Final Team Reflection / Final Report

Teams 16

Thomas Jinton	jintont
Jennifer Krogh	kroghj
Ludvig Lindell	ludlinde
Jesper Lundgren	jeslundg
Johan Nilsson	nilssojo
Emma Pettersson	emmp
Antonia Welzel	welzel

Agile Software Project Management [DAT257]

Information Technology
Chalmers Institute of Technology
Gothenburg, Sweden
2020-06-01
Version 1.0

Contents

1 Customer Value and Scope

1.1 Scope

A: What was

We decided on our project idea in the beginning of the course, where we identified a need for a product that could work as a companion to make the pub crawl easier for all the attendants. The pub crawl takes place four times a year on the Chalmers campus. We wanted to therefore create an application that could be of value to students that go on the pub crawl. We contacted a student who is a regular at the pub crawls and was very interested in a product that would help her avoid the typical problems that student face at a pub crawl, like standing in long lines when there are pubs with shorter queues.

After the first discussion with our stakeholder, we started listing important features which could create value for students that go on the pub crawl. We then went on to make a project scope, that contained a business model canvas and a mock-up in order to create an overview of our project. The business model canvas lists the relevant actors and activities regarding our project, as well as the value proposition which lists the most value-creating features of the application that were identified with the stakeholder.

The team then started writing user stories to get an overview of the project from a more technical perspective and to create our product backlog.

A document detailing our Project Scope can be found on GitHub. Click *here* to access it. If you cannot click the link, simply visit our GitHub and head into the "Deliverables" folder. We have also handed it in on Canvas.

Our **Priority of Features**, based on discussions with our stakeholder, looked like this:

- ☑ List of pubs available, with information about them
- ☑ Queue time for the pubs, with the ability to vote
- ☑ A map of the Chalmers area, with the pubs marked on it
- ☐ A profile for the users, so they can see each other on the map

B: What might, or should, be

In a future project, we would have tried to focus on having more regular meetings with our stakeholder. This would have increased our understanding of their needs and facilitated other tasks related to the project, like grooming the product backlog.

We would also consider assigning the role of a product owner to a specific person or a smaller number of people, to represent the stakeholder's interest and to stay on top of the product backlog. We were all product owners in this project, where we all had an equal responsibility regarding the project's scope and how it creates value. While this ensured that every team member had this always in mind when making decisions in the sprint planning, it also caused the team to, at times, neglect this area because of for example technical issues in the project that

required more focus.

Furthermore, there were sometimes too many opinions about features in the application and the product backlog and its priority order, which caused discussions to go on very long without really gaining anything from this.

C (A → B): Feedback designed to reduce the gap

To assure continuous focus on the project scope and its value creation, we would focus on setting more regular meetings with an external stakeholder and also by assigning the role of the product owner to one to three people at most, depending on the size of the team. This would have made our sprint planning meetings more efficient since there was confusion and discussion, especially related to the users stories and their priorities.

As will be discussed later, a dedicated product owner would allow the above mentioned to be done in parallel with other responsibilities in the team.

1.1.1 Initiative and epics

A: What was

Our initiative was *As a user, I want a web page for the pub rounds.* In addition to it, we had two epics: *I want information about the pubs* and *I want a map with the pubs.*

We didn't "use" our epics, instead they were mostly there because we needed to have them. They were implemented after we begun with our user stories and after we had discussed our idea. Because of that we didn't feel like they helped us all that much, and the user stories were sufficient for us. In a bigger project, we can definitively see more of a benefit, where several teams have to have a clear vision.

B: What might, or should, be

If we look at our list of priority of features (can be found under section *1.1 Scope*), two of them have epics. We should have created one epic for each feature, especially since the queue times were very important and definitely deserving of one.

C (A → B): Feedback designed to reduce the gap

The next time, we should think more about the correlation between the features we want to implement and how they should be structured by forming more defined epics that covers the whole project. This also relates back to the previous section about a dedicated product owner. Partly because it would help create structure for the project but also because this together with a product owner would shorten the long discussions and lessen the confusion regarding the requested features and their priorities.

1.2 Success Criteria

A: What was

For this project, we obviously wanted to create a valuable product for our stakeholder. We achieved our **Minimum Viable Product** (MVP) for the project, which was a website with the

following functions:

- ☑ List of pubs available
- ☑ Queue time for the pubs

In our team reflection for week 5, we also wrote that *"... Our main goal in this course is to learn about working as a team by using available agile practices. To get there, we need to know how to use the different Scrum techniques and be able to improve our work with these methods. Another important aspect is to learn how and what to communicate to become a more efficient group..."*.

We feel that our communication has gotten a lot better, both due to our frequent meetings and chatting in between them, and our pair programming sessions. We have also learned a lot more about Scrum and sprints, including sprint planning, stand-ups and retrospectives.

