in all of the subgroups.

In the first couple of sprints, the user stories were not broken down into smaller tasks. This resulted in subgroups having poor overview of the sprint's current progress and it was not clear what should be done next throughout the sprint. After these first few sprints, the group reflected upon this and decided that each subgroup, once assigned user stories, should break them into smaller tasks. These tasks were then to be assigned to individual developers. The developers were responsible for tracking the task's progress in Trello by moving it between columns (such as In Progress, Ready To Verify, etc). This gave the entire group a much better overview of current sprint progress, allowing for better cooperation and communication. This in turn made the team more focused on delivering upon the vision that was established at the current sprint's planning.

The roles that were set at the beginning of the project did not change at any point. At the end of the project, we still had a Scrum Master and a secretary because it had worked well through the entirety of the project. The group also maintainted their stance on individual responsibility: all group members had to be responsible for completing tasks and delivering. Also, if a group member felt that they had fallen behind, there was opportunity at every Scrum meeting to shed light in the issue and ask for help. This was all part of the social contract which the group had agreed upon. Despite establishing and agreeing upon a social contract, the group did not refer back to the contract enough. The social contract was not discussed at any sprint reviews, and thus whether or not individuals were following the social contract was never discussed nor addressed.

3.3 Problems During Sprints

During the sprints, the team experienced numerous problems. These are described below.

3.3.1 Change in Server Structure

During the first sprint review (see appendix C.1) the team acknowledged that the server was a problem. A discussion was held, and the team managed to solve the issue at hand. However, in the second sprint, the team were force to arrange an extra meeting to discuss new and bigger server issues (see appendix C.1). The team decided to develop a server themselves which would be better suited for the team's needs.

The problem with the existing server was that it was uncomplicated in theory; follow the instructions on the provided Github repository that contains documentation for the MOPED and, when done, continue to develop the autonomous system. The problem, however, was that the instructions were inadequate and that the complete system had not been tested outside of the lab environment that it is normally used in. The system consisted of three elements; the MOPED, a Wordpress installation with its related MySQL database, as well as a server that communicates with both Wordpress and the MOPED. The team successfully installed Wordpress after making minor code changes. To get the server up and running two team members had to put in several days of work. The server did not run on Windows, and not on the Linux subsystem on Windows[5] either. This is due to Windows file system not being compatible with the server start-up script. Eventually, the team got the server running on a computer with Ubuntu operating system after some modifications in the bash script used to set up the server.

The aim was to not code directly on the MOPED, but instead to develop plugins that were to be installed on the MOPED via the server. There were a few finished plugins in the repository from the beginning. After successfully getting the intended server to run, the next natural step was to connect the MOPED to the server and to install one of the plugins on the MOPED. This is where the team faced the next problem. According to the server, the installation was successful, but nothing happened on the MOPED. To be able to debug we needed to connect an RS-232 cable from the MOPED's VCU to a display. The course supervisors provided the cable, but the cable was only compatible with Windows 7 and earlier versions of Mac OS. Unfortunately, we had access to neither.

In the middle of the second sprint, an extra meeting was held. During this meeting the team debated what technical solution that should be proceeded with. After a group discussion, the team decided that the intended server system would be abandoned and instead a custom Java server would be developed. The suggested setup was already pre-configured for use in the system, and all of the documentation assumes that you are using it, but the reality was that it did not function properly. The team based the decision on that the work with the server system had been filled with problems from the start, and that as soon as one problem was solved the next one would arise. The assessment was that there was a risk to continually run into new problems and that the problems might even be caused by the modifications that the team had done to solve prior problems. Both the risk and workload was well on its way to outweigh the benefits of staying with the intended solution.

3.3.2 Lack of User Stories and Tasks

Towards the end of the first initial respective sprints the team encountered a number of issues regarding the backlog: there were simply not enough tasks defined in the backlog to cover a whole week's worth of work. The tasks were also described in such a manner that they were too narrow and specific.

The lack of user stories sometimes led to the belief that there was no more work left for the week. The reason behind this problem was arguably that the group was too committed to the Scrum-framework, which in turn was misunderstood by the majority of the group. This was later solved by creating the user stories in such a way that they were more general and less technical. This change allowed the individual sub-teams and smaller groups to work more independently and thus rely less on each other in order to make progress. Various technical problems were also easily worked around because a solution was never initially defined from the first place, meaning that there was greater flexibility when approaching said problems. This issue is more thoroughly discussed during the second sprint retrospective, as seen in appendix C.1.

3.3.3 Alternative Testing Procedures

Very early on in the project, it became obvious that a uniformed official testing procedure was going to be difficult to fully maintain due to limitations related to hardware issues. A lot of the focus was centered around hardware and having the various components to function properly so that the testing could follow.

The plethora of hardware issues lead to the group coming up with a solid scaffolding code but nothing that was neither finished nor deliverable.

One way around this was to analyze the parts that were causing the problems and trying to work around them. For example, instead of being stuck with the camera not properly operating or with failures with collecting the sensor data, simple mock-ups were used to mimic the hardware. This allowed us to test our code without relying on the hardware functioning properly.

Making the transition from the mock-ups to the actual hardware was in most cases very smooth and easy to test because it was the only variable into the system.

This lead to a relatively quick and seamless integration with the other parts, such as server and app.

3.3.4 Subgroups Out of Sync

During the second sprint, the mathematics course had a test which lead to the majority of the team focusing on studying for said test. A consequence of this was that the subgroups ended up out of sync. A conflict arose between a few members where the ones not focusing on math continued working while the others were left behind. After the mathematics test was done the members that had studied for the test had to catch up, but the ones who had not studied for the test did not want to wait for the others, and time to sit down together to synchronize work was not found.

The conflict was dealt with on a Sprint Review-meeting (see appendix **??**) by a discussion among the team. Following this, the subgroups were changed and each individuals priorities regarding the math course, as well as current knowledge of the system and what they themselves desired to work with, was taken into consideration. The group also put more focus into good communication after this, especially if subgroups did not work together at all times.

3.3.5 Out of Sync with Product Owner

During the project, the team had an assigned Product Owner. Each sprint there would be a meeting with him to ask questions and make sure the team created a product according to the Product Owner's needs. Due to complications, such as sickness, meetings were canceled and the team lost most of the communication with the Product Owner. This lead to an insufficient amount of feedback and some guessing from the team to be able to finish the assignment. The team's solution was to develop their own vision for the result, while taking guidance from the information that was given by the Product Owner at the start of the project, and created user stories to reach this vision. This resulted in the group developing features for the demonstration that was a bit different from other teams' features.

3.3.6 MOPED Hardware-Issues

Tracking MOPED issues was made difficult due to the group needing to switch MOPED every week. This meant that problems and issues that were present in previous weeks could be nonexistent the next week. In addition to the situation stated above, a couple of hardware issues occurred that prevented the team from moving forward.

Firstly, a great issue was the ultrasound sensor attached to the front of the MOPED. This sensor was essential for the advancement of ACC (Adaptive Cruise Control) and due to issues, the ACC development was postponed from the integration side. The main issue was that the sensor was either not responding or that it was broken. When trying to fix this during a workstation on Lindholmen, it was known that the issue was not only in the hardware but also in the software that handled the hardware. No solution was given to this issue until later the same week. This issue made the development of ACC on the MOPED more difficult than previously anticipated.

Once this was solved, an immediate test, and a successful one as well, of ACC was done.

Another issue which was encountered was the camera. The camera was a crucial component of image recognition. The camera malfunction was not detrimental however, as our camera mock-up was somewhat adequate. This allowed the person developing image recognition to be able to develop image recognition despite the issue. Once the group received a functioning camera, a few simple tweaks were made in the image recognition code to make it function properly.

The real struggle was not the hardware issues on the MOPEDs themselves, it was the process of spending several hours on fixing one MOPED, and then having to give it away to another group and receiving a new MOPED. The group had little knowledge of eventual issues in the newly provided MOPED and thus the process started from the beginning once again.

3.3.7 Testing

In the middle of the project, the team had a lot of problem testing code. The reason was that because of the hardware problems, the development on the MOPED was behind that of the app and server. This resulted that the team had a lot of user stories in the Ready to Verify section on Trello.

The team later found a solution by dropping the MOPED in their testing and started to develop unit test and smaller mock-ups and simulations instead. This allowed the server and the app to be tested without the MOPED. When the team got the MOPED to work in the last sprint, it took just two hours to get the code, which had been tested in simulations and on MOPED mock-ups, to function properly on the actual MOPED. The team also decided to work together more in a structured way, with daily To Do lists, where one or two members of each development group could get together and do proper testing of the entire system. This lead to more work being completed while the team worked together as a big group.

3.3.8 Lack of Documentation

A few sprints into the project, the importance of documenting the progress was discovered. With documentation, the subgroups could share information that was vital to other subgroups. For example, a subgroup waiting for a specific part of the system could see how had much progress had been made and thereafter could adjust their own schedule. On the fourth sprint retrospective meeting (see Appendix C.1) the team discussed that comments in Trello should be added on the cards that represented the different tasks. The comments contained important information about the task and how they were tested. Even with the daily Scrum meetings, this led to all team members getting a better overview of the work, and quicker updates about certain points that could be of interest.

3.3.9 No Technical Meetings

While following the Scrum methodology means having many meetings during the week, the group did not have any technical meetings. This led to minor problems since the subgroups did not always work towards the same technical solution. For example, the communication protocols between the server and the MOPED had to be changed slightly due to the subgroups working on a solution that fit them and not a solution that matched the other part. This was solved automatically when time spent working together increased, but related to this the group also started to document communication protocols. Had technical meetings been held from the start, or at least from the point when the new server was developed, it would have benefited the entirety of the team greatly by increasing understanding of the architecture and assuring that all of the team was aiming for the same goal.

3.3.10 Malfunctioning CAN-Network

A Controller Area Network bus (CAN bus) is a robust vehicle bus standard, designed to allow micro controllers and devices to communicate with each other in applications without a host computer. It is a message-based protocol.

The CAN bus served as the backbone to all of the communications between the different parts of the MOPED.

Due to its crucial role, any minor failure would cause the whole chain to fall short. Such failure may happen on the hardware level, causing the messages not to be fully read or written onto it. Another scenario is the mismatching software trying to read and write data using different conventions and data structures.

3.3.11 Switching MOPED

As previously mentioned, the group had to share MOPEDs with other groups due to the number of available MOPEDs being less than the number of groups. The plan was to have a MOPED for a period of six days and then pass your MOPED to the group that was next in line. One group had to work without a unit, which led to one group constantly being without a MOPED.

This setup led to numerous hours of fixing errors or malfunctions attached to the received moped, as well as configuring the received MOPED to work with the previously used software.

3.3.12 Camera Malfunction

Midway into the project, a decision was made to make use of the camera attached to the MOPED. However, handling the camera was not an immediate success due to it not functioning properly in the beginning. Several tests were made to ensure whether the issue was in the Raspberry Pi, or if the issue was with the camera hardware.

Through testing the camera with another team's code, it could be observed that the hardware was malfunctioning and a replacement was due. However, there were no available functioning cameras during that time.

Following days, a mock-up code that could represent a camera and its functionality was made to ensure that the technique that was thought out was working, therefore a foundation for the code with a working hardware was made.

In conclusion, working this way lead to a quick and easy install from mock-up to a real functioning camera and because there was a foundation, it was very easy to integrate it with the server and gave way to stable and successful image capturing as well as sending the images to the server.

3.3.13 MOPED Repair Session

The team got a chance to solve the MOPEDs hardware issues during sprint five. This was to be done at the lab located at Lindholmen Science Park. After spending seven hours together with the other teams and technical supervisors, a majority of the hardware problems were solved. This was the first time during the project that the team had a MOPED with fully functional sensors.

The battery problem stemmed from a dead cell, which was an issue that could not be fixed. A new battery had to be ordered, and this led to features that needed testing to be considered done and delivered, could not be tested for three days while waiting for the new battery. All the testing, therefore, got postponed to the next sprint.

3.3.14 Lack of Backups

The subgroup working with the MOPED had a poor backup routine for the files on the microSD card that was used for the MOPED. This resulted in the group needing to rewrite code when the data on the SD-card was lost while hardware was repaired. After this incident, the group started to do backups and publishing them to Slack when major changes were made.

The decision to not use GitHub as a version-control for the MOPED-code was made because the code on the MOPED was heavily reliant on hardware. Instead, a backup was made on a different computer and sent to Slack, with the idea of having Slack as a sort of cloud storage. This worked perfectly when encountering issues with corrupted files or accidentally deleting all the files on the Raspberry Pi.

3.3.15 Slow CAN-Data Transfer

To reduce the reaction time and increase the reliability of the active cruise control (and the system as a whole) the group tried to reduce the response time within the different parts of the MOPED. To do this, the team had to make sure the backbone of all communication on the MOPED was working with the lowest possible delay rate.

During the initial state of code and hardware, the ultrasound sensors could only handle a frequency of 1 Hz. This means that the MOPED can only react to the object in front of it once every second. This was not suitable for the team's use of the sensors and changes needed to be made.

At first, the team tried to implement code another group constructed to increase the frequency. However, this did not work due to being too dependent on a major part of the other team's code-base. Instead, changes were made in the MOPED. One change was reducing memory usage from other unnecessary debug logging. Another change was increasing the rate of which the data was sent to the server which lead to a slight increase in frequency.

Although slightly changed and increased, the data transfer was never perfect and as fast as preferred. This was despite a lot of thought going into the process of increasing the frequency, as well as a lot of discussions with other groups that successfully increased the frequency. Another team, which had sucessfully increased their frequency by a substantial margin, tested their code on other MOPEDs. The teams present realized that there were differences between MOPEDs. In conclusion, the team came to terms with the fact that the MOPED made available to them was not able to reach a much higher frequency, and thus work in this area was deemed finished.

3.3.16 Problems with Power to VCU

During the last sprint, the main focus was to have a perfectly capable MOPED for testing and calibrating to ensure a successful demonstration on the due date.

The final MOPED was received 5 days before the demonstration. This allowed time for testing the whole system and looking for any defect hardware or software.

Upon testing platooning, the motor started to shut off its vehicle components and the gyro started twitching uncontrollably. Trying to analyze the issue, the group quickly realized that a cable had a loose connection. The cable was essential in powering the VCU, which in turn controlled the motor. Due to the lack of time and no nearby working stations being available, the only solution was to cut off the wires and solder them together.

3.4 Final Product

The final product that was presented at the demonstration consisted of three major building blocks. An Android application running on a smart-phone, a Java-based server and the actual MOPED itself.

The Android application held the functionality which allowed the user to maneuver the MOPED. The user could set the speed, the direction of steering and activate programs such as ACC (Adaptive Cruise Control), ALC (Adaptive Lateral Control), and autonomous steering. To see the application's design, see appendix F. To simplify the steering view, functionality was built to either enable or disable some of the visible controls depending on the user's input. For example, if the user decides to start platooning, all other controls are disabled on the screen. This simplifies the application usage in a logical way while also removing the possibility for the app to send contradicting signals to the server.

The commands that the user sets in the Android application are sent to the server via certain TCP (Transmission Control Protocol) sockets. All the calculations needed for the programs to run are handled by the server, and it also forwards the simple commands straight to the MOPED.

The server has a GUI (Graphical User Interface) which acts as a simple debug screen as it continuously updates and display important MOPED data. The GUI also renders the images from the MOPED-attached camera with the data created by the image recognition class located on the server. The purpose of the image recognition class is to scan the images sent from the MOPED. The scanning is conducted in such a way that it detects certain objects, both by shape and color contrasts. This results in a more robust and reliable algorithm with fewer false-positives compared to a color-only image recognition algorithm. However, the MOPED does not only send images, it also sends data from the sensors to the server, which in turn responds with a steering and velocity command.

The majority of the calculations are made on the server, but the MOPED does have certain security methods. These methods are primarily used in the case of a connection mishap between the MOPED and server, or if something is malfunctioning in the system.

4 Discussion

Hardships and unexpected amounts of problems lead to several changes being made during the project, which are discussed in the following sections.

4.1 Final Product

The subsections below contains discussions about the process towards the final product. The final product consists of the server, Android application, code on the MOPED and the simulator.

4.1.1 Server

The server was one of the areas that the team spent the most time and effort on, and its role changed a lot as time went by. Due to the many problems the team had with the server system provided by sics, the team decided to develop their own server written Java, and that formed a big part of the rest of the project.

The change from using the intended server system to developing a new server meant that the team needed to redo major parts of the second sprint backlog when being in the middle of the sprint. The team realized that they would probably not reach their set velocity of that sprint. An argument that the team saw as integral was that even though it might delay the delivery because there is a possibility that they would not be able to finish their current backlog, the team would easily catch up later due to having a better understanding of their architecture. Had the team been one of the only groups that experienced problems with the intended server system the team would have stuck with it, but since none of the other groups had been able to successfully do a complete system setup, the team deemed it taking them less time to develop their own server, even though none of the team member had done anything like it prior. As a side note, most of the problems with the server setup could probably have been avoided if the system had been tested outside of the lab environment where it is usually used.

The system design the team decided on was a modular design, and to have the moped and Android application be black boxes from which the server got commands and data from. The server could then respond back to with steering signals to the MOPED. This architecture gave the team a lot of benefits compared to another solution which would have been to have the app talk directly to the MOPED.

The system architecture on which the team decided gave the team the ability to work with continuous integration. That meant that we did not have to change the code on the moped too much. This proved to be one of the deciding factors to the success of the project. In the end of the project after all the hardware problems had delayed a lot of the work with the MOPED, but due to the architecture, the server development was not delayed as much. This allowed the group working on the server to unit test all the functionality and algorithms. After successfully testing the programs and receiving a functional MOPED, it took two hours from the first integration test until the MOPED ran ALC.

The architecture also enables the team to see a more real-world perspective and to quickly make changes in the server development environment rather than to edit code on the MOPED. This translates to a company pushing code to a cloud server rather than to require car owners to bring their car to a mechanic to update the software. By doing this the team cut the development time almost in half and could deliver a working ALC just three days before the demonstration.

The architecture also allowed the team to add new programs and functions to the system. This is because of the open and easy extendable nature of the architecture. The team always tried to have a real-world comparison and to develop for a real-life scenario to keep the team on track even when the moped was not functional.

There were a few problems the team did not realize when deciding on the architecture. When developing the entire system in-house you also have to account for all the safety aspects of a third link in the chain, and you actually have to build everything from the ground up. This is a very time-consuming task and the group members working on the server had a very high work load, partly because it was the first time anyone in the group developed something of that nature. The members of the team were used to develop simpler user applications and had never before developed a server environment before which delayed the development further.

Even though it led to many hardships, the team learned a lot from the experiences with the server, developing a large multi-part system, and how to design high bandwidth application for network use. All of the team agree that the decision of developing a custom server was the right one, and it contributed greatly to the team's success.

4.1.2 Android Application

The android application, named Wireless M.O.P.E.D. Controller, was designed to be the user's only interface while controlling the MOPED. The app contains functionality to upon start-up connect to the server via IP and port. Once connected, the user is taken to another view where he or she can manually steer the MOPED, set an exact speed in cm/s, start adaptive cruise control or image recognition-based steering, as well as start platooning.

The code in the application was built following the Model-View-Controller (MVC) design pattern. This design pattern divides the code into clear parts and introduces clear structural guidelines for communication between different parts of the code. The views handle user-input. The controller controls the views and enables the views to communicate data to the model. The model contains the application's main logic, and receives data from the views via the controller. The model can send data to the different views via an implemented "observer" system. For a visual representation of the MVC implementation in the app, see appendix F.

Despite developing the app to be as thin as possible in regards to logic, it converts the user's steering inputs to properly structured steering signals which the MOPED can understand. These steering signals are sent to the server and can then be directly forwarded to the MOPED without any need to modify or interpret the user's manual steering input. This was done to let the server fully see the app as a "black box" as mentioned above in the server section (4.1.1). Also, this simplifies eventual modification if future development requires the application to be directly connected to the MOPED.

4.1.3 Code on the MOPED

The code running on the MOPED represents the interface to the logic-processing parts handled elsewhere in the system. It must work closely with the hardware as well as try to communicate and gather raw data in a more abstract way. Having the code provided by the MOPED developers written in Python did require us to continue using this interpreted language to further develop the code-base on the car.

During the design process and due to the nature of Scrum, modularity was the best strategy to achieve and test vertical slicing. This resulted in a component-based solution, whose features can be requested and activated on certain signals.

Running the entry script on the MOPED triggers two different threads to start their own separate set of protocols, each of which is independent of the other, facilitating the integration and testing process and guarantees a more robust structure for future extensions of functionality.

Most of the time spent developing the logic and protocols on the MOPED had to be mirrored on the server, to make sure that when sending data it was being received and nothing was being lost or misinterpreted.