B: What might, or should, be

If we were to do this project again, we would have put more time towards really understanding Scrum and the different agile practices from the start of the course. We all had more basic knowledge on these topics from lectures and some light reading, and figured we would be learning-by-doing. However this took a lot of time and led to a slow start of the project.

C (A $\rightarrow$ B): Feedback designed to reduce the gap

We should have taken the time to learn more about Scrum by reading more and discuss together as a group about what Scrum and its roles mean within and for the team.

1.3 User Stories

All our user stories followed the standard pattern: "As a X, I want Y, so that Z". The team made sure that all user stories were in the proper format and that they all had acceptance criteria. We also made sure that the user stories followed the INVEST-criteria. We focused a lot on the value aspect of a user story, but had issues with the independent aspect and often created user stories that had to be broken down later when we had gained a better understanding and realized that they were too big.

1.3.1 Task Breakdown

A: What was

Every user story is broken down into several smaller tasks with **Independence** and similar aspects in mind. The name of the tasks all follow the same pattern: first, the number for the user story. Then, the number of the task (starting from 1). Last, a description of the task.

New user stories were continuously created and improved, therefore task breakdowns took place during the sprint planning meeting. This way, we were able to do task breakdowns on user stories that were up-to-date with the course of the project. User stories were chosen from the top of the product backlog, since these are the ones with the highest priority.

We had issues with our task breakdown most sprints, and it was something that we also committed to improving almost every week. We found that it was hard to make tasks independent and that often caused problems in the code or delays during a sprint.

B: What might, or should, be

We became better and better at breaking down tasks, and believe that apart from getting better by having more experience, having a more complete website made the task breakdowns easier to do. In the future, we would have liked our task breakdowns to be more independent from the beginning.

C (A $\rightarrow$ B): Feedback designed to reduce the gap

One improvement could have been to spend more time on breaking down tasks while planning our sprints. On the other hand, we already felt that our meetings were taking too much time, and spending even more time on task breakdowns is not something that we would have liked. Maybe the way we did is was decent and that we should accept that there were some problems in the beginning, which would get better by experience.

1.3.2 Acceptance Criteria

A: What was

Each user story described a problem and a wish together with a set of acceptance criteria that must be fulfilled before the user story could be marked as ”done”. They were written as checklists on the Trello card.
We also added acceptance criteria to some of the tasks we had for us as a programmer, since we realized that there were some issues with how much the task assumes and when it could be considered finished.

B: What might, or should, be

We were generally satisfied with our acceptance criteria and felt that they reflected the goal with the user stories. However, there were some user stories where we could have added more acceptance criteria to make the goal with the user story more clear.

Also we should have been more consistent with acceptance criteria for the more complex tasks, to make them easier to understand and work with. We added acceptance criteria to tasks spontaneously where there was confusion or when there were ideas on task implementation that resulted from group discussions, but there were tasks that were equally as complex and did not have any acceptance criteria since the task seemed clear. Further, some acceptance criteria for tasks were relatively general because not every team member had worked with the same content and it was hard to set criteria from a programmer’s perspective.

C (A $\rightarrow$ B): Feedback designed to reduce the gap

We could have added more acceptance criteria to the individual user stories to make them more defined.
In the future, if we decide to again add acceptance criteria to the tasks, we should be more consistent with adding them and also only add usable acceptance criteria that help the programmer(s) assigned to the task.

1.3.3 Effort Estimation

A: What was

Effort estimation took also place during the sprint planning meetings and especially in the beginning this helped us realize if the user story did not follow the INVEST-criteria. We did our effort estimation using hours, so that we could relate it to the number of hours the whole group will spend this week on coding. We overestimated most tasks in the beginning of the project, and then underestimated tasks that seemed relatively small but turned out to involve more work. This was most likely due to how the task was worded and/or not enough knowledge in the technical aspects. We were able to slowly solve this by adding acceptance criteria to the tasks that helped give it more direction and reduce confusion.

B: What might, or should, be

If we were to do this project again, we would have spent more time in the beginning on getting the task breakdowns right and trying to learn as much as we could on how to do this efficiently. We would possibly also have shorter phase/sprints related to defining and understanding what the project's scope should be to make the team more agile under this process.

In the future, we would have also liked to try a different form of effort estimation, where we would estimate tasks and user stories based on their level of difficulty also using the Fibonacci numbers. Instead we based our estimation on time since we didn't have much experience since hours was something we could more easily try to estimate. This however lead to group members sometimes working overtime or spending far less on a task, because this estimation ended up being more of an average and did not apply to every team member's skill level. Possibly the estimation should be done with the responsible team/developer in mind.