Having one entry point to the code meant more flexibility when changing the MOPED every week. It could be easily installed in case of something went wrong with the storing medium.

Keeping track of the code using version controlling systems or by manually keeping backups available online, was crucial when finalizing tests, due to the fact that the slightest of changes might have caused the system to crash, and having a functioning restore point meant more safe coding and testing.

During the early stages of the development, the habit of logging everything on the screen was needed. Having to work with a Raspberry Pi running no form of GUI meant that the usual perks of a modern-day Integrated Development Environment (IDE) were missing. Towards the end of the course, the logging system turned into a full-fledged control center where the team could see in real time all of the data that is being collected and transferred, which was handy when troubleshooting in the later stages where much more complex components and logic were integrated.

4.1.4 Simulator

The simulator was a big area of discussion, both during the sprint planning meetings but also during daily scrums. One part of the group thought that the focus should be on the MOPED to have a working prototype to show on the demonstration. Others thought that the group should not depend on the hardware and that a simulator should be developed because of this, so if a working prototype would not be finished, the functionality could be demonstrated on the simulator instead.

In sprint six the team finally decided to cancel all development of the simulator. The decision was based on that the MOPED was tested and showed no issues. The team therefore decided to put all effort into finishing the MOPED for the demonstration.

The reason behind waiting with the simulator until sprint five was due to that the team underestimated the importance of being able to test the whole chain of modules at once. Unit testing, and trying out code on the MOPED worked very well for each individual module, but as soon as the MOPED did not work, the group could not guarantee that the code worked as intended.

In hindsight, having a simulator would have helped demonstrating the capabilities of our implementation. The live demonstration was unable to provide the scenario which the ACC was designed for; having a simulator would have been helpful in order to show a multitude of different scenarios and provide data on under which circumstances our implementation was able to perform optimally.

4.2 Scrum

The team's implementation and understanding of Scrum has been constantly evolving, which meant that the best practices agreed upon at the start of the project did naturally change as well. In addition to the process evolving, the team's mindset regarding the workload evolved as well. At the beginning, subgroups with many members accompanied with broad user stories resulted in several people working towards the same goal, all the while not fulfilling any of them effectively. The group did manage to solve this issue by creating better user stories focusing more on the vision of getting platooning to work. After the user stories were created, they were further divided into different tasks to achieve the vision. In Trello it is possible to assign members to different tasks. By doing this, people knew what to do.

During the project, the subgroups learned how to contribute more and start working together in a big ten-member team. During the last sprint, the team tried to work more collaboratively by working together in the same room. This was done because the team figured out that sitting together working helped when questions needed quick answers.

In the last sprint, the team changed the way they structured their work. To have a working prototype at the demonstration, the group started to focus on the vision instead of giving effort to tasks and agreed to work overtime. The importance of focusing on the product instead of the project had been talked about by Maria Carlsson on Zenuity's lecture on Agile work, so the group decided on trying this approach[6]. This was discussed during the fifth sprint retrospective meeting (see appendix C.1), the changes would not have been necessary if the first weeks would have been used better, which was not the case. Instead, the subgroups were too big and the lack of knowledge about how to use Scrum resulted in subgroups working when tasks could not be finished at the time. The group then learned to work around the problem and find alternative ways to work.

All in all, there are many positive aspects of using the Scrum methodology. If something negative should be said about it, it was that the team sometimes felt that it was limiting. Consider a case when a subgroup is unable to deliver due to hardware problems; Scrum still requires the groups to spend time on meetings, when it probably could be better spent on trying to fix problems or writing code. However, looking back it mostly felt limiting during sprints that were filled with problems, and much less so when the work flow was good. The short sprint time, and pressure to deliver made it feel stressful. Instead of attempting to adhere to the Scrum structure while still delivering good results, the group could have chosen to prioritize one of these aspects given that the project was only worked on part-time and that a working product was only a secondary goal.

Another aspect, which lecturer Christian Fritiof talked about during his seminar about Scrum and Startups[7], is that Scrum is very easy to start implementing, but also very hard to master. As mentioned before, this project was the first time using Scrum for all members, which can explain why the implementation wasn't as effective as hoped, throughout the project.

4.3 Roles, Teamwork and Social Contract

As mentioned earlier, daily scrums were held by the team three times a week. The notes from these meetings were in the first few sprints written by the secretary. These notes were not detailed enough and slowed down the meetings significantly. To solve this, the team agreed in the third sprint retrospective meeting (see Appendix C.1) that each individual should write down their own progress and then presenting it to the rest of the team during the meeting. The other administrative work for the secretary remained the same. By filling out the daily scrums individually before the meeting, the members became more aware of their own process and the document became clearer and easier to follow. By doing this, the team members had to think through what had been accomplished since the last time as well as what was to be done next. This way, the meetings could focus more on if someone needed help or on various questions.

Throughout the project, the social contract (see appendix A) gradually lost its purpose. It was clear that all members remembered what was written, but as time passed, the group referred to it less. For example, in the beginning, all members arrived on time to meetings. This was expected, according to the contract, but later on, more and more members started to arrive later than planned, without a valid excuse. This was against the contract, but since the group had no clear and concrete way of preventing this behavior, nothing really happened.

A way to utilize the contract better would be to revisit it during the sprint retrospective. If the group would agree that the contract was not followed during the sprint, a discussion could be held and a possible solution decided. For example, if a group member was late, he or she would have to bring ”fika” to the next meeting. This concept was discussed during a meeting, not in relation to the contract, but was downvoted due to the team wanting to have a more relaxed work culture. This work culture might have contributed to the increasing amount of late arrivals, in hindsight.

4.4 Structure of Sprints

The team kept the same outlined sprint structure during all of the sprints. As a result, all of the group members experienced a continuity in the expected time spent on the project each week.

Since Fridays were spent only on meetings, as the sprint retrospective and planning would take about six hours on average, one full day of code time got lost. This was not desired, due to all the problems taking up a big chunk of time. Also, when the Scrum planning was held on Friday afternoons, a lot of the information and the discussions became partly forgotten over the weekend, leading to a loss of flow.

A better alternative would have been to split the sprint retrospective and planning to Friday afternoon and Monday morning, respectively. The problem however was that the mathematical course had a lecture on Monday mornings, and it would have collided with the intended time for the sprint planning. If the meeting could be placed in that time slot however, this would lead to a better motivation in the team, and there would not be a need for each team member to revisit the sprint planning document due to not remembering what was decided. Another positive change might have been that the meetings would in total be shorter. As the team was expecting to spend the entire Friday on meetings, the meetings would usually last longer than they needed and focus would be lost.

The group opted to have the regular meeting every other day instead of the more standard daily meeting. Seeing as the project was only intended to occupy 20h of work every week, this worked well; in order for every member of the group to have something to show at these meetings, two days proved to be a sufficient interval in which to produce any meaningful results. It should be noted that the subgroups had undocumented daily "scrums" in order for their work to progress smoothly - this was a good alternative which made it possible for the whole group to work flexibly while having the subgroups retain their focus on the current goal.

In the beginning of the project, we had a larger focus on participation rather than the documentation and information sharing in regards to the Scrum meetings. This proved a faulty approach due to the importance of good documentation, which the team initially did not put as much thought into due to participation playing such big part of the meeting. When the group later decided to put bigger emphasis on the documentation, it was no longer necessary for each member to remember all the information presented on the meeting - instead, the group decided to let all members provide apt documentation in preparation for the meeting. This proved to be the better approach; in fact, by having all members provide written descriptions of their completed work, the group might have been able to forgo physical daily scrums. However, a vital part of the meeting is the option to pose questions to others and have a live discussion - the physical meeting would have to be replaced with some kind of live communication medium at the very least.

4.5 Time Distribution

All team members logged their working hours in order to track the time spent by the entire team as well as by every individual. Helping in providing meaningful statistics per sprint and for the project in its entirety.

When analyzing the work-log, it was clear that the group spent more time on meetings and sprint reviews than on actual work, which can also be viewed as productive time, in the sense that the team saved time during the integration and testing stage by having a clear understanding of the tasks at hand.

4.6 KPI - Key Performance Indicator

The different KPIs measured during the project reflects the process in multiple ways. During the project three KPIs were tracked: team happiness, number of bugs per line of code and the velocity based on delivered user stories of each sprint.

4.6.1 Team Happiness

An evaluation on the Happiness KPI chart (see Appendix B) shows an overview of the team happiness during the project. The first weeks the average Happiness KPI affected by the issues with the server and defective MOPED. The team felt that little to no progress was made which clearly shows in the chart with the lowest average KPI of 3.4 in the first sprint.

Once problems began being solved the happiness KPI rose. The team learned that the MOPED's hardware issues did not have to affect the team's efficiency and that issue therefore affected the group less. In the last sprint, when testing could finally be done, the happiness skyrocketed for some of the members when results finally could be seen after weeks of work. The highest average happiness KPI could be found in this sprint, and was as high as 9.1.

Despite bad days, the happiness increased as the team learned to work together and found routines that worked. In the end, the team agreed on that this is what matters.With an average KPI close to 6, the team seems to have had more good days than bad days at least.

4.6.2 Bugs

The Bug per KLOC-KPI was a good idea in theory but did not work out as good as planned. The problem was that there were very few system-breaking bugs left when a sprint ended due to the small nature of the system. The complete system ended up at just about three KLOC, which meant that there was not that much code that did not run every single time integration tests were made for the system. This lead to that the majority of the problems that were left was style problems and bad syntax. The bugs-KPI would have been more relevant in a bigger project with more lines of code and with more people. If the KPI was to be used by Volvo for example, where the test-engineers and the developer are not the same people and where one single person cannot know all the code by themselves, it would have been more relevant.

4.6.3 Velocity

Throughout the project, the group used the notion of effort to determinate the range and time a user story or task would take. When planning the first sprint the group decided that the weekly total effort, also known as velocity, should be 50. The effort of 5 is one week work for one person. Using this system is a good way to estimate how long time a task would take.

But this did not work well in our group. The reasons for this were several; firstly, the group assumed that a effort value of five represented 20 hours of work. But written assignments, lectures, and meetings were not included in that time. In a typical working environment, with 40 working hours per week, the approach with effort seems usable. However, in a part time project with mandatory meetings and a lack of time to create a well working schedule, this strategy is not the best option. Looking at the time document, shown in Appendix E, the time spent on administrative work, meetings and other course-related events occupied on average half of the groups time spent.

Secondly, when the group estimated that an effort value of five represented 20 hours, that was extremely wrong. This is not only true because of the reason above, but also because many times the effort values were set lower than what was actually the case when recounting the time spent. In an attempt to combat this in the later sprints, the group made the decision to, if unsure of the work load, sometimes set a higher effort. This gave a bit of breathing room if a task proved to be more complex than first imagined.

Estimating the correct effort for a user story or task is perhaps one of the parts of the Scrum methodology that is easier when a team has collected some experience with the method. As first time users of Scrum, and while developing a client-server application for the first time, setting a correct estimation is not easy. In hindsight, the group should have extracted the hours that were spent not working towards the project owner from the set 20 hours. By knowing the hours that could be used to the Product Owner the group's effort towards tasks could have been more correct.

During the first sprint, the group almost met its mark in regards to velocity. However, it should be noted that many of the completed user stories were technical in its nature and targeted towards the team instead of the Product Owner. An example of this would be one user story which consisted of the whole team reading the provided documentation. The group knew that it was necessary to have knowledge of the system when starting out the project and therefore opted for that solution. This might be frowned upon in the Scrum methodology (and rightly so) since even though the velocity was high, very little actual value was provided to the Product Owner.

The velocity after sprint two, four and six was not at all near the goal. In sprint two the backlog was changed mid-sprint, leading to the group not being able to deliver full velocity due to lack of time. The group was aware that this would be a consequence when changing the backlog but opted to do it anyway; the group theorized that a new architecture could mean that the velocity would pick up. However, hardware problems caused the velocity of sprint four to be very low, meaning that the prediction failed.

The last sprint also went off its mark, this time on the positive side. This was due to overtime spent on trying to finish the prototype before the demonstration.

4.7 Sharing of MOPEDs

The project being organized so that twelve groups shared ten MOPEDs and that every MOPED where passed periodically to every group the setup was a huge detriment to the project as a whole. Some of the repeated problems was that the group received MOPEDs with sensors that were malfunctioned, cables that was wrongfully placed or simply missing. Screws where often passed along with the MOPEDs but no one did know how they should be put back.

With only two charges for the battery and groups that was not willing to share it was nearly impossible to work with the MOPED on the ground to test speed and steering and how the software worked on the MOPED.

Every time the group did get a new MOPED we almost always needed to find all the wrongs and work around them or trying to solve them. It resulted in hours of work that could have been spent on user stories and tasks. The group always received broken MOPEDs and it was hard to know if the problem was just the MOPEDs or if it was the group that did something to destroy the project not even knowing it.

4.8 Reflection on Used Practices

The first two weeks were mainly focused on the administrative parts of the project, which meant that almost no time was spent on coding. The smaller groups often sat together and worked, but this did not occur repeatedly. During the course of the project, more exactly at the fourth sprint-review (see appendix C.1), the team noticed that smaller groups working together more frequently had better communication and knew what was next, in comparison to the groups that worked separately.

This resulted in that further the project continued, the more the members tried to sit together and work. In this way, discussions were quicker and easier to have, and when questions arose, someone often had a quick reply.

At the fifth- and sixth sprint-review meeting, as seen in appendix C.1 and C.1, the team discussed that the focus was not always in the top. Sometimes, working together in one room would negatively impact productivity. If a member of the team lost focus and started doing something else, it was easy for others to do so as well, leading to the whole team working inefficiently. This was easily prevented by having short breaks now and then.

4.9 Evaluation of D1 - D4

During the project, the group was assigned four assignments. The first assignment D1 was about what the group has learned from the LEGO-exercise, the second assignment was about the team's vision about the end product, the third was a middle evaluation about the project and the fourth describes the demonstration at Lindholmen Science Park.

4.9.1 D1

The first assignment, D1, was a reflection from a Scrum exercise the group had done, which could be seen in appendix D. It was discussed in which way the group approached and applied Scrum, as well as deciding upon what KPIs that were to be tracked over the course of the project. While the reflection on Scrum was a good starting point, there was a lack of reflection on this matter later on in the project. Still, the group was able to keep the mindset outlined in the assignment, even though going back to discuss the application of Scrum regularly would have made it easier to apply the methodology continuously and stay true to the guidelines outlined in assignment D1.

4.9.2 D2

The second assignment, D2, was to define an initial product backlog as well as the group's vision for the project, shown in appendix D. As described earlier, the way of writing and assigning user stories was changed multiple times during the project to fit the needs of the group. The first backlog, although rewritten multiple times, gave a lot of insight into what was needed to be done in the project. Later versions of the backlog were based on the initial one written during this assignment.

4.9.3 D3

D3, the third assignment was a halfway evaluation reflecting on the work so far in terms of both process and product, seen in appendix D. In the assignment, the process is structured with the problems mentioned sprint by sprint (chronological). The team evaluates the process they made, an example of the process and what will be done next "The new server is written in Java and has the functionality to connect to both server and moped. This has not been tested and verified yet, however. The server also has methods for validating the MOPEDS up, and ports used in the connection. Our next goal is to receive data from the app and interpret it on the server, as well as develop a foundation for the adaptive cruise control which includes an algorithm that calculates what speed the MOPED should have to maintain a certain distance to the preceding vehicle".

In the assignment, the group describes the process and the work so far. The process for the Android application has a more exact evolution development described, then the other two parts. The process with the server had not be tested and the MOPEDs process was limited with the excuse that components were broken.

It was around this time the team started to think around the problems with the MOPED:s, this because the team could not deliver what the group required. At first, one of the group's problem was that "...the user stories were too big. While we had good results in terms of delivery, our communication was lackluster and suboptimal. This led to two groups trying to solve the same problem, and it affected the team negatively to a large extent". This problem was fixed "...the second sprint we quickly realized that the user stories we created limited our work, since they did not leave any room for creativity; we learned that this was an important factor since the project was heavily filled with problems. Having room for creativity meant that we could progress forward by working around the problems, even though the work we did might not have been strictly related to a specific user story. The current user stories gave no room to expand or build upon due to their specific and often too technical description. We had a discussion about it, and we agreed upon changing how to write our user stories to give us more room to breathe if something goes wrong."

4.9.4 D4

The D4 assignment was the final presentation at Lindholmen Science Park. The working prototype was to demonstrate it's platooning capabilities along with four other MOPEDs implemented by the other groups. Unfortunately, the group's prototype suffered several problems during the demo. Occasionally, the MOPED stopped following the preceding car and drove past it whenever the distance between them grew too large. This was due to a hardware malfunction in the engine of the MOPED that affected the engine's ability to drive at slow speeds. Because of this, when the leading car drove at a speed slower than planned the MOPED did not respond until the proceeding MOPED was about one meter away. When the leader of the line stopped, the group's MOPED instead had accelerated to quickly, and ended up driving past the other vehicles. The reason why the group's MOPED was programmed this way was that the group had another vision for how the platooning would work. The group's system was developed for a scenario on a highway, where cars were driving smoothly with small changes in speed. The team's ACC was created based on information received at a meeting with the product owner and based on a real-life scenario where this technology is applied by VOLVO today. A Scrum of Scrum meeting decided that the speed of the colon would be 75 cm/s, but the demo was not carried out to the decided specifications. The lead MOPED of the platooning group instead drove similarly to a city traffic scenario with frequent stops, low velocity, and sharp turns. This scenario, in combination with the engine-issues, was not ideal for the ACC the group had developed.

5 Conclusion

Based on the problems that occurred during the course of the project, the group learned many things. One of the most important lessons was finding different ways to work around problems, instead of trying to solve them as-is, or letting them stop the development process completely.

When the MOPED problems started to resemble a recurring issue, the group should have stopped working under the assumption that the hardware was fully functional. Instead, the group should have considered alternate approached to reach their goal, such as developing a simulator right from the start. The group also could have designed an overall architecture sooner, along with creating broader user stories at the beginning which could have made the team progress faster. Broader, more vague tasks invokes creativity and finding ways to work around problems that might arise and can not be solved at that specific time.

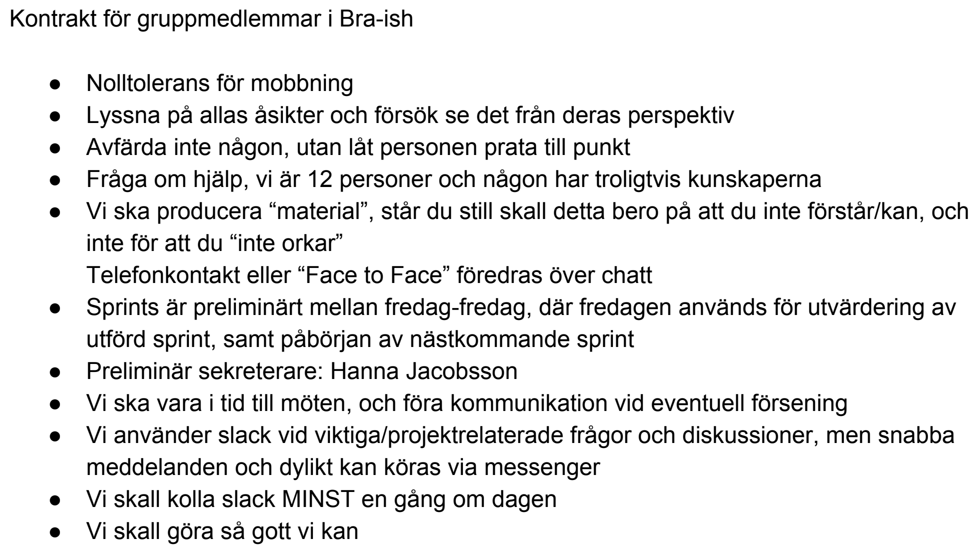
Working in a group this big proved to be a challenge. Good and clear communication proved to be key for the group to work efficiently. One lesson the group learned, was to tackle big obstacles in a large project. Without a working prototype, it was important to make everything modular and break things down into small parts. To vertically slice user stories was also important when working on continuous development.

Even though the group specified vertically sliced user stories, it was sometimes difficult to follow the process of the user stories due to lack of documentation. When a team member or subgroup documents its process and progress, another member or subgroup could understand, and continue working on the same task. This proved to be a correct assessment, as shown when the group improved their way of handling information connected to user stories and tasks.

During the project it would have been good if the Scrum master had more continuous overview of the project, and had a better understanding of how far along all the subgroups were. This would mean that the team could balance the workload better, and prioritize tasks that were needed by a certain subgroup for them to be able to continue development.