C (A $\rightarrow$ B): Feedback designed to reduce the gap

As will be mentioned later, we were all product owners in this project, however having for example one team member act as the product owner could have helped facilitate the issues with user stories, like their size, in the beginning of the project, since this person would have put more focus specifically on this.

Trying a new estimation technique could also have helped solve issues for all group members in regard to time spent on the project.

1.4 Key Performance Indication

Our **Key Performance Indications** (KPIs) mostly focused on how the team members felt the work went during the sprint. We will discuss each of them and how we felt they affected our work, if they helped us or if we felt they were unnecessary. We will also say whether we would want to use them again in future projects or not.

If you, the reader, are interested in our KPI measurements please refer to the previous team reflections, which can be found *here*. If you cannot click the link, simply visit our GitHub and head into the "Deliverables/Team Reflections" folder.

1.4.1 How many points, out of the week's velocity, did we achieve in relation to our week's target?

A: What was

We used this KPI to check how many points we completed in regards to our velocity. Then we also wrote how many hours each team member put down towards their tasks. This helped us better understand how our estimates were linked to the time spent. This way we could then improve our ability to make better estimates. It also verified that members, on average, put down an equal amount of work each week.

Difference between completed and estimated points	Score
Week 4	4
Week 5	2
Week 6	13
Week 7	0
Average	4.75

The team was satisfied with this KPI, as it provided a very clear answer to how well we estimated our tasks and our available time.

B: What might, or should, be

This KPI could be extended as a chart to make comparison between sprints easier. Since this KPI only indicates if we were successful in our estimates it would benefit greatly from some KPI that could better inform us in how/what to change to better estimate the tasks.

C (A $\rightarrow$ B): Feedback designed to reduce the gap

One way is to look at other possible KPIs available and combine those with this one. Another method would be to try to categories the tasks and see how that category relates to the time it takes to complete the task.

1.4.2 Have we been able to improve problems we identified the week before during the sprint review? (1 - 10, where 10 is major improvement, 5 relatively unchanged, 1 is a lot worse than before)

A: What was

This KPI measured to what degree the group members feel that the problems found the last sprint were solved. It helped us to verify whether a specific approach had worked, or if we needed to try something else. This helped us work more efficient and ultimately, it let us deliver a better product.

The average score during the four weeks that we measured this KPI was 6.29, which means a small improvement. We can also see that during week 5, we greatly improved our task breakdowns and estimates. The improvement can also be seen in our first KPI, where the difference between our completed points and estimated velocity was very low. Week 5 was also the week where we felt the most satisfied with the sprint. During week 6 we decided to create a meeting agenda to structure our meetings, this was however forgotten in the sprint planning and therefore we did not improve. Week 7 our goal was to implement better acceptance criteria and to

use GitHub's code review function. This was implemented a little bit, but most of the group felt that implementing code reviewing more during the last week was a bit overambitious and unnecessarily complicated.

All in all, we didn't improve a lot on the specific things we chose.

Improving from last week	Average score
Week 4	6.3
Week 5	7.86
Week 6	4.857
Week 7	6.17
Average	6.29

B: What might, or should, be

Since we did not improve that much on average something did obviously not work as intended. We would have liked to have improved the things we chose.

Some weeks we chose several things that we wanted to improve, this made it difficult to properly rate how well we improved, especially when we improved one of the chosen things but not the others. Therefore the scores tended to lean more towards indicating that no improvement was made.

We felt like the KPI was decent, but that we did not really work towards it, for example choosing areas to improve but then we did not make the effort to actually improve them in the sprint.

C (A $\rightarrow$ B): Feedback designed to reduce the gap

By choosing only one subject, we would get far more accurate scores each week. We also would need to properly commit to actually work on the things that we want to improve.

1.4.3 How satisfied, on a scale of 1 to 10, is each team member with this week's sprint? Where 10 is very happy and 1 is very unhappy

A: What was

This KPI was used to measure how satisfied each group member was with the sprint. It could be used to verify that all members felt that they were a part of the team and that the team as a whole was making progress with both the project, teamwork and other Scrum related skills.

Satisfaction with previous sprint	Average score
Week 4	7.3
Week 5	7.57
Week 6	6.57
Week 7	7.3
Average	7.18

Overall, most of us were satisfied with the majority of the sprints. We feel yet again that this KPI might have been to general, since the group members could be satisfied with some parts of

the sprint and dissatisfied with other parts, leading to a lower score. At the same time, it was good to have a KPI simply measuring how we felt and when someone put a lower score it was a good starting point to discuss and solve issues that had arisen. It also made our very general KPI more specific whenever it was needed.

B: What might, or should, be

Breaking down the KPI into smaller KPIs would make sure that we could be more specific while evaluating the previous sprint.

C (A $\rightarrow$ B): Feedback designed to reduce the gap

Break down the KPI.