To summarize: the team was able to provide a working prototype for the demonstration held at the end of the project. In regards to working with Scrum, it is clear that a certain amount of experience is required in order to fully understand the methodology. Moreover, it is obvious that it is intended for a full-time, long running project rather than a part-time shorter project like this one. Nevertheless, the project provided a good way to gain experience with and learn about agile development processes which, after all, are central to professional software development today.

Appendix A Social contract



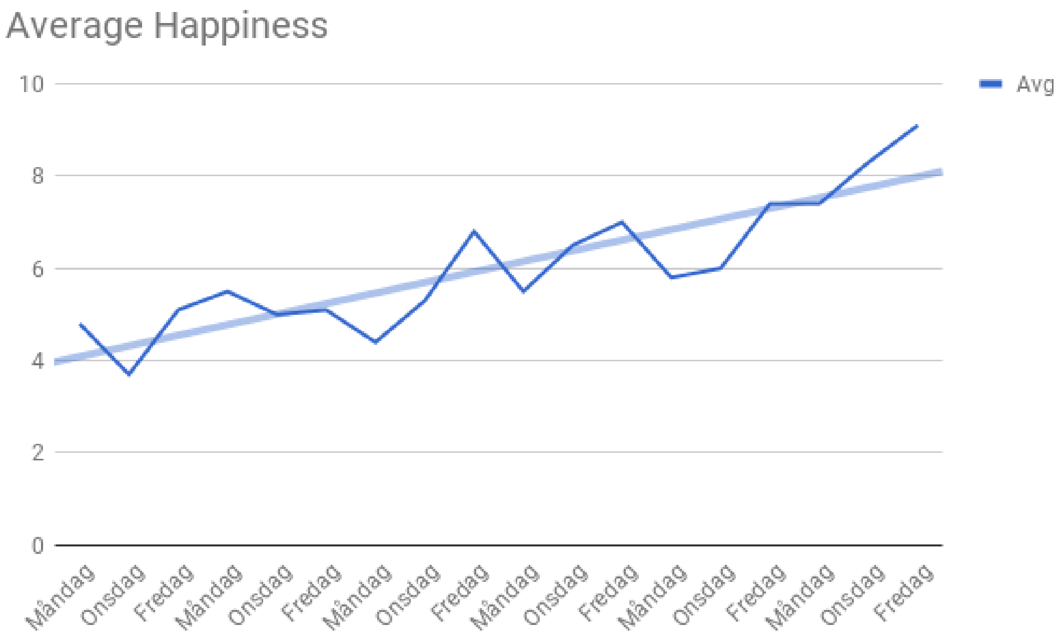
Kontrakt för gruppmedlemmar i Bra-ish

- Nolltolerans för mobbning
- Lyssna på allas åsikter och försök se det från deras perspektiv
- Avfärda inte någon, utan låt personen prata till punkt
- Fråga om hjälp, vi är 12 personer och någon har troligtvis kunskaperna
- Vi ska producera “material”, står du still skall detta bero på att du inte förstår/kan, och inte för att du “inte orkar”
  Telefonkontakt eller “Face to Face” föredras över chatt
- Sprints är preliminärt mellan fredag-fredag, där fredagen används för utvärdering av utförd sprint, samt påbörjan av nästkommande sprint
- Preliminär sekreterare: Hanna Jacobsson
- Vi ska vara i tid till möten, och föra kommunikation vid eventuell försening
- Vi använder slack vid viktiga/projektrelaterade frågor och diskussioner, men snabba meddelanden och dylikt kan köras via messenger
- Vi skall kolla slack MINST en gång om dagen
- Vi skall göra så gott vi kan

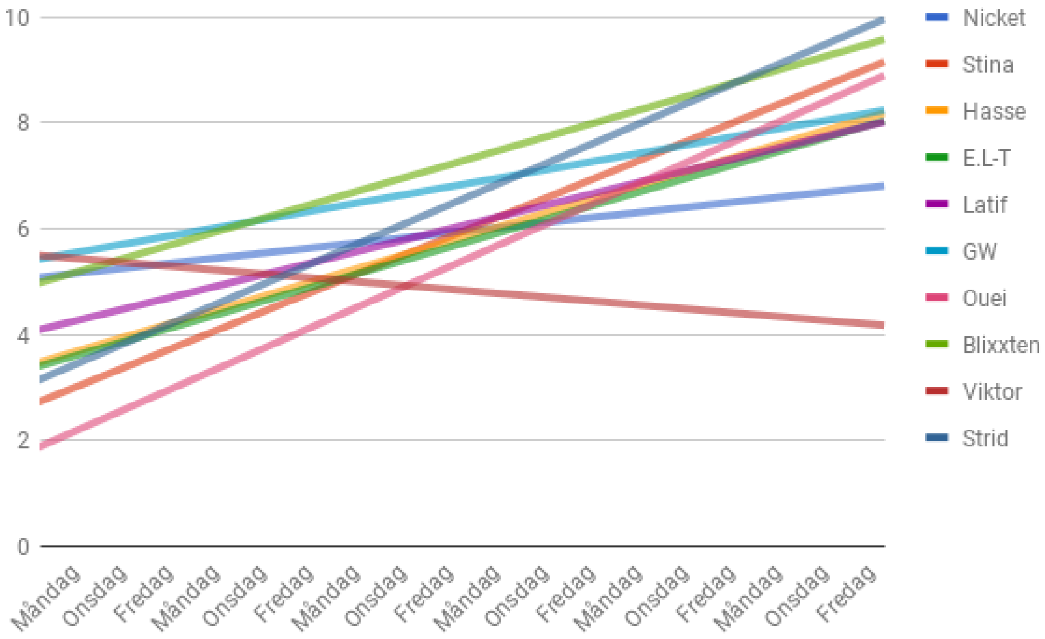
Appendix B KPI Charts

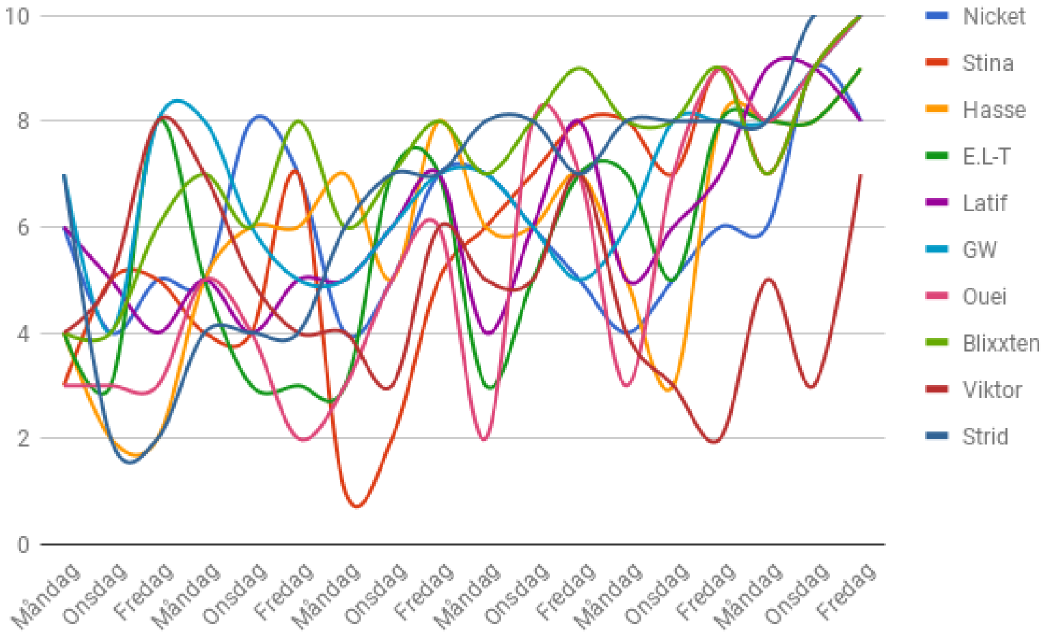
B.1 Happiness KPI

B.1.1 Average

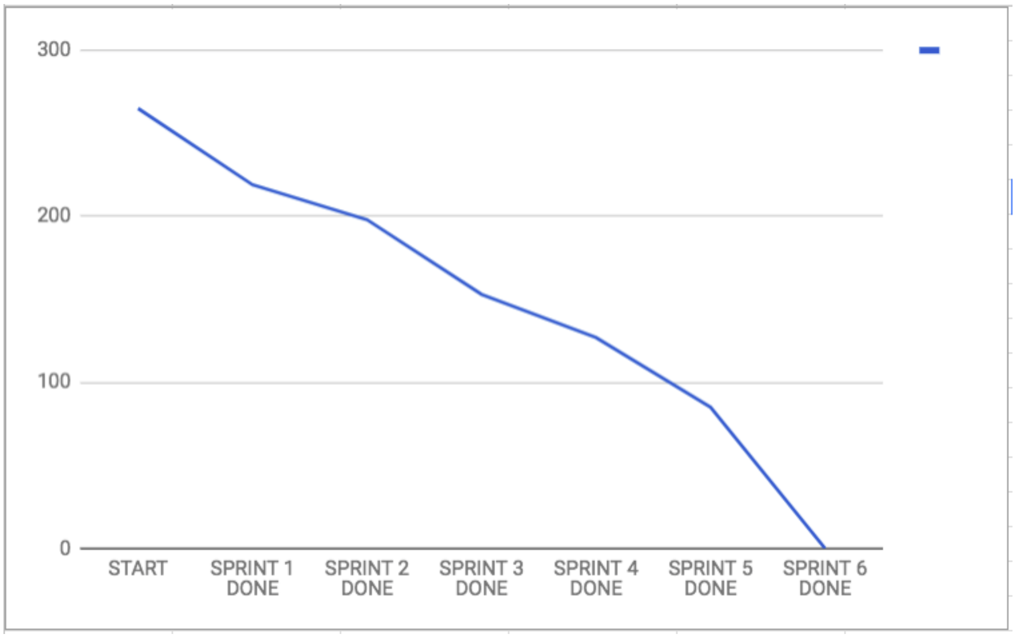


B.1.2 Member averages



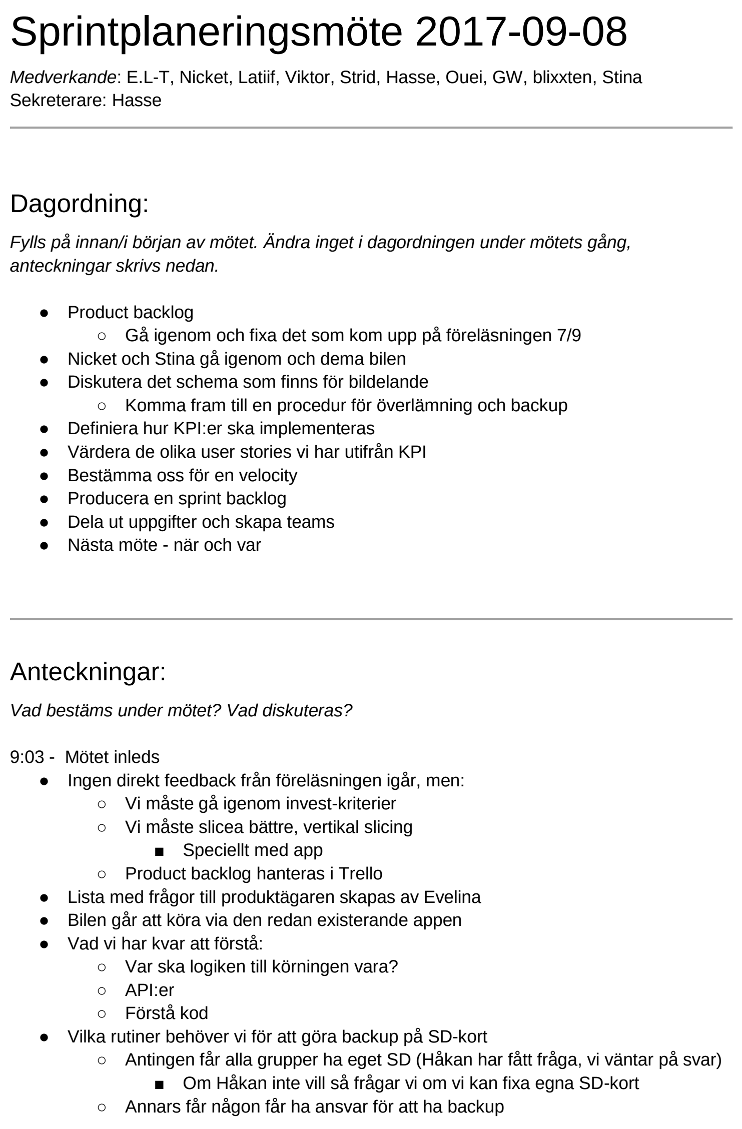


B.2 Velocity


300
200
100
0
START
SPRINT 1 DONE
SPRINT 2 DONE
SPRINT 3 DONE
SPRINT 4 DONE
SPRINT 5 DONE
SPRINT 6 DONE

Appendix C Meetings

C.1 Sprint 1 Planning



Sprintplaneringsmöte 2017-09-08

Medverkande: E.L-T, Nicket, Latiif, Viktor, Strid, Hasse, Ouei, GW, blixxten, Stina
Sekreterare: Hasse

Dagordning:

Fylls på innan/i början av mötet. Ändra inget i dagordningen under mötets gång, anteckningar skrivs nedan.

- Product backlog
  - Gå igenom och fixa det som kom upp på föreläsningen 7/9
- Nicket och Stina gå igenom och dema bilen
- Diskutera det schema som finns för bildelande
  - Komma fram till en procedur för överlämning och backup
- Definiera hur KPI:er ska implementeras
- Värdera de olika user stories vi har utifrån KPI
- Bestämma oss för en velocity
- Producera en sprint backlog
- Dela ut uppgifter och skapa teams
- Nästa möte - när och var

Anteckningar:

Vad bestäms under mötet? Vad diskuteras?

9:03 - Mötet inleds

- Ingen direkt feedback från föreläsningen igår, men:
  - Vi måste gå igenom invest-kriterier
  - Vi måste slicea bättre, vertikal slicing
    - Speciellt med app
  - Product backlog hanteras i Trello
- Lista med frågor till produktägaren skapas av Evelina
- Bilen går att köra via den redan existerande appen
- Vad vi har kvar att förstå:
  - Var ska logiken till körningen vara?
  - API:er
  - Förstå kod
- Vilka rutiner behöver vi för att göra backup på SD-kort
  - Antingen får alla grupper ha eget SD (Håkan har fått fråga, vi väntar på svar)
    - Om Håkan inte vill så frågar vi om vi kan fixa egna SD-kort
  - Annars får någon får ha ansvar för att ha backup

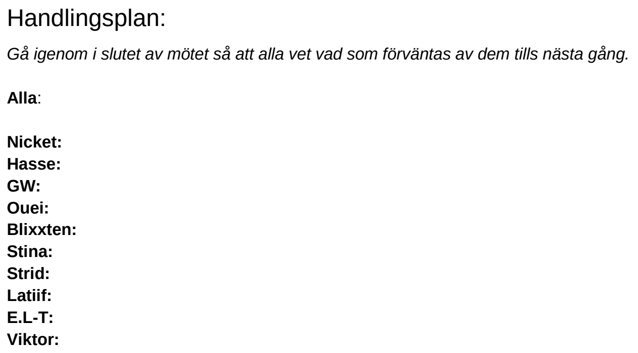
- Bildelningsschema - se bilaga
- Vid bilbyte kontaktar andra gruppen Niclas via Slack och han ser till att gruppen får bilen

Rast 10:51 till 11:10

- KPI - hur ska vi arbeta med det?
    - Entusiasm i gruppen
        - Inför varje möte - alla gör en check-in
    - Velocity
        - Fibonacci brukar användas i scrum - vi kör på det (bestämt av omröstning)
        - Måste vara medvetna och konsekventa för att kunna jämföra bra
        - Håkan har blivit frågad hur det ska hanteras - vi väntar på svar
    - Buggar
        - Hur mäter vi aktiva buggar?
        - Git Issues + Trello
        - Hur kopplar vi buggar till user stories?
        - Bugg ≠ halvfärdiga saker
        - Separat Bugg Backlog och Project Backlog
    - Kommunikation
        - Diskutera fråga varje möte
        - Hur går det
            - Under möten
            - Chatter/Slack
        - Använda siffror/skala? Skapa graf efter
    - Hur många som är valda vs. hur mycket som faktiskt genomförs
- Daily Scrum i första rasten på varje föreläsning
- Om vi får en kritisk bugg - får vi ta den direkt eller måste vi vänta?
    - Vi behöver ej vänta
    - Jag upplever konsensus i frågan
- Diskussion kring ändring i backlog samt av velocity
    - Finns det fall då man kan ändra båda

Lunch 11:45 - 13:00

- KPI sätts med hjälp av PlanITpoker
- Visionslista:
    - Gyrostyrning via mobil
    - Krypto
    - Säkerhetszon
    - Gå in i, följa, gå ur
    - Bra UX



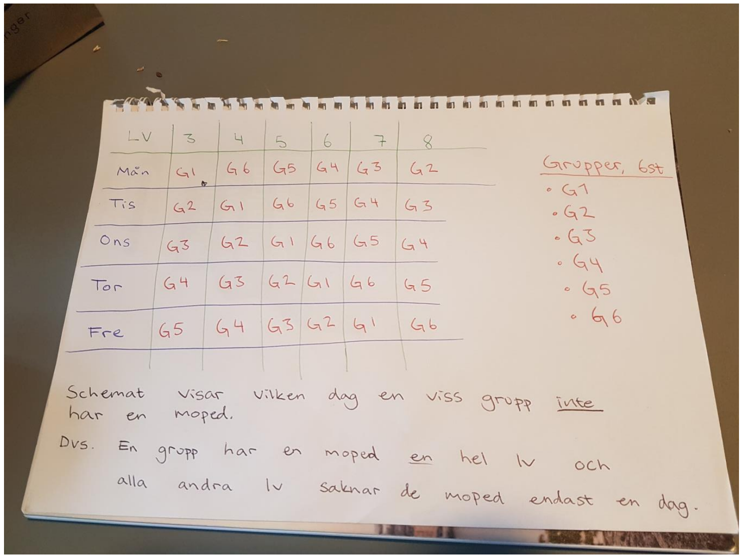
Handlingsplan:

Gå igenom i slutet av mötet så att alla vet vad som förväntas av dem tills nästa gång.

Alla:

Nicket:
Hasse:
GW:
Ouei:
Blixxten:
Stina:
Strid:
Latiif:
E.L-T:
Viktor:

Bilagor



LV	3	4	5	6	7	8
Mån	G1	G6	G5	G4	G3	G2
Tis	G2	G1	G6	G5	G4	G3
Ons	G3	G2	G1	G6	G5	G4
Tor	G4	G3	G2	G1	G6	G5
Fre	G5	G4	G3	G2	G1	G6

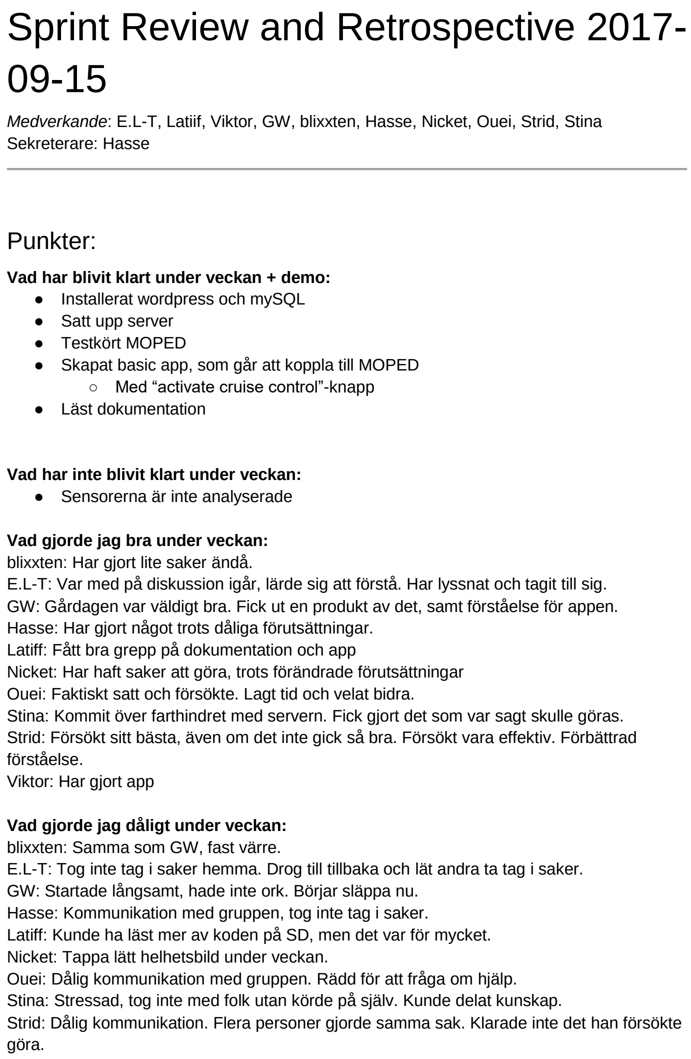
Grupper, 6st

- G1
- G2
- G3
- G4
- G5
- G6

Schemat visar vilken dag en viss grupp inte har en moped.

Dvs. En grupp har en moped en hel lv och alla andra lv saknar de moped endast en dag.

C.2 Sprint 1 Review and Retrospective



Sprint Review and Retrospective 2017-09-15

Medverkande: E.L-T, Latiif, Viktor, GW, blixxten, Hasse, Nicket, Ouei, Strid, Stina
Sekreterare: Hasse

Punkter:

Vad har blivit klart under veckan + demo:

- Installerat wordpress och mySQL
- Satt upp server
- Testkört MOPED
- Skapat basic app, som går att koppla till MOPED
  - Med "activate cruise control"-knapp
- Läst dokumentation

Vad har inte blivit klart under veckan:

- Sensorerna är inte analyserade

Vad gjorde jag bra under veckan:

blixxten: Har gjort lite saker ändå.
E.L-T: Var med på diskussion igår, lärde sig att förstå. Har lyssnat och tagit till sig.
GW: Gårdagen var väldigt bra. Fick ut en produkt av det, samt förståelse för appen.
Hasse: Har gjort något trots dåliga förutsättningar.
Latiff: Fått bra grepp på dokumentation och app
Nicket: Har haft saker att göra, trots förändrade förutsättningar
Ouei: Faktiskt satt och försökte. Lagt tid och velat bidra.
Stina: Kommit över farthindret med servern. Fick gjort det som var sagt skulle göras.
Strid: Försökt sitt bästa, även om det inte gick så bra. Försökt vara effektiv. Förbättrad förståelse.
Viktor: Har gjort app

Vad gjorde jag dåligt under veckan:

blixxten: Samma som GW, fast värre.
E.L-T: Tog inte tag i saker hemma. Drog till tillbaka och lät andra ta tag i saker.
GW: Startade långsamt, hade inte ork. Börjar släppa nu.
Hasse: Kommunikation med gruppen, tog inte tag i saker.
Latiff: Kunde ha läst mer av koden på SD, men det var för mycket.
Nicket: Tappa lätt helhetsbild under veckan.
Ouei: Dålig kommunikation med gruppen. Rädd för att fråga om hjälp.
Stina: Stressad, tog inte med folk utan körde på själv. Kunde delat kunskap.
Strid: Dålig kommunikation. Flera personer gjorde samma sak. Klarade inte det han försökte göra.

Viktor: Gjorde mindre i början på veckan än vad han skulle kunna gjort.

Vad har vi gjort bra under veckan:

- Fick saker producerade trots dåliga förutsättningar
- Alla hade en uppgift
- Arbetet gav faktiskt BRA resultat, inget halvdant
- Vi lyckades sätta oss och jobba
- När vi väl kom igång gick det snabbt
- Vi vet mer om hur vi ska jobba

Vilka problem har vi haft och hur löstes dessa:

- Servern
  - Lösning: MASSOR av tid + trial and error
  - Stoppade allt och alla - sensorer kunde inte göras
  - Skapade stress i gruppen

Vad skulle vi kunna gjort bättre?

- Kommunikation
- För stora user stories
  - dela upp i tasks istället
- Folk är sena till möten
- För få Daily Scrum
- För få personer på Lindholmen

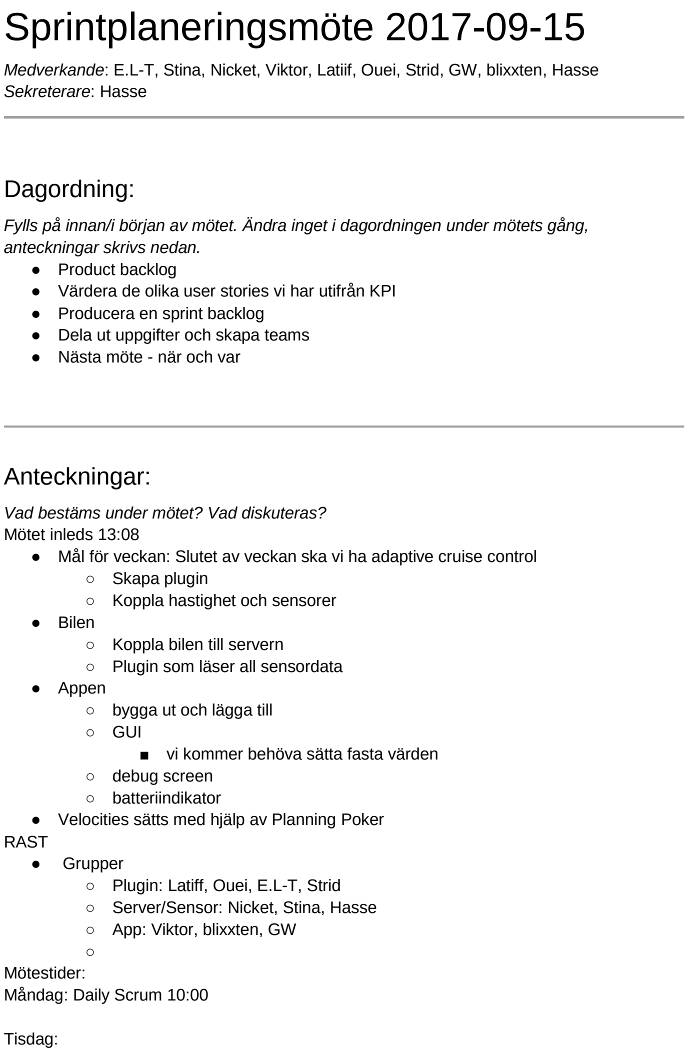
Ändringar till nästa sprint:

- Bestäm möten för hela veckan på fredag
- Fler Daily Scrum: in person måndag, onsdag, fredag. Mindre grupper tisdag och torsdag.
- Scrum master i varje mindre grupp - dessa följer med till Lindholmen om de har frågor
- Testning - en person är testansvarig

Övrigt:

- Dokumentation - borde vi börja med det?
  - Javadoc?
  - SDD
  - Wiki
- Rapport
  - Dokument där alla kan skriva personliga tankar

C.3 Sprint 2 Planning



Sprintplaneringsmöte 2017-09-15

Medverkande: E.L-T, Stina, Nicket, Viktor, Latiif, Ouei, Strid, GW, blixxten, Hasse
Sekreterare: Hasse

Dagordning:

Fylls på innan/i början av mötet. Ändra inget i dagordningen under mötets gång, anteckningar skrivs nedan.

- Product backlog
- Värdera de olika user stories vi har utifrån KPI
- Producera en sprint backlog
- Dela ut uppgifter och skapa teams
- Nästa möte - när och var

Anteckningar:

Vad bestäms under mötet? Vad diskuteras?
Mötet inleds 13:08

- Mål för veckan: Slutet av veckan ska vi ha adaptive cruise control
    - Skapa plugin
    - Koppla hastighet och sensorer
- Bilen
    - Koppla bilen till servern
    - Plugin som läser all sensordata
- Appen
    - bygga ut och lägga till
    - GUI
        - vi kommer behöva sätta fasta värden
    - debug screen
    - batteriindikator
- Velocities sätts med hjälp av Planning Poker

RAST

- Grupper
    - Plugin: Latiff, Ouei, E.L-T, Strid
    - Server/Sensor: Nicket, Stina, Hasse
    - App: Viktor, blixxten, GW
    -

Mötestider:
Måndag: Daily Scrum 10:00

Tisdag:

Onsdag:
Torsdag:
Fredag:

Handlingsplan:

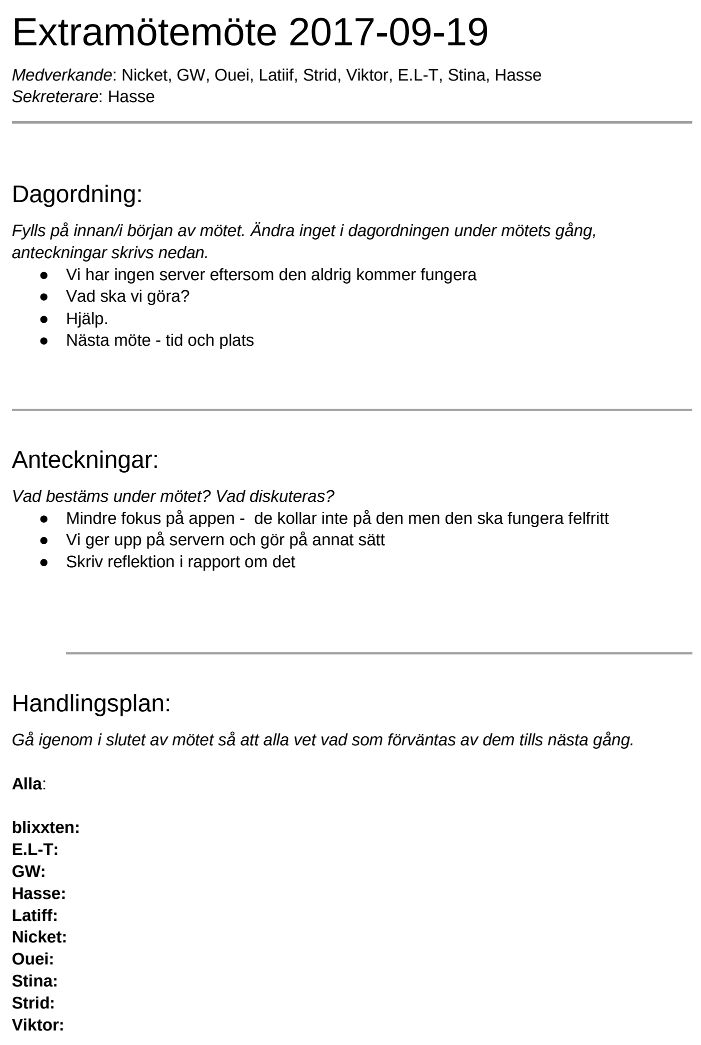
Gå igenom i slutet av mötet så att alla vet vad som förväntas av dem tills nästa gång.

Alla: Alla grupper tar belsut om vem som har vilken roll

blixxten:
E.L-T:
GW:
Hasse:
Latiff:
Nicket:
Ouei:
Stina:
Strid:
Viktor:

Bilagor:

C.4 Sprint 2



Extramötemöte 2017-09-19

Medverkande: Nicket, GW, Ouei, Latiif, Strid, Viktor, E.L-T, Stina, Hasse
Sekreterare: Hasse

Dagordning:

Fylls på innan/i början av mötet. Ändra inget i dagordningen under mötets gång, anteckningar skrivs nedan.

- Vi har ingen server eftersom den aldrig kommer fungera
- Vad ska vi göra?
- Hjälp.
- Nästa möte - tid och plats

Anteckningar:

Vad bestäms under mötet? Vad diskuteras?

- Mindre fokus på appen - de kollar inte på den men den ska fungera felfritt
- Vi ger upp på servern och gör på annat sätt
- Skriv reflektion i rapport om det

Handlingsplan:

Gå igenom i slutet av mötet så att alla vet vad som förväntas av dem tills nästa gång.

Alla:

blixxten:
E.L-T:
GW:
Hasse:
Latiff:
Nicket:
Ouei:
Stina:
Strid:
Viktor:

Bilagor:

C.5 Sprint 2 Review and Retrospective

Sprint Review and Retrospective 2017-09-22

Medverkande: Stina, Viktor, Latiif, blixxten, Ouei, GW (20 minuter sen)
Sekreterare: blixxten

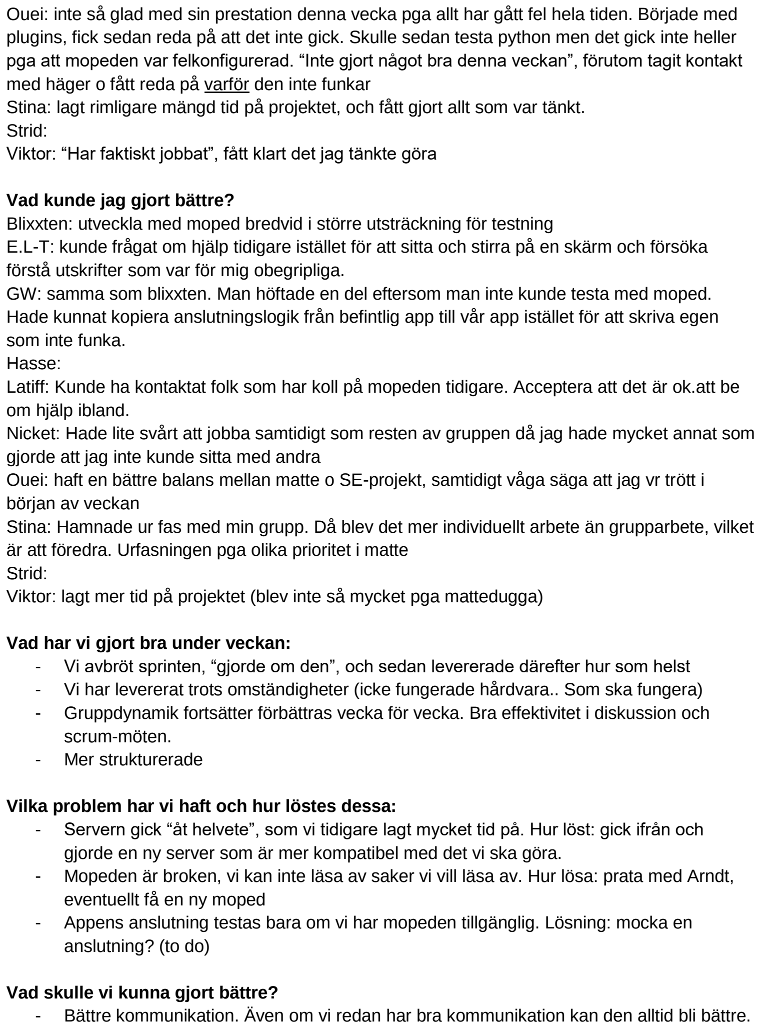
Punkter:

Vad har blivit klart under veckan + demo:
- styrning från appen funkar (höger, fram, bak, vänster) och kunna reglera vinklar hos dessa också.

- Manuell anslutning mellan moped och app fungerar.
  - DEMO: Det fanns en anslutningsvy med en indikation på att anslutning finns. En vy för batteriinfo finns också, men inte med funktioner än.
- Reflektinosrapport om bytet i serverutveckling
- Koppla moped till server (gammla server)
- Lagt in ip-validering i server (nya servern)
- Nya server testad

Vad har inte blivit klart under veckan:
- Analys av sensor
  - Anledning: (tidsbrist och) att moped inte funkar som den ska
- Debug screen får ingen data, men den är gjord
- Kunna läsa sensordata via nätverkssockets

Vad gjorde jag bra under veckan:
Bblixxten: Har varit här med, även om det är långt ifrån och han inte har föreläsningar. Nöjd med sig själv att han är här. Nöjd med att appen är simpel men fullt funktionerad
E.L-T:även om saker inte fungerat så har jag medverkat en del. Dock är jag inte nöjd med min presentation denna veckan då inget har fungerat. Först ej plugins, sedan att mopeden var fel konfigurerad så köra Python koden funkade inte heller.
GW: Tycker att, prestationsmässigt, att jag förbättrat med jämfört med förra veckan. Mer jobb denna veckan. Nöjd med appen, speciellt nu när ansltningen är löst (den satt jag väldigt mycket o jobbade med, slet mig i håret)
Hasse:
Latiff: hade tagit reda på att felet ligger i mopeden. “Run.py” måste funka. Det går att skapa connection mellan moped o server. bra!
Nicket:Jobbat bra och verkligen haft koll på helheten



Ouei: inte så glad med sin prestation denna vecka pga allt har gått fel hela tiden. Började med plugins, fick sedan reda på att det inte gick. Skulle sedan testa python men det gick inte heller pga att mopeden var felkonfigurerad. “Inte gjort något bra denna veckan”, förutom tagit kontakt med häger o fått reda på varför den inte funkar
Stina: lagt rimligare mängd tid på projektet, och fått gjort allt som var tänkt.
Strid:
Viktor: “Har faktiskt jobbat”, fått klart det jag tänkte göra

Vad kunde jag gjort bättre?
Blixxten: utveckla med moped bredvid i större utsträckning för testning
E.L-T: kunde frågat om hjälp tidigare istället för att sitta och stirra på en skärm och försöka förstå utskrifter som var för mig obegripliga.
GW: samma som blixxten. Man höftade en del eftersom man inte kunde testa med moped. Hade kunnat kopiera anslutningslogik från befintlig app till vår app istället för att skriva egen som inte funka.
Hasse:
Latiff: Kunde ha kontaktat folk som har koll på mopeden tidigare. Acceptera att det är ok.att be om hjälp ibland.
Nicket: Hade lite svårt att jobba samtidigt som resten av gruppen då jag hade mycket annat som gjorde att jag inte kunde sitta med andra
Ouei: haft en bättre balans mellan matte o SE-projekt, samtidigt våga säga att jag vr trött i början av veckan
Stina: Hamnade ur fas med min grupp. Då blev det mer individuellt arbete än grupparbete, vilket är att föredra. Urfasningen pga olika prioritet i matte
Strid:
Viktor: lagt mer tid på projektet (blev inte så mycket pga mattedugga)

Vad har vi gjort bra under veckan:

- Vi avbröt sprinten, “gjorde om den”, och sedan levererade därefter hur som helst
- Vi har levererat trots omständigheter (icke fungerade hårdvara.. Som ska fungera)
- Gruppdynamik fortsätter förbättras vecka för vecka. Bra effektivitet i diskussion och scrum-möten.
- Mer strukturerade

Vilka problem har vi haft och hur löstes dessa:

- Servern gick “åt helvete”, som vi tidigare lagt mycket tid på. Hur löst: gick ifrån och gjorde en ny server som är mer kompatibel med det vi ska göra.
- Mopeden är broken, vi kan inte läsa av saker vi vill läsa av. Hur lösa: prata med Arndt, eventuellt få en ny moped
- Appens anslutning testas bara om vi har mopeden tillgänglig. Lösning: mocka en anslutning? (to do)

Vad skulle vi kunna gjort bättre?

- Bättre kommunikation. Även om vi redan har bra kommunikation kan den alltid bli bättre.

- Samarbete bättre mellan grupper. Om vi gjort det hade app-lag och server-lag hunnit med att upprätta anslutning däremellan.
- Sprint-kort ska vara mer öppna så att man inte är låst som utvecklare.
- Förbättra user stories så att de uppmuntrar mer kreativt arbete.

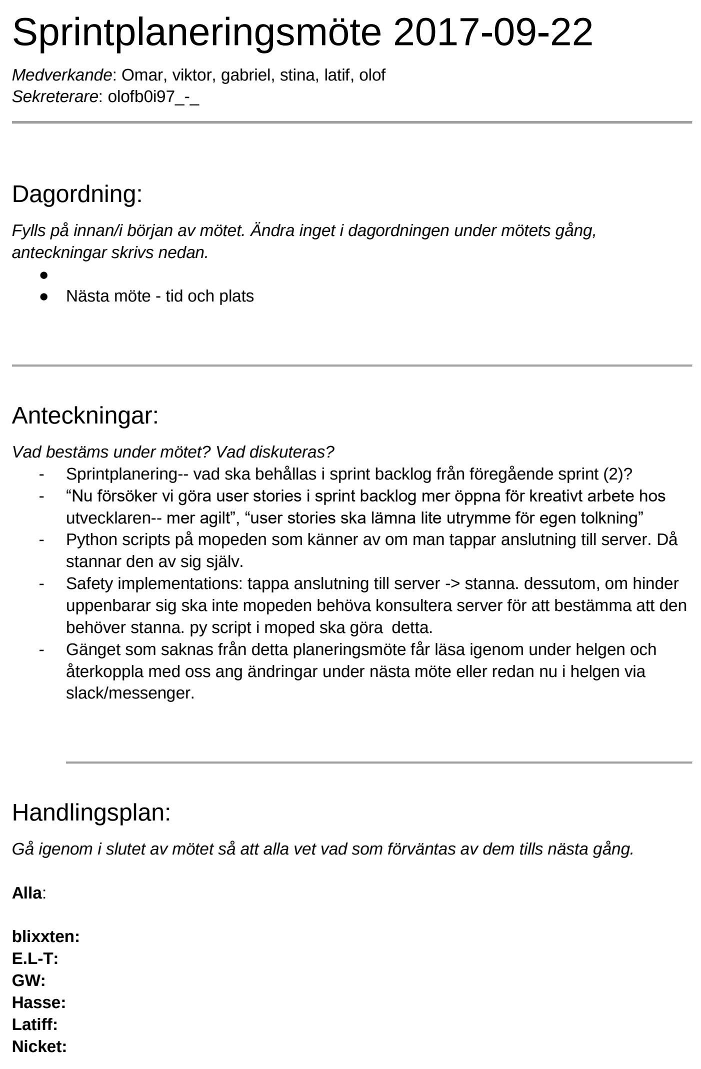
Ändringar till nästa sprint:

- När vi märker att något går fel, tveka inte på att kontakta Arndt eller andra personer som kan sin grej
- Göra mer öppna user stories enl beskrivning ovan

Övrigt:

Bilagor:

C.6 Sprint 3 Planning



Sprintplaneringsmöte 2017-09-22

Medverkande: Omar, viktor, gabriel, stina, latif, olof
Sekreterare: olofb0i97_-_

Dagordning:

Fylls på innan/i början av mötet. Ändra inget i dagordningen under mötets gång, anteckningar skrivs nedan.

-
- Nästa möte - tid och plats

Anteckningar:

Vad bestäms under mötet? Vad diskuteras?

- Sprintplanering-- vad ska behållas i sprint backlog från föregående sprint (2)?
- “Nu försöker vi göra user stories i sprint backlog mer öppna för kreativt arbete hos utvecklaren-- mer agilt”, “user stories ska lämna lite utrymme för egen tolkning”
- Python scripts på mopeden som känner av om man tappar anslutning till server. Då stannar den av sig själv.
- Safety implementations: tappa anslutning till server -> stanna. dessutom, om hinder uppenbarar sig ska inte mopeden behöva konsultera server för att bestämma att den behöver stanna. py script i moped ska göra detta.
- Gänget som saknas från detta planeringsmöte får läsa igenom under helgen och återkoppla med oss ang ändringar under nästa möte eller redan nu i helgen via slack/messenger.

Handlingsplan:

Gå igenom i slutet av mötet så att alla vet vad som förväntas av dem tills nästa gång.

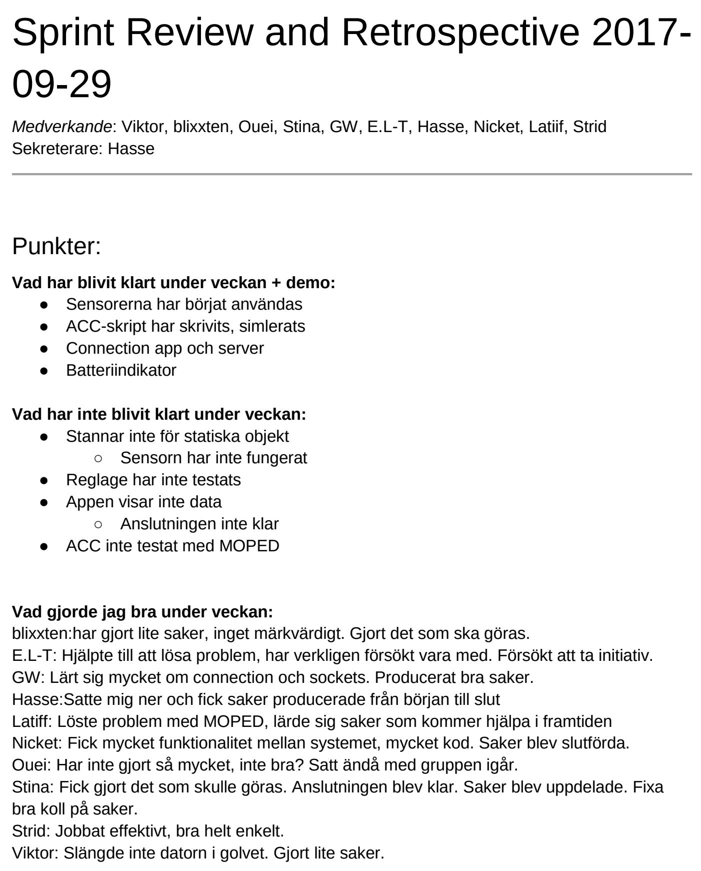
Alla:

blixxten:
E.L-T:
GW:
Hasse:
Latiff:
Nicket:

Ouei:
Stina:
Strid:
Viktor:

Bilagor:

C.7 Sprint 3 Review and Retrospective



Sprint Review and Retrospective 2017-09-29

Medverkande: Viktor, blixxten, Ouei, Stina, GW, E.L-T, Hasse, Nicket, Latiif, Strid
Sekreterare: Hasse

Punkter:

Vad har blivit klart under veckan + demo:

- Sensorerna har börjat användas
- ACC-skript har skrivits, simlerats
- Connection app och server
- Batteriindikator

Vad har inte blivit klart under veckan:

- Stannar inte för statiska objekt
  - Sensorn har inte fungerat
- Reglage har inte testats
- Appen visar inte data
  - Anslutningen inte klar
- ACC inte testat med MOPED

Vad gjorde jag bra under veckan:
blixxten:har gjort lite saker, inget märkvärdigt. Gjort det som ska göras.
E.L-T: Hjälpte till att lösa problem, har verkligen försökt vara med. Försökt att ta initiativ.
GW: Lärt sig mycket om connection och sockets. Producerat bra saker.
Hasse:Satte mig ner och fick saker producerade från början till slut
Latiff: Löste problem med MOPED, lärde sig saker som kommer hjälpa i framtiden
Nicket: Fick mycket funktionalitet mellan systemet, mycket kod. Saker blev slutförda.
Ouei: Har inte gjort så mycket, inte bra? Satt ändå med gruppen igår.
Stina: Fick gjort det som skulle göras. Anslutningen blev klar. Saker blev uppdelade. Fixa bra koll på saker.
Strid: Jobbat effektivt, bra helt enkelt.
Viktor: Slängde inte datorn i golvet. Gjort lite saker.

Vad gjorde jag mindre bra under veckan:
blixxten: såg inte viktig flaskhals
E.L-T: Lät problem med MOPEDen stå i vägen när annat kanske kunde ha gjorts
GW: Dåligt fokus och dålig koll på trello
Hasse: Kom inte på möte med riktigt dålig ursäkt
Latiff: Blev distraherad, hade ingen koll på trello
Nicket: Dålig kommunikation.
Ouei: Mått dåligt, ur synk. Fastnat i problem. Låtit frustration ta över.
Stina: Ansträngt sig mer för att förbättra kommunikation i gruppen
Strid: Uppdatera trello
Viktor: tog inte ansvar för att få saker gjort efter att ha mått dåligt

Vad har vi gjort bra under veckan:

- Fick mycket testat och avcheckat
- Bra kommunikation mellan grupperna
- Skrivit utvärdering snabbt innan det glömdes

Vilka problem har vi haft och hur löstes dessa:

- Hårdvaruproblem
- Svårt med MOPED-byte
- Har inte lösts, väntar på tillfälle (måndag 1/10-17) på att få det fixat

Vad skulle vi kunna gjort bättre?

- En grupp har inte fungerat
- Osynkade på Trello
- Prata med handledare och arnt snabbare.
- Arbetsfördelningen mellan grupperna
- Testning har tagit lång tid

Ändringar till nästa sprint:

- Andra grupper
- Vara mer aktiva i chatter
- Berätta för folk om man hitta något relevant på klass slacken
- Alla skriver egen daily scrum och berättar om det under möten
- Möte (minst) en gång i veckan för att genomföra tester med alla grupper.
- Alla måste ansvara för att genomföra egna tester.
- Skriva kommentar till trello kort om vad/hur/när man har testat.

Övrigt:

Schema:
9.00 - 9.15 Daily Scrum

9.15 - 9.30 Paus
9.30 - 10.30 Eget arbete
10.30 - 11.45 Sprint Review
11.45 - 13.00 Lunch
13.00 - 14.00 Sprintplanering
14.00- 14.15 Paus
14.15 - Klar Sprintplanering

Bilagor:

C.8 Sprint 4 Planning

Sprintplaneringsmöte 2017-09-29

Medverkande: blixxten, GW, Stina, Ouei, E.L-T, Latiif, Hasse, Nicket, Strid
Sekreterare: Hasse

Dagordning:

Fylls på innan/i början av mötet. Ändra inget i dagordningen under mötets gång, anteckningar skrivs nedan.

- Ta hand om product backlog
- Värdera user stories i planningpoker
- Bestämma kommande möten - tid och plats

Anteckningar:

Vad bestäms under mötet? Vad diskuteras?

- Vissa kort skrivs om, delas upp eller specificeras
- Om vi tappar connection med servern så ska den stanna - detta måste finnas på MOPEDen
- Även tvärnit om saker dyker upp väldigt nära - säkerhetszon?
- Kommande vecka kanske vi ska satsa på att göra kameran användbar
- En person på apputveckling - folk behövs till kameran
- Säkerhet sparas till veckan efter

Nästa veckas mål:

Platooning (***Plootoning<3***):

- ACC - hastigheten
    - bestämma hastighet
    - F-sensor
    - Appkoppling
- Styrning, centrering
    - Bildinläsning
        - Skicka till server
            - Komprimering
        - Bildigenkänning
            - Hitta symbol att använda
            - Centrera
                - Styrning
    -

- Nya grupper:
  - ACC (2) - Nicket och Hasse (parallellt med bildigenkänning)
  - Bildigenkänning (3) - Viktor, Strid och GW
  - MOPED (3) - Latiff, E.L-T och Ouei
  - App (1) - blixxten
  - Kommunikation (1) - Stina

Mötestider:
Måndag 12:45
Onsdag 9:30
Fredag: 9:00

Handlingsplan:

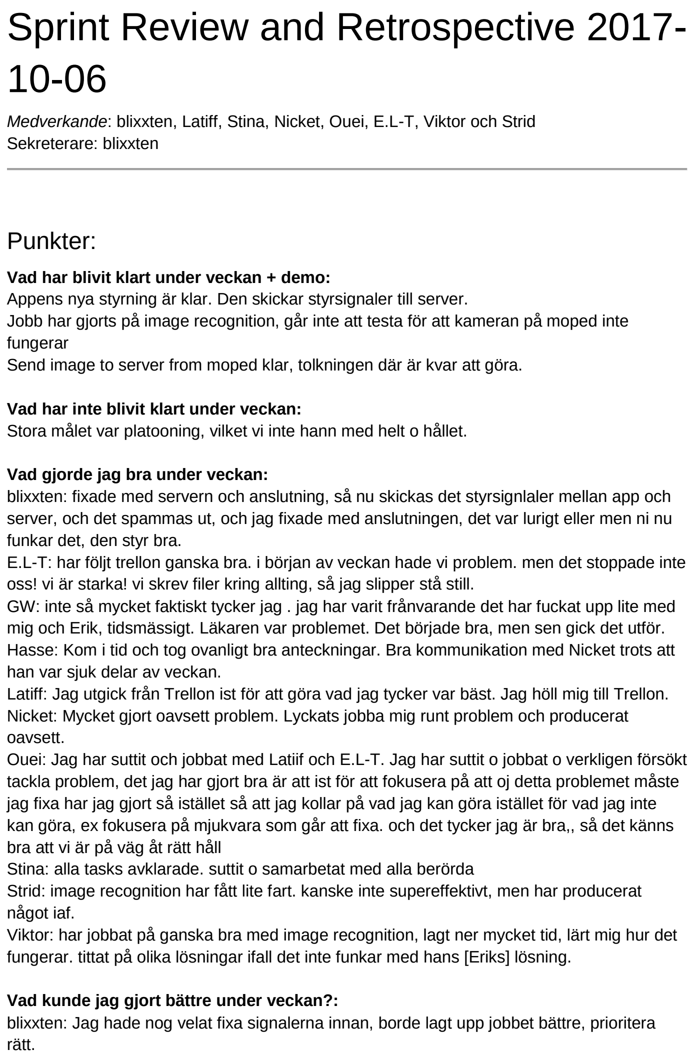
Gå igenom i slutet av mötet så att alla vet vad som förväntas av dem tills nästa gång.

Alla:

blixxten:
E.L-T:
GW:
Hasse:
Latiif:
Nicket:
Ouei:
Stina:
Strid:
Viktor:

Bilagor:

C.9 Sprint 4 Review and Retrospective



Sprint Review and Retrospective 2017-10-06

Medverkande: blixxten, Latiff, Stina, Nicket, Ouei, E.L-T, Viktor och Strid
Sekreterare: blixxten

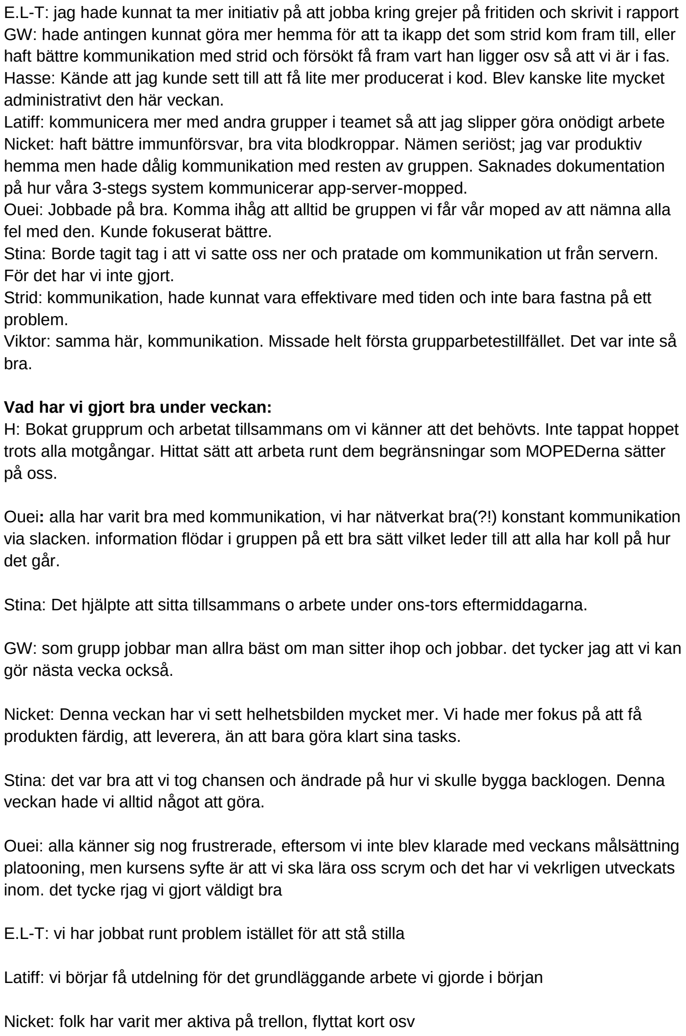
Punkter:

Vad har blivit klart under veckan + demo:
Appens nya styrning är klar. Den skickar styrsignaler till server.
Jobb har gjorts på image recognition, går inte att testa för att kameran på moped inte fungerar
Send image to server from moped klar, tolkningen där är kvar att göra.

Vad har inte blivit klart under veckan:
Stora målet var platooning, vilket vi inte hann med helt o hållet.

Vad gjorde jag bra under veckan:
blixxten: fixade med servern och anslutning, så nu skickas det styrsignlaler mellan app och server, och det spammas ut, och jag fixade med anslutningen, det var lurigt eller men ni nu funkar det, den styr bra.
E.L-T: har följt trellon ganska bra. i början av veckan hade vi problem. men det stoppade inte oss! vi är starka! vi skrev filer kring allting, så jag slipper stå still.
GW: inte så mycket faktiskt tycker jag . jag har varit frånvarande det har fuckat upp lite med mig och Erik, tidsmässigt. Läkaren var problemet. Det började bra, men sen gick det utför.
Hasse: Kom i tid och tog ovanligt bra anteckningar. Bra kommunikation med Nicket trots att han var sjuk delar av veckan.
Latiff: Jag utgick från Trellon ist för att göra vad jag tycker var bäst. Jag höll mig till Trellon.
Nicket: Mycket gjort oavsett problem. Lyckats jobba mig runt problem och producerat oavsett.
Ouei: Jag har suttit och jobbat med Latiif och E.L-T. Jag har suttit o jobbat o verkligen försökt tackla problem, det jag har gjort bra är att ist för att fokusera på att oj detta problemet måste jag fixa har jag gjort så istället så att jag kollar på vad jag kan göra istället för vad jag inte kan göra, ex fokusera på mjukvara som går att fixa. och det tycker jag är bra,, så det känns bra att vi är på väg åt rätt håll
Stina: alla tasks avklarade. suttit o samarbetat med alla berörda
Strid: image recognition har fått lite fart. kanske inte supereffektivt, men har producerat något iaf.
Viktor: har jobbat på ganska bra med image recognition, lagt ner mycket tid, lärt mig hur det fungerar. tittat på olika lösningar ifall det inte funkar med hans [Eriks] lösning.

Vad kunde jag gjort bättre under veckan?:
blixxten: Jag hade nog velat fixa signalerna innan, borde lagt upp jobbet bättre, prioritera rätt.



E.L-T: jag hade kunnat ta mer initiativ på att jobba kring grejer på fritiden och skrivit i rapport
GW: hade antingen kunnat göra mer hemma för att ta ikapp det som strid kom fram till, eller haft bättre kommunikation med strid och försökt få fram vart han ligger osv så att vi är i fas.
Hasse: Kände att jag kunde sett till att få lite mer producerat i kod. Blev kanske lite mycket administrativt den här veckan.
Latiff: kommunicera mer med andra grupper i teamet så att jag slipper göra onödigt arbete
Nicket: haft bättre immunförsvar, bra vita blodkroppar. Nämen seriöst; jag var produktiv hemma men hade dålig kommunikation med resten av gruppen. Saknades dokumentation på hur våra 3-stegs system kommunicerar app-server-mopped.
Ouei: Jobbade på bra. Komma ihåg att alltid be gruppen vi får vår moped av att nämna alla fel med den. Kunde fokuserat bättre.
Stina: Borde tagit tag i att vi satte oss ner och pratade om kommunikation ut från servern. För det har vi inte gjort.
Strid: kommunikation, hade kunnat vara effektivare med tiden och inte bara fastna på ett problem.
Viktor: samma här, kommunikation. Missade helt första grupparbetestillfället. Det var inte så bra.

Vad har vi gjort bra under veckan:
H: Bokat grupprum och arbetat tillsammans om vi känner att det behövts. Inte tappat hoppet trots alla motgångar. Hittat sätt att arbeta runt dem begränsningar som MOPEDerna sätter på oss.

Ouei**:** alla har varit bra med kommunikation, vi har nätverkat bra(?!) konstant kommunikation via slacken. information flödar i gruppen på ett bra sätt vilket leder till att alla har koll på hur det går.

Stina: Det hjälpte att sitta tillsammans o arbete under ons-tors eftermiddagarna.

GW: som grupp jobbar man allra bäst om man sitter ihop och jobbar. det tycker jag att vi kan gör nästa vecka också.

Nicket: Denna veckan har vi sett helhetsbilden mycket mer. Vi hade mer fokus på att få produkten färdig, att leverera, än att bara göra klart sina tasks.

Stina: det var bra att vi tog chansen och ändrade på hur vi skulle bygga backlogen. Denna veckan hade vi alltid något att göra.

Ouei: alla känner sig nog frustrerade, eftersom vi inte blev klarade med veckans målsättning platooning, men kursens syfte är att vi ska lära oss scrym och det har vi vekrligen utveckats inom. det tycke rjag vi gjort väldigt bra

E.L-T: vi har jobbat runt problem istället för att stå stilla

Latiff: vi börjar få utdelning för det grundläggande arbete vi gjorde i början

Nicket: folk har varit mer aktiva på trellon, flyttat kort osv

Stina: superbra, då kan man se vad folk faktiskt jobbar med

Vilka problem har vi haft och hur löstes dessa:
H: Icke fungerade MOPED - Svårt att lösa, bara att arbeta runt.

Problem med moped: handledarna sa direkt att en annan grupp har samma fel som oss. det gick inte att fixa felet. då fick vi en ny laddare, och det gick inte heller. Igår kom de tut en ny mjukvara som gick att installera på mopeden och då fungerade det bättre.
det var fel på SCUs mjukvara
Det var fel på kameran. något vi inte vet varför, men vi testade med någon annans kod där kameran fungerade, men det funkade ändå inte på vår moped

största problemet var att ingen kunde fixa det. de kan inte fixa mjukvaruproblemen.
gick alltså till detta tillfälle utan att få någon egentligen hjälp. eller jo (säger E.L-T) vi fick lite hjälp. Vi fick reda på att pyhonen fungerar och hjälp va en grupp att kolla vår kamera och fick bekräftat att den inte fungerade. Det samma gällde för frontsensorn.

vi har inte haft mopeden 100% fungerande. det har hindrat oss.
vi har ingen bra tracker för att se vad som testats och vad som inte testats, även hur saker har testats. testningen överlag är lite oklar

testa i python är svårt. men det Nicket menar är att så fort man testar bör man dokumentera någonstans så att andra kan se vad som fungerar som det skall utan tvekan.

om man inte är i full fas med gruppen och vill arbeta med något och det saknas dokumentation på bl.a. testning så är det omöjligt att veta att något faktiskt fungerar. Lätt lösning på detta är kommunikation. Samtidigt hade det varit najs om det stod utförligare i Trellon om testning osv

Vad skulle vi kunna gjort bättre?
Vara ännu bättre på att uppdatera småstatusar i Trellon, göra den mer levande och en större källa av information. Påminn varandra om att uppdatera källor för information, och påminna varandra om saker generellt.

Hur ser det ut med testning på appen? (blixxten vred på huvudet :(). Servern är jUnit testad)

dokumentation är problem, eller snarare saknaden av den

Ändringar till nästa sprint:
Skapa en mapp och struktur för dokumentation av kod
Hålla trellon mer levande. föra in mer småsaker där, detaljer om arbetet för att se hur det går för andra grupper med lätthet
Ny och redigerad kolumn i Trello. Syfte att förenkla sättet vi visar att något är testat.
Ready to verify (needs testing) innebär att något behöver testas. alla saker som läggs in under done SKA VARA TESTADE

jobba ännu mer utifrån att mopeden inte funkar, och inte skylla på att mopeden inte funkar.
istället jobba runt och lägga mer fokus på att bevisa att vår kod funkar trots att vi inte har

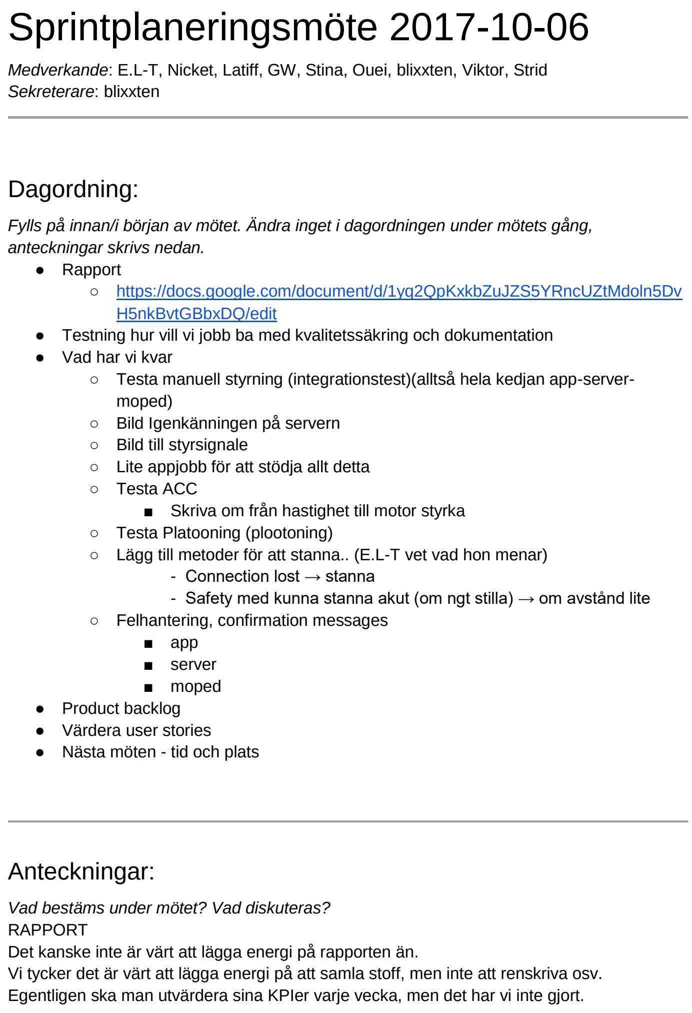
någon fungerande moped. vi måste ändå visa upp att vår kod funkar i teorin, den är testad: såhär.

Ha tillfällen där vi ska träffas och prata igenom vad som skall skrivas i rapporten. Ofta blir det att man arbetar ensamt utan att veta om det man tillför dokumentet är relevant.

Övrigt:

Bilagor:

C.10 Sprint 5 Planning



Sprintplaneringsmöte 2017-10-06

Medverkande: E.L-T, Nicket, Latiff, GW, Stina, Ouei, blixxten, Viktor, Strid
Sekreterare: blixxten

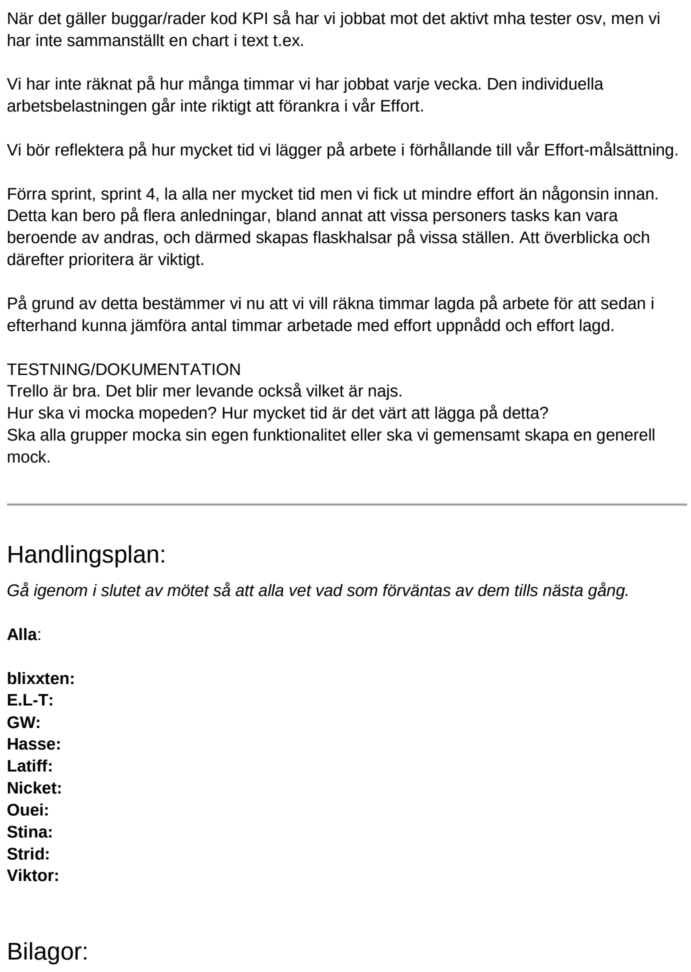
Dagordning:

Fylls på innan/i början av mötet. Ändra inget i dagordningen under mötets gång, anteckningar skrivs nedan.

- Rapport
  - https://docs.google.com/document/d/1yq2QpKxkbZuJZS5YRncUZtMdoln5DvH5nkBvtGBbxDQ/edit
- Testning hur vill vi jobb ba med kvalitetssäkring och dokumentation
- Vad har vi kvar
  - Testa manuell styrning (integrationstest)(alltså hela kedjan app-server-moped)
  - Bild Igenkänningen på servern
  - Bild till styrsignale
  - Lite appjobb för att stödja allt detta
  - Testa ACC
    - Skriva om från hastighet till motor styrka
  - Testa Platooning (plootoning)
  - Lägg till metoder för att stanna.. (E.L-T vet vad hon menar)
    - Connection lost → stanna
    - Safety med kunna stanna akut (om ngt stilla) → om avstånd lite
  - Felhantering, confirmation messages
    - app
    - server
    - moped
- Product backlog
- Värdera user stories
- Nästa möten - tid och plats

Anteckningar:

Vad bestäms under mötet? Vad diskuteras?
RAPPORT
Det kanske inte är värt att lägga energi på rapporten än.
Vi tycker det är värt att lägga energi på att samla stoff, men inte att renskriva osv.
Egentligen ska man utvärdera sina KPIer varje vecka, men det har vi inte gjort.



När det gäller buggar/rader kod KPI så har vi jobbat mot det aktivt mha tester osv, men vi har inte sammanställt en chart i text t.ex.

Vi har inte räknat på hur många timmar vi har jobbat varje vecka. Den individuella arbetsbelastningen går inte riktigt att förankra i vår Effort.

Vi bör reflektera på hur mycket tid vi lägger på arbete i förhållande till vår Effort-målsättning.

Förra sprint, sprint 4, la alla ner mycket tid men vi fick ut mindre effort än någonsin innan. Detta kan bero på flera anledningar, bland annat att vissa personers tasks kan vara beroende av andras, och därmed skapas flaskhalsar på vissa ställen. Att överblicka och därefter prioritera är viktigt.

På grund av detta bestämmer vi nu att vi vill räkna timmar lagda på arbete för att sedan i efterhand kunna jämföra antal timmar arbetade med effort uppnådd och effort lagd.

TESTNING/DOKUMENTATION
Trello är bra. Det blir mer levande också vilket är najs.
Hur ska vi mocka mopeden? Hur mycket tid är det värt att lägga på detta?
Ska alla grupper mocka sin egen funktionalitet eller ska vi gemensamt skapa en generell mock.

Handlingsplan:

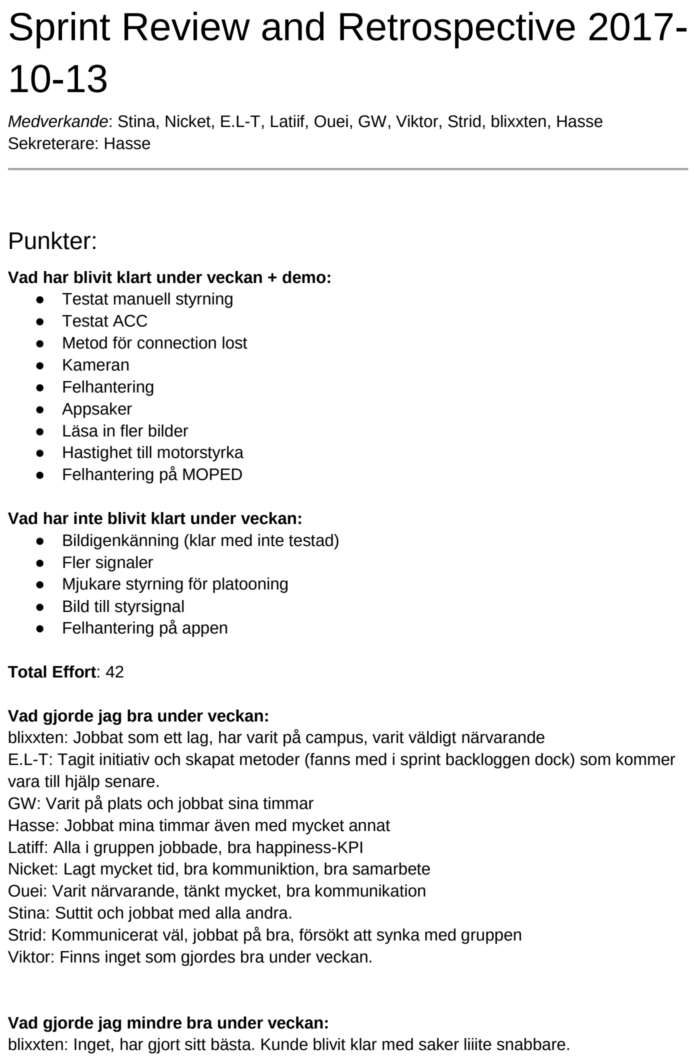
Gå igenom i slutet av mötet så att alla vet vad som förväntas av dem tills nästa gång.

Alla:

blixxten:
E.L-T:
GW:
Hasse:
Latiff:
Nicket:
Ouei:
Stina:
Strid:
Viktor:

Bilagor:

C.11 Sprint 5 Review and Retrospective



Sprint Review and Retrospective 2017-10-13

Medverkande: Stina, Nicket, E.L-T, Latiif, Ouei, GW, Viktor, Strid, blixxten, Hasse
Sekreterare: Hasse

Punkter:

Vad har blivit klart under veckan + demo:

- Testat manuell styrning
- Testat ACC
- Metod för connection lost
- Kameran
- Felhantering
- Appsaker
- Läsa in fler bilder
- Hastighet till motorstyrka
- Felhantering på MOPED

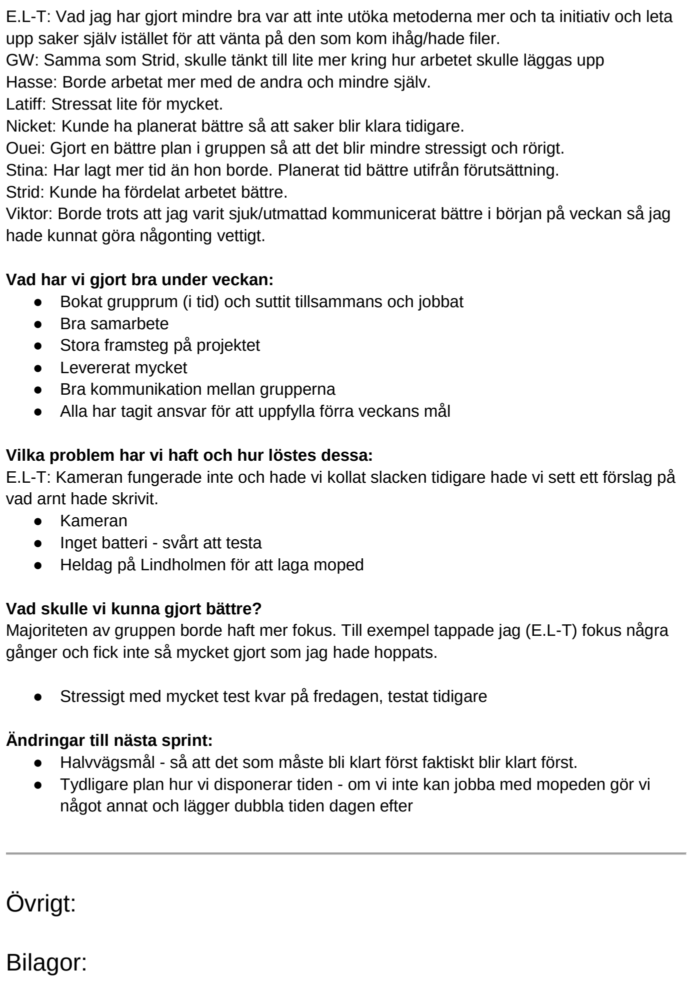
Vad har inte blivit klart under veckan:

- Bildigenkänning (klar med inte testad)
- Fler signaler
- Mjukare styrning för platooning
- Bild till styrsignal
- Felhantering på appen

Total Effort: 42

Vad gjorde jag bra under veckan:
blixxten: Jobbat som ett lag, har varit på campus, varit väldigt närvarande
E.L-T: Tagit initiativ och skapat metoder (fanns med i sprint backloggen dock) som kommer vara till hjälp senare.
GW: Varit på plats och jobbat sina timmar
Hasse: Jobbat mina timmar även med mycket annat
Latiff: Alla i gruppen jobbade, bra happiness-KPI
Nicket: Lagt mycket tid, bra kommuniktion, bra samarbete
Ouei: Varit närvarande, tänkt mycket, bra kommunikation
Stina: Suttit och jobbat med alla andra.
Strid: Kommunicerat väl, jobbat på bra, försökt att synka med gruppen
Viktor: Finns inget som gjordes bra under veckan.

Vad gjorde jag mindre bra under veckan:
blixxten: Inget, har gjort sitt bästa. Kunde blivit klar med saker liiite snabbare.



E.L-T: Vad jag har gjort mindre bra var att inte utöka metoderna mer och ta initiativ och leta upp saker själv istället för att vänta på den som kom ihåg/hade filer.
GW: Samma som Strid, skulle tänkt till lite mer kring hur arbetet skulle läggas upp
Hasse: Borde arbetat mer med de andra och mindre själv.
Latiff: Stressat lite för mycket.
Nicket: Kunde ha planerat bättre så att saker blir klara tidigare.
Ouei: Gjort en bättre plan i gruppen så att det blir mindre stressigt och rörigt.
Stina: Har lagt mer tid än hon borde. Planerat tid bättre utifrån förutsättning.
Strid: Kunde ha fördelat arbetet bättre.
Viktor: Borde trots att jag varit sjuk/utmattad kommunicerat bättre i början på veckan så jag hade kunnat göra någonting vettigt.

Vad har vi gjort bra under veckan:

- Bokat grupprum (i tid) och suttit tillsammans och jobbat
- Bra samarbete
- Stora framsteg på projektet
- Levererat mycket
- Bra kommunikation mellan grupperna
- Alla har tagit ansvar för att uppfylla förra veckans mål

Vilka problem har vi haft och hur löstes dessa:

E.L-T: Kameran fungerade inte och hade vi kollat slacken tidigare hade vi sett ett förslag på vad arnt hade skrivit.

- Kameran
- Inget batteri - svårt att testa
- Heldag på Lindholmen för att laga moped

Vad skulle vi kunna gjort bättre?

Majoriteten av gruppen borde haft mer fokus. Till exempel tappade jag (E.L-T) fokus några gånger och fick inte så mycket gjort som jag hade hoppats.

- Stressigt med mycket test kvar på fredagen, testat tidigare

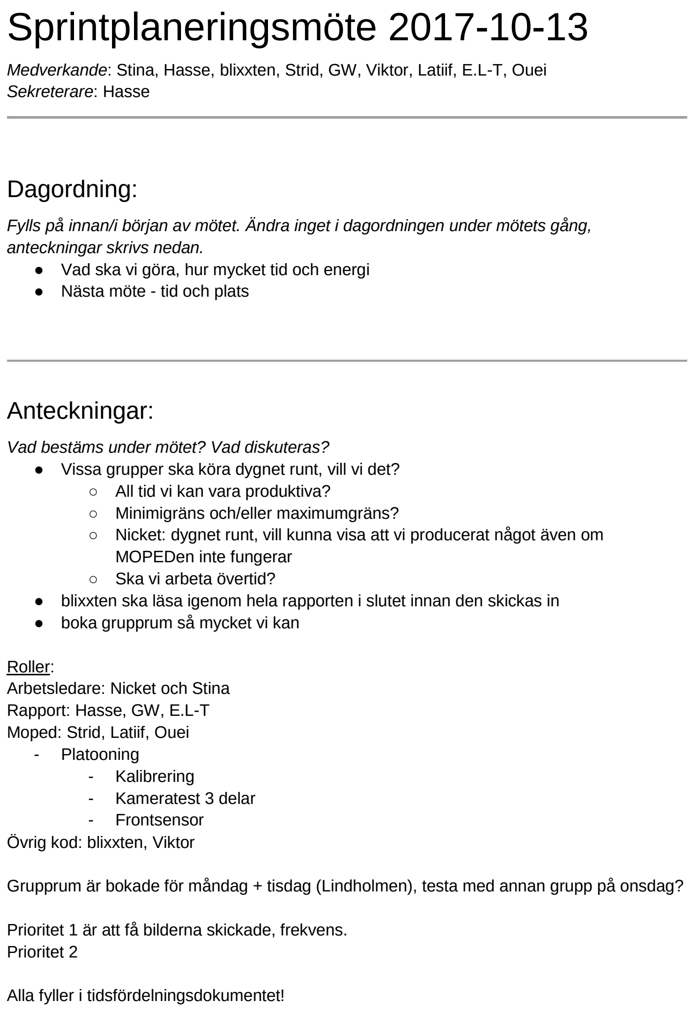
Ändringar till nästa sprint:

- Halvvägsmål - så att det som måste bli klart först faktiskt blir klart först.
- Tydligare plan hur vi disponerar tiden - om vi inte kan jobba med mopeden gör vi något annat och lägger dubbla tiden dagen efter

Övrigt:

Bilagor:

C.12 Sprint 6 Planning



Sprintplaneringsmöte 2017-10-13

Medverkande: Stina, Hasse, blixxten, Strid, GW, Viktor, Latiif, E.L-T, Ouei
Sekreterare: Hasse

Dagordning:

Fylls på innan/i början av mötet. Ändra inget i dagordningen under mötets gång, anteckningar skrivs nedan.

- Vad ska vi göra, hur mycket tid och energi
- Nästa möte - tid och plats

Anteckningar:

Vad bestäms under mötet? Vad diskuteras?

- Vissa grupper ska köra dygnet runt, vill vi det?
  - All tid vi kan vara produktiva?
  - Minimigräns och/eller maximumgräns?
  - Nicket: dygnet runt, vill kunna visa att vi producerat något även om MOPEDen inte fungerar
  - Ska vi arbeta övertid?
- blixxten ska läsa igenom hela rapporten i slutet innan den skickas in
- boka grupprum så mycket vi kan

Roller:
Arbetsledare: Nicket och Stina
Rapport: Hasse, GW, E.L-T
Moped: Strid, Latiif, Ouei

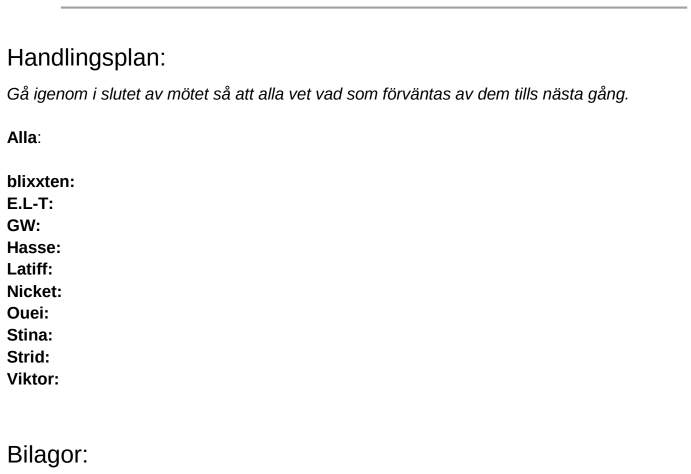
- Platooning
  - Kalibrering
  - Kameratest 3 delar
  - Frontsensor

Övrig kod: blixxten, Viktor

Grupprum är bokade för måndag + tisdag (Lindholmen), testa med annan grupp på onsdag?

Prioritet 1 är att få bilderna skickade, frekvens.
Prioritet 2

Alla fyller i tidsfördelningsdokumentet!



Handlingsplan:

Gå igenom i slutet av mötet så att alla vet vad som förväntas av dem tills nästa gång.

Alla:

blixxten:
E.L-T:
GW:
Hasse:
Latiff:
Nicket:
Ouei:
Stina:
Strid:
Viktor:

Bilagor:

C.13 Sprint 6 Review and Retrospective

Sprint Review and Retrospective 2017-10-20

Medverkande: GW, Strid, Stina, Nicket, Hasse, E.L-T, Latiif, Ouei, blixxten
Sekreterare: Hasse

Punkter:

Vad har blivit klart under veckan + demo:

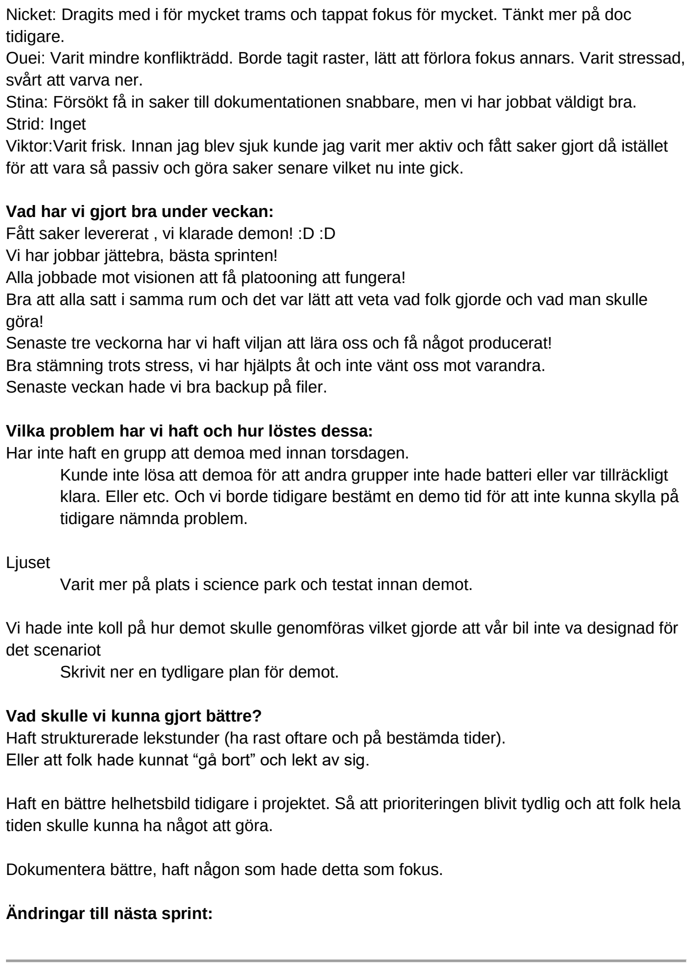
- Platooning och ACC
- Demo

Vad har inte blivit klart under veckan:

- Rapport
- Tester
- Dokumentation
- Upprensning
- Städa "bugs" i vår kod
- Stakeholder value
- Prototype

Vad gjorde jag bra under veckan:
blixxten: Varit med väldigt mycket, varit överallt och jobbat. Hjälpt till bra.
E.L-T: Tagit initiativ och försökt få folk att fylla i dokumenten och ändå försökt skriva så mycket jag kan i rapporten.
GW: Fått saker gjorda, skrivit rapport och haft riktigt bra kommunikation med de andra rapportskrivarna. Skrivit riktigt bra och funnits här för er.
Hasse: Jobbat tillsammans med alla andra och haft bra koll på processen.
Latiif: Har fördjupat mig i projektet i sin helhet, varit med med olika delgrupper och jobbade på olika tasks. Lagt så mycket tid jag kunde. Haft alla mina Daily scrums ifyllda.
Nicket: Fokus på att leverera och lagt tid som behövs för att få det klart. Bra kommunikation som har gjort att vi alla har kunnat samarbeta. Haft en bra helhetsbild.
Ouei: Lagt all tid som han kunnat. Jobbat konstant. Varit fokuserad längre tid än han brukar. Gjort listor som är underbara. Bra kommunikation och nått målen.
Stina: Jobbat på bra. Varit på plats och LEVERERAT! Gjort sin del i att få en färdig produkt.
Strid: Varit effektiv, väldigt effektiv. Lagt mycket tid. Fått mycket gjort och samarbetat väl.
Viktor:

Vad gjorde jag mindre bra (göra bättre) under veckan:
blixxten: inget
E.L-T: Kunde haft bättre fokus kontinuerligt under dagen.
GW: Ofokuserad mot slutet av dagarna, vid kväll typ, adhd:n kickade in.
Hasse: Prioriterat sömn mer, kunde presterat bättre med mer sömn.
Latiif: Kunde inte kontrollera min stressnivå.



Nicket: Dragits med i för mycket trams och tappat fokus för mycket. Tänkt mer på doc tidigare.
Ouei: Varit mindre konflikträdd. Borde tagit raster, lätt att förlora fokus annars. Varit stressad, svårt att varva ner.
Stina: Försökt få in saker till dokumentationen snabbare, men vi har jobbat väldigt bra.
Strid: Inget
Viktor:Varit frisk. Innan jag blev sjuk kunde jag varit mer aktiv och fått saker gjort då istället för att vara så passiv och göra saker senare vilket nu inte gick.

Vad har vi gjort bra under veckan:

Fått saker levererat , vi klarade demon! :D :D
Vi har jobbar jättebra, bästa sprinten!
Alla jobbade mot visionen att få platooning att fungera!
Bra att alla satt i samma rum och det var lätt att veta vad folk gjorde och vad man skulle göra!
Senaste tre veckorna har vi haft viljan att lära oss och få något producerat!
Bra stämning trots stress, vi har hjälpts åt och inte vänt oss mot varandra.
Senaste veckan hade vi bra backup på filer.

Vilka problem har vi haft och hur löstes dessa:

Har inte haft en grupp att demoa med innan torsdagen.

Kunde inte lösa att demoa för att andra grupper inte hade batteri eller var tillräckligt klara. Eller etc. Och vi borde tidigare bestämt en demo tid för att inte kunna skylla på tidigare nämnda problem.

Ljuset

Varit mer på plats i science park och testat innan demot.

Vi hade inte koll på hur demot skulle genomföras vilket gjorde att vår bil inte va designad för det scenariot

Skrivit ner en tydligare plan för demot.

Vad skulle vi kunna gjort bättre?

Haft strukturerade lekstunder (ha rast oftare och på bestämda tider).
Eller att folk hade kunnat “gå bort” och lekt av sig.

Haft en bättre helhetsbild tidigare i projektet. Så att prioriteringen blivit tydlig och att folk hela tiden skulle kunna ha något att göra.

Dokumentera bättre, haft någon som hade detta som fokus.

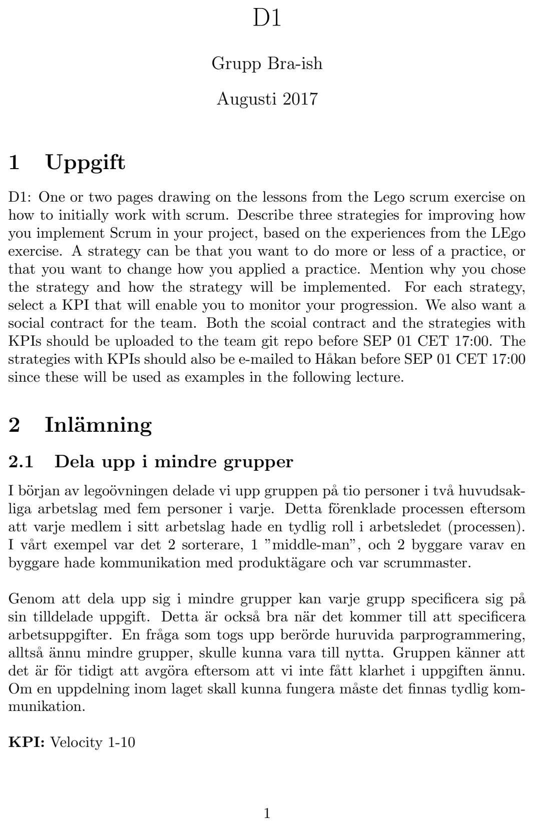
Ändringar till nästa sprint:

Övrigt:

Bilagor:

Appendix D Assignments

D.1 D1



D1

Grupp Bra-ish

Augusti 2017

1 Uppgift

D1: One or two pages drawing on the lessons from the Lego scrum exercise on how to initially work with scrum. Describe three strategies for improving how you implement Scrum in your project, based on the experiences from the LEgo exercise. A strategy can be that you want to do more or less of a practice, or that you want to change how you applied a practice. Mention why you chose the strategy and how the strategy will be implemented. For each strategy, select a KPI that will enable you to monitor your progression. We also want a social contract for the team. Both the scoial contract and the strategies with KPIs should be uploaded to the team git repo before SEP 01 CET 17:00. The strategies with KPIs should also be e-mailed to Håkan before SEP 01 CET 17:00 since these will be used as examples in the following lecture.

2 Inlämning

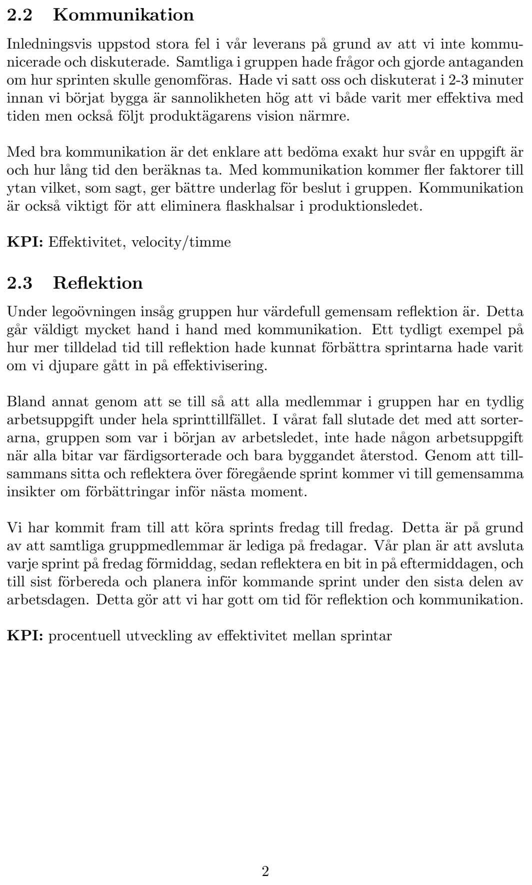
2.1 Dela upp i mindre grupper

I början av legoövningen delade vi upp gruppen på tio personer i två huvudsakliga arbetslag med fem personer i varje. Detta förenklade processen eftersom att varje medlem i sitt arbetslag hade en tydlig roll i arbetsledet (processen). I vårt exempel var det 2 sorterare, 1 ”middle-man”, och 2 byggare varav en byggare hade kommunikation med produktägare och var scrummaster.

Genom att dela upp sig i mindre grupper kan varje grupp specificera sig på sin tilldelade uppgift. Detta är också bra när det kommer till att specificera arbetsuppgifter. En fråga som togs upp berörde huruvida parprogrammering, alltså ännu mindre grupper, skulle kunna vara till nytta. Gruppen känner att det är för tidigt att avgöra eftersom att vi inte fått klarhet i uppgiften ännu. Om en uppdelning inom laget skall kunna fungera måste det finnas tydlig kommunikation.

KPI: Velocity 1-10

D.2 D1



2.2 Kommunikation

Inledningsvis uppstod stora fel i vår leverans på grund av att vi inte kommunicerade och diskuterade. Samtliga i gruppen hade frågor och gjorde antaganden om hur sprinten skulle genomföras. Hade vi satt oss och diskuterat i 2-3 minuter innan vi börjat bygga är sannolikheten hög att vi både varit mer effektiva med tiden men också följt produktägarens vision närmre.

Med bra kommunikation är det enklare att bedöma exakt hur svår en uppgift är och hur lång tid den beräknas ta. Med kommunikation kommer fler faktorer till ytan vilket, som sagt, ger bättre underlag för beslut i gruppen. Kommunikation är också viktigt för att eliminera flaskhalsar i produktionsledet.

KPI: Effektivitet, velocity/timme

2.3 Reflektion

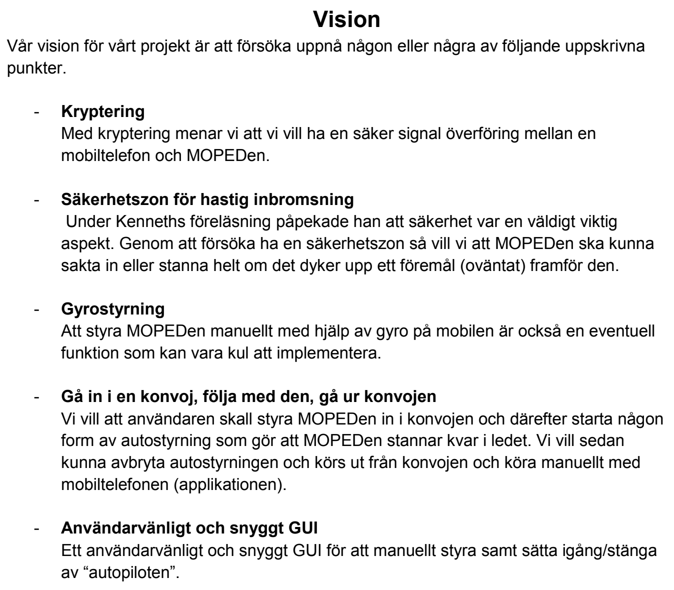
Under legoövningen insåg gruppen hur värdefull gemensam reflektion är. Detta går väldigt mycket hand i hand med kommunikation. Ett tydligt exempel på hur mer tilldelad tid till reflektion hade kunnat förbättra sprintarna hade varit om vi djupare gått in på effektivisering.

Bland annat genom att se till så att alla medlemmar i gruppen har en tydlig arbetsuppgift under hela sprinttillfället. I vårat fall slutade det med att sorterarna, gruppen som var i början av arbetsledet, inte hade någon arbetsuppgift när alla bitar var färdigsorterade och bara byggandet återstod. Genom att tillsammans sitta och reflektera över föregående sprint kommer vi till gemensamma insikter om förbättringar inför nästa moment.

Vi har kommit fram till att köra sprints fredag till fredag. Detta är på grund av att samtliga gruppmedlemmar är lediga på fredagar. Vår plan är att avsluta varje sprint på fredag förmiddag, sedan reflektera en bit in på eftermiddagen, och till sist förbereda och planera inför kommande sprint under den sista delen av arbetsdagen. Detta gör att vi har gott om tid för reflektion och kommunikation.

KPI: procentuell utveckling av effektivitet mellan sprintar

D.3 D2



Vision

Vår vision för vårt projekt är att försöka uppnå någon eller några av följande uppskrivna punkter.

- **Kryptering**
  Med kryptering menar vi att vi vill ha en säker signal överföring mellan en mobiltelefon och MOPEDen.

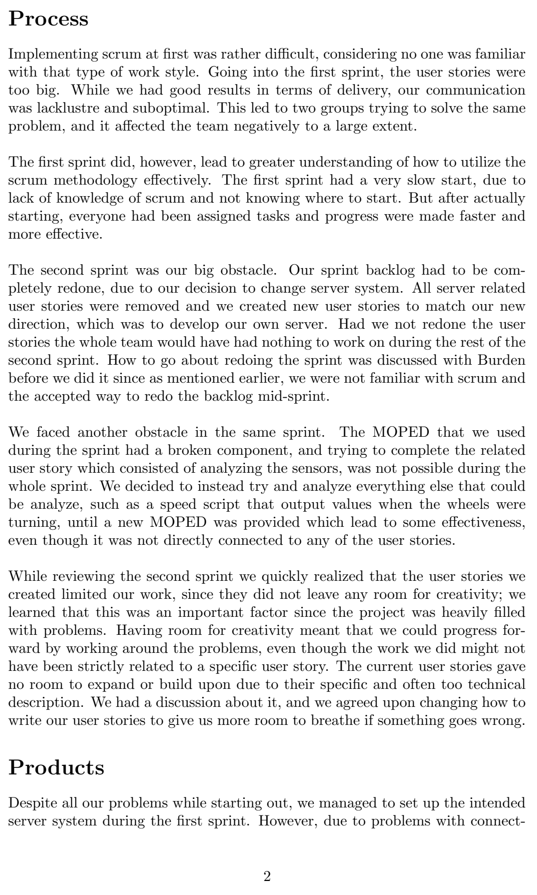
- **Säkerhetszon för hastig inbromsning**
  Under Kenneths föreläsning påpekade han att säkerhet var en väldigt viktig aspekt. Genom att försöka ha en säkerhetszon så vill vi att MOPEDen ska kunna sakta in eller stanna helt om det dyker upp ett föremål (oväntat) framför den.

- **Gyrostyrning**
  Att styra MOPEDen manuellt med hjälp av gyro på mobilen är också en eventuell funktion som kan vara kul att implementera.

- **Gå in i en konvoj, följa med den, gå ur konvojen**
  Vi vill att användaren skall styra MOPEDen in i konvojen och därefter starta någon form av autostyrning som gör att MOPEDen stannar kvar i ledet. Vi vill sedan kunna avbryta autostyrningen och körs ut från konvojen och köra manuellt med mobiltelefonen (applikationen).

- **Användarvänligt och snyggt GUI**
  Ett användarvänligt och snyggt GUI för att manuellt styra samt sätta igång/stänga av “autopiloten”.

D.4 D3



Process

Implementing scrum at first was rather difficult, considering no one was familiar with that type of work style. Going into the first sprint, the user stories were too big. While we had good results in terms of delivery, our communication was lacklustre and suboptimal. This led to two groups trying to solve the same problem, and it affected the team negatively to a large extent.

The first sprint did, however, lead to greater understanding of how to utilize the scrum methodology effectively. The first sprint had a very slow start, due to lack of knowledge of scrum and not knowing where to start. But after actually starting, everyone had been assigned tasks and progress were made faster and more effective.

The second sprint was our big obstacle. Our sprint backlog had to be completely redone, due to our decision to change server system. All server related user stories were removed and we created new user stories to match our new direction, which was to develop our own server. Had we not redone the user stories the whole team would have had nothing to work on during the rest of the second sprint. How to go about redoing the sprint was discussed with Burden before we did it since as mentioned earlier, we were not familiar with scrum and the accepted way to redo the backlog mid-sprint.

We faced another obstacle in the same sprint. The MOPED that we used during the sprint had a broken component, and trying to complete the related user story which consisted of analyzing the sensors, was not possible during the whole sprint. We decided to instead try and analyze everything else that could be analyze, such as a speed script that output values when the wheels were turning, until a new MOPED was provided which lead to some effectiveness, even though it was not directly connected to any of the user stories.

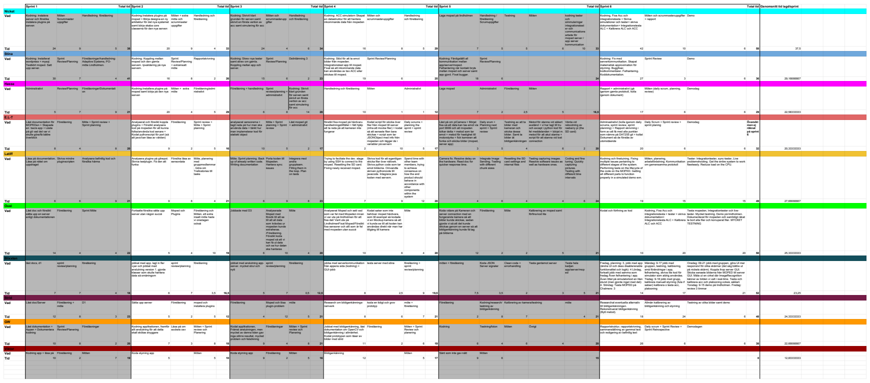
While reviewing the second sprint we quickly realized that the user stories we created limited our work, since they did not leave any room for creativity; we learned that this was an important factor since the project was heavily filled with problems. Having room for creativity meant that we could progress forward by working around the problems, even though the work we did might not have been strictly related to a specific user story. The current user stories gave no room to expand or build upon due to their specific and often too technical description. We had a discussion about it, and we agreed upon changing how to write our user stories to give us more room to breathe if something goes wrong.

Products

Despite all our problems while starting out, we managed to set up the intended server system during the first sprint. However, due to problems with connect-

Appendix E Time Distribution

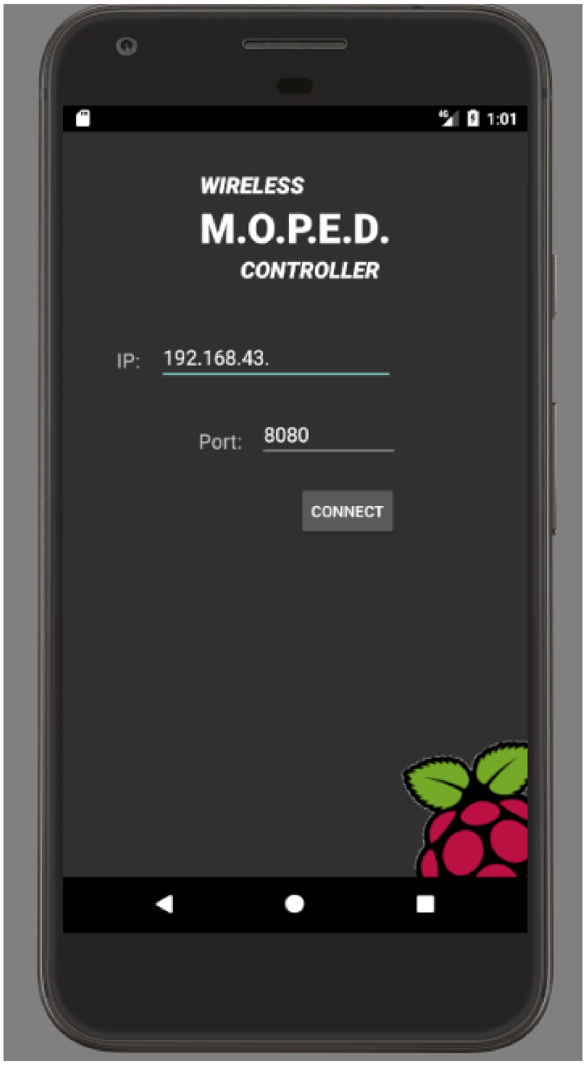
E.1 Time/member



	Sprint 1			Total tid	Sprint 2			Total tid	Sprint 3			Total tid	Sprint 4			Total tid	Sprint 5					Total tid	Sprint 6			Total tid	Genomsnitt tid lagd/sprint
Nicket																											
Vad	Kodning: instalera server och försöka instalera plugins på serven	Möten Scrummaster uppgifter	Handledning föreläsning		Kodning:Instalera plugins på moped + Börja designa en ny arkitektur för det nya systemet samt börja skaba core classerna för den nya serven	Möten + extra möte och scrummaster uppgifter	Handledning och föreläsning		Kodning: Skrivit klart grunden för serven samt skrivit en första vertion av acc samt simulering för acc	Möten och scrummasteruppgifter	Handledning och föreläsning		Kodning: ACC simulatorn Skapat en datastruktur för att hantera inkommande data från mopeden	Möten och scrummasteruppgifter	Handledning och föreläsning		Laga moped på lindholmen	Handledning / föreläsning Scrumuppgifter	Testning	Möten	kodning tester och simmuleringar integrationstester och communicationsarbete för moped server / app server kommunication		Kodning, Fixa Acc och integrationstesta + Skriva simulationer och tester+ skriva dokumentation+ Integrationstesta ALC + Kalibrera ALC och ACC	Möten och scrummasteruppgifter + rapport	Demo		
Tid	24	9	5	38	20	9	4	33	24	4	6	34	16	8	5	29	7	5	5	6	10	33	42	10	6	58	37,5
Stina																											
Vad	Kodning: Installerat wordpress + mysql. Testkört moped. Satt upp server.	Sprint Review/Planning	Föreläsningar/handledning: Adaptive Systems, PO-möte Lindholmen.		Kodning: Koppling mellan moped och den gamla servern. Ipvalidering på nya servern.	Sprint Review/Planning + extrainsatt möte	Rapportskrivning		Kodning: Skrev nya tester samt skrev om gamla. Koppling mellan app och server.	Sprint Review/Planning	Delinlämning 3		Kodning: Stöd för att ta emot bilder från mopeden. Integrationstest app till moped. Fixat så att inkommande data kan användas av tex ACC eller skickas till moped.	Sprint Review/Planning			Kodning: Färdigställt all kommunikation mellan app/server/moped. Felhantering när kontakt bryts mellan moped och server samt app gjord. Fixat buggar.	Sprint Review/Planning					Kodning: Fix med serverkommunikation. Skapat matematisk approximation för styrning. Buggfixar, kodkommentarer. Felhantering. Koddokumentation.	Sprint Review	Demo		
Tid	30	7	4	41	16	8	2	26	15	6	2	23	19	6		25	18	6				24	27	3	6	36	29,16666667
Hasse																											
Vad	Administrativt	Review/Planning	Föreläsningar/Dokumentation		Kodning:Installera plugins på moped samt börja på den nya serven	Möten + extra möte	Föreläsning/administrativt		Föreläsning + handledning	Sprint review/planning + administrativt	Kodning: Skrivit klart grunden för serven samt skrivit en första vertion av acc samt simulering för acc		Handledning och föreläsning	Möten	Administrativt		Laga moped	Administrativt	Föreläsning	Möten			Rapport + administrativt (gå igenom gamla protokoll, kolla igenom dokument)	Möten (daily scrum, planning, review)	Demodag		
Tid	5	7	9	21	20	4	5	29	6	8	14	28	10	1	1	12	7	2	2,5	5		16,5	17	6	6	29	22,58333333
E.L-T																											
Vad	Läst documentation för MOPEDen + Skapade en bacis app + Läste på git vad det var vi skulle göra/få bättre överblick	Föreläsning	Möte + Sprint review + Sprint planning		Analyserat och försökt koppla plugins + Försökt analysera kod på mopeden för att kunna folka/använda kod senare + Kodat pythonscript för port (så att java kan läsa av värden)	Föreläsning	Sprint review + Möte + Sprint planning		analyserat sensorerna + tagit reda på hur man ska använda data + tänkt hur man implemeterar kod för statiskt object	Möte + Sprint planning + Sprint review	Läst moped git + administrativt		försökt fixa moped på hårdvaru handledningstillfället + fått hjälp att ta reda på att kameran inte fungerar	Kodat script för skicka över filer från moped till server (mha att mocka filer) + kodat så att senaste filen bara skickas + script som tar JSONObject med info från mopeden och lägger de i variabler på servern	Daily scrums + planning the sprint + sprint review		Läst på om piCamera + Börjat fixa så att data kan tas emot via port 9999 och att mopeden tolkar detta + metod som tar emot + metod för hastighet till motorstyrka + fick kameran att funka och skicka bilder (moped, server app)	Daily srum + Planning next sprint + Sprint review	Testning av att ta bilder med kameran och skicka dessa bilder. Samt ta bilder åt bildigenkänningen	Metod för stanna vid säkert avstånd + vi har lagt till tru och except i python kod för fel meddelanden + börjat m. metod för att akut stanna + script för att stanna vid lost connection	Vänta vid rebootning av rasberry pi (the SD card)		Adminastrativt (kolla igenom daily scrums, sprint review, sprint planning) + Rapport skrivining i form av att få med alla punkter som nämns på DAT255 git + kollat Dokument så de förstås av utomstående	Daily Scrum + Sprint review + sprint planing	Demodag	**Översikt men ej räknat på sprint 6**	
Tid	8	4	8	20	9	2	5	16	10	5	2	17	2	10	5	17	3	5	4	6	2	20	20	6	6	32	20,33333333
Latiff																											
Vad	Läsa på documentation, Läsa på nätet om uppdraget	Skriva mindre pluginscripten	Analysera befintlig kod och försöka härma		Analysera plugins på gitrepot. Skriva testplugin. Få den att funka	Försöka läsa av sensordata	Möte, planering med gruppmedlemmar . Göra om Trellostories till tasks		Möte. Sprint planning. Back up of already written code. Writing documentation	Porta koden till Mopeden. Hantera sync issues	Integrera med andra gruppteam. Filling them in the loop. Plan on tests		Trying to faciliate the dev. stage by using SSH to connect to the moped. Resetting the SD card. Fixing newly received moped.	Skriva kod för att egentligen skicka filer över nätverk. Skriva python code som tar emot bilderna. Omvandla skriven pythoncode till javacode. Integrera java koden med servern.	Spend time with other team members, trying to achieve consensus on how the end product should behave in accordance with other components within the system		Camera fix. Resolve delay on the hardware. Read doc for quicker response time.	Integrate Image Sending. Testing with different chunk sizes	Resetting the SD card settings and internal files	Testing capturing images. Resolve software issues as well as hardware ones.	Coding and fine tuning. Quickly capturing pictures. Testing with different time intervals		Kodning och finetuining. Fixing multiple issues pertaining to different stages of the system. Performing tests on the lifecycle of the code on the MOPED. Getting all different parts to function properly in a simulated demo evn.	Möten, planering, arbetsfördelning. Kommunikation om gemensamma protokoll	Tester: Integrationtester, sync-tester, Live problemshooting. Get the entire system to work flawlessly. Reduce load on the CPU.		
Tid	6	6	9	21	9	8	4	21	6	9	8	23	7	9	12	28	4	6	4	4	6	24	19	15	15	49	27,66666667
Ouei																											
Vad	Läst doc och försökt sätta upp en server enligt dokumentationen	Föreläsning	Sprint Möte		Fortsatte försöka sätta upp server utan någon succé	Moped och Plugins	Föreläsning och Möten, ett extra insatt möte hade vi onsdagen också		Jobbade med D3	Analyserade Moped med försök till att se till att all data som krävdes ur mopeden kunde extraheras. /Föreläsning. Försökt koda moped så att vi kan få ut data och se hur datan ska hanteras	Möte		Analyserat Moped och sett vad som var fel med Mopeden innan vi var ute på lindholmen för att fixa det/ Varit ute på LindholmenFixat Moped/Försökt fixa sensorer och allt som är fel med mopeden utan succé	Kodat saker som inte behöver moped hårdvara, som till exempel så kodade vi en Mockup kamera så att vi kunde se till att koden kan användas direkt när man har tillgång till kamera	Möte		Koda vidare på Kameran och server connection med en fungerande kamera så att bilder kunde skickas, sedan gjorde vi så att det kunde skickas genom en server så att bildigenkänning kunde få tag på bilderna	Föreläsning	Möte	Kalibrering av moped samt förfina kod lite			Kodat och förfining av kod	Kodning, Fixa Acc och integrationstesta + tester + skriva dokumentation+ Integrationstesta ALC + Kalibrera ALC och ACC	Testa mopeden, Integrationtester och live-tester. Mycket testning. Demo på lindholmen. Dokumenterat för mopeden och samtidigt råkat ta bort alla filer och korruperat filer. MYCKET TESTNING		
Tid	14	4	4	22	5	4	10	19	6	10	4	20	10	6	4	20	12	2	2	5		21	10	20	20	50	25,33333333
Blixxten																											
Vad	läst docs, d1	sprint review/planning	föreläsning		jobbat med app. lagt in fler vyer och jobbat med anslutning version 1. gjorde klasser som skulle hantera data så småningom	sprint review/planning	föreläsning		jobbat med anslutning app-server. mycket strul och nytt	sprint review/planning	föreläsning		jobba med serverkommunikation från appens sida (kodning) + GUI-jobb	testa server med stina	föreläsning + sprint review/planning		möten + föreläsning	Koda JSON Server signaler	Clean code + errorhandling	Testa gentemot server	Testa hela kedjan app/server/moped		Fredag, planning: 3, jobb med app (ändra UI och dess disable/enable funktionalitet och logik): 4 Lördag, fortsatt jobb med samma som fredag Även felhantering i app. Även tittat på simulatorkod en liten stund (men gjorde inget med det): 4. Söndag: Testa MOPED på Chalmers: 3	Måndag: 9-17 jobb med gruppen, testning, kalibrering, små förändringar i app, felhantering, skriva lite kod för simulator som aldrig användes. Tisdag: 9-18 jobb med grupp, kalibrera manuell styrning (fula if satser) kalibrera o testa acc, platooning	Onsdag: 09-21 jobb med gruppen, göra UI mer responsivt för olika skärmar (det såg bättre ut på nickets skärm). Koppla ihop server GUI. Skicka senaste bilderna från MOPED till server GUI. Måla ut en cirkel där ImageRecognition känner av bilden vi valt i real-time. Testa och kalibrera acc och platooning också, såklart. Torsdag: 9-15 demo på lindholmen. Fredag: review 3 timmar		
Tid	7	7	4	18	6	7	3,5	16,5	5	7	0,5	12,5	8	2,5	9	19,5	7,5	3,5	2	3	5	21	14	17	21	52	23,25
Strid																											
Vad	Läst doc/Server	Föreläsning + möte	D1		Sätta upp server	Föreläsning	moped och installera plugins		Föreläsning	Moped och lösa plugin-problem	möte		Research om bildigenkännings-ramverk	koda en tidigt och grov prototyp	möte + föreläsning		Föreläsning	Kodning/research/ testning av bildigenkänning	Kalibrering av kamera/testning		möte		Researchat eventuella alternativ till bildigenkänningen. Rekonstruerat bildigenkänning (Bytt metod).	Allmän kalibrering av bildigenkänning och styrning	Testning av olika bilder samt demo		
Tid	12	8	3	23	5	2	5	12	4	12	5	21	8	6	7	21	2	8	7		4	21	18	24	6	48	24,33333333
GW																											
Vad	Läst dokumentation + Appen + Dokumentera kodning	Sprint Review/Planering	Föreläsningar		Kodning applikationen, framför allt anslutning för att detta skall skötas snyggare	Läsa på om sockets osv	Möten + Sprint review och Planering		Kodat applikationen, Främst anslutningen, men mycket av denna tiden gav inga störe resultat, mycket problem och felsökning	Föreläsningar	Möten + Sprint review och Planering		Jobbat med bildigenkänning, läst dokumentation om OpenCV och bildigenkänning i allmänhet. Kodat prototypen som läser av bilder med strid	Föreläsning	Möten + Sprint Review och planering		Kodning	Testning/foton	Möten	Övrigt			Rapportstruktur, rapportskrivning, sammanställning av gammal text och redigering av befintlig text	Daily scrum + Sprint Review + Sprint Retrospective	Demodagen		
Tid	10	8	2	20	7	3	8	18	9	4	8	21	11	2	6	19	6	4	6	4		20	26	6	6	38	22,66666667
Viktor																											
Vad	Kodning app + läsa på	Föreläsning	Möten		Koda styming app		Möten		Koda styming app	Föreläsning	Möten		Bildigenkänning		Möten		Sånt som inte gav nått	Möten									
Tid	10	2	7	19	5		5	10	9	2	5	16	12		5	17	9	6				15				0	12,83333333

Appendix F Android Application

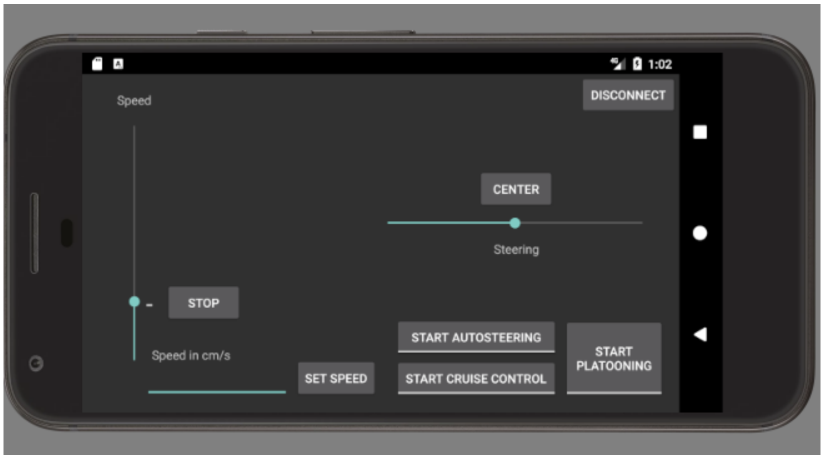
F.1 Connect view



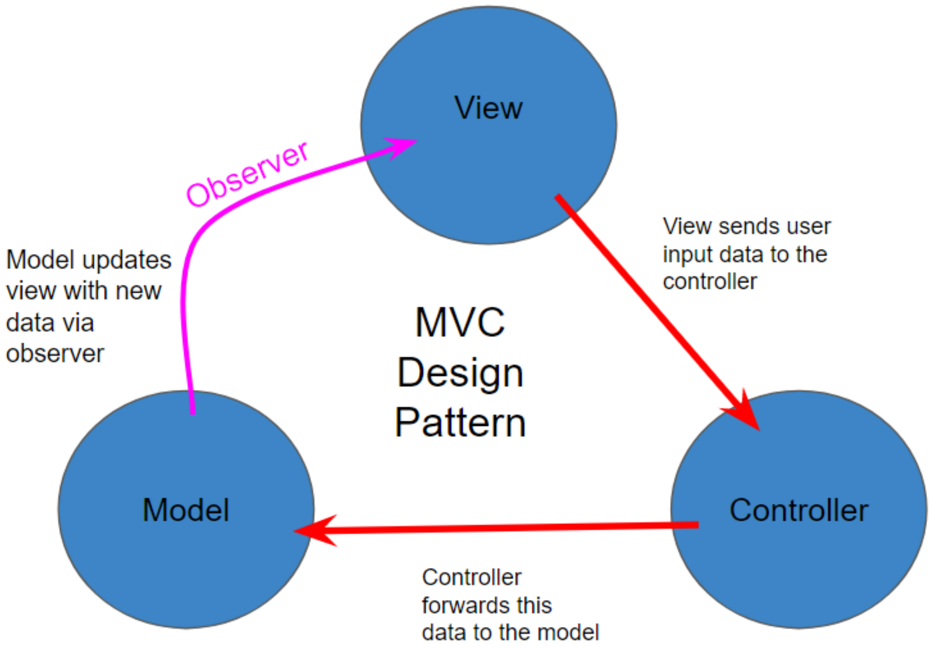
IP: 192.168.43.

Port: 8080

F.2 Steering view



F.3 MVC Implementation


View
Observer
Model updates view with new data via observer
View sends user input data to the controller
MVC Design Pattern
Model
Controller
Controller forwards this data to the model

F References

[1] Transportanalys. (2017, April) Vägtrafikskador 2016. [Online]. Available: https://www.trafa.se/globalassets/statistik/vagtrafik/vagtrafikskador/2016/vagtrafikskador-2016.pdf?

[2] Volvo. (2017, October) Trucks talking to each-other in multi-brand platooning project. [Online]. Available: http://www.volvogroup.com/en-en/news/2017/oct/trucks-talking-to-each-other-in-multi-brand-platooning-project.html

[3] E. A. M. A. (ACEA). (2016, March) What are the benefits of truck platooning? [Online]. Available: http://www.acea.be/news/article/what-are-the-benefits-of-truck-platooning

[4] P. Green. (2017, September) Splitting stories into small, vertical slices. [Online]. Available: http://blogs.adobe.com/agile/2013/09/27/splitting-stories-into-small-vertical-slices/

[5] S. Cooley. (2017) Windows Subsystem for Linux Documentation. [Online]. Available: https://msdn.microsoft.com/en-us/commandline/wsl/about

[6] M. Carlsson, “Shape the future of driving,” University Lecture, 2017.

[7] C. Frithiof, “Startup life,” University Lecture, 2017.