1.5 Acceptance Tests

A: What was

Acceptance tests were performed by the team with the test lists the group has written to see whether the user story not only passes in its technical aspects, but also from a user and value perspective in how well the user story was implemented and in turn how much value it creates for the users.

We also performed acceptance tests with our stakeholder when we had our meetings, where she was able to see different feature and give feedback on how well they worked. However, this was more of a visual test and had its limitations.

The tests were done by performing tests for the features that the stakeholder expressed interest in. The stakeholder then saw the results and came with feedback related to this. The tests also included discussions with open-ended questions to the stakeholder to learn what she thought of the product and its value creation.

B: What might, or should, be

We would have liked to have preformed user tests by having our stakeholder directly test our product on her platform. This would have given us feedback on if the functionality we had implemented was easy to understand. This was however not quite possible with the current situation in mind.

C (A $\rightarrow$ B): Feedback designed to reduce the gap

We could have made bigger efforts in order to get the website up and running online, instead of only on our local computers, which would enable our stakeholder to properly test our product. That would however be at the expense of something else, and we do not have something concrete that we would remove instead. The easiest solution would of course be to meet in person, but due to covid-19, it was not an option. An alternative solution could have been to use something like TeamViewer for remote control testing but this also makes the process complicated for the stakeholder and is unfavourable due to other related reasons like security and integrity.

2 Social Contract and Effort

2.1 Social Contract

A: What was

Our Social Contract can be found on GitHub. Click *here* to access it. If you cannot click the link, simply visit our GitHub and head into the ”Deliverables” folder. We have also handed it in on Canvas, but GitHub holds the most recent version of the document.

We created the contract in the beginning of the course and only updated it once, to remove the requirement to have video on during our sprint planning. No one really wanted it. We did not refer to the contract very often, since we did not run into any conflicts.

B: What might, or should, be

Our social contract was satisfying. We did not need to add anything after we first wrote it, which is a sign that we did it properly the first time. Perhaps we should have referred to it more often than we did, but since we did not really have a need to... we didn't.

C (A $\rightarrow$ B): Feedback designed to reduce the gap

Since we do not want to change anything, there is nothing to write here.

2.2 Time Spent

A: What was

Overall on the course we were to spend 20 hours per week. Removing the time for meetings, we decided that 10 hours for coding including testing was a fair number to strive for. If someone in the group had stuff that made them unable to do 10 hours, the total velocity was adjusted with that taken into account.

Tasks were divided to fill up those 10 (or less) hours for each person and we managed to do that well. Everyone had about the same amount they were suppose to spend. However due to some tasks being inaccurately estimated and also the difference in skill, some group members spent less and others spent more time than what was planned.

B: What might, or should, be

For the future it would have been better if the work hours were more evenly divided amongst the members. The decision to have 10 hours for coding as a standard was a good number, when it wasn't exceeded. Our meetings often took so long that the total ended up being 20 hours and here there is room for improvement as discussed previously.

C (A $\rightarrow$ B): Feedback designed to reduce the gap

To achieve a more evenly distribution, a different estimation system could've been used. Instead of estimating the hours a task would take, the difficulty of the task could be estimated. The person responsible for the task could also have a greater say in its estimation. Another option could be to estimate tasks from a perspective of greatest possible time and then give

each individuals different velocities. To know which method would be best they'd have to be evaluated under individual sprints.

3 Design decisions and product structure

3.1 Design Decisions and Customer Value

A: What was

One of the major decisions we had to take was whether to do a website or an app. When waiting the positive and negative aspects of both the options we came to the conclusion that a website would be easier to implement. After that was decided, bootstrap was picked as our front-end framework since it has a wide library for building neat looking websites.

When it came to what we were going to implement and the design of it, we made most of our decisions based on our stakeholders opinion. For example making it possible to go straight to the map by pressing the address in each pub card.

For the implementation of the features, we made the decisions ourselves as long as the final functionality was what the stakeholder wanted. We did this since it didn't really matter for our stakeholder how we did something, just what we did. An example of our own decision was to use OpenStreetMap.org to implement the map. This was simply chosen because it was free.

B: What might, or should, be

Since our decisions was made based on our stakeholder when it came to costumer value aspects, we feel like it worked well in making sure we made a product targeted at our costumer. Having the stakeholder take part in deciding in the actual implementation would not give any costumer value and it's therefor not something we would want to change.

C (A $\rightarrow$ B): Feedback designed to reduce the gap

In our opinion, no changes are needed in how we made design decisions. What could have been done however is exploring more options through shorter sprints. Another possibility is to let the team work with different libraries in parallel the first sprint or two. This would have allowed us to make a more qualitative decision in how the framework and libraries we used relate to other options e.g. VueJS.

3.2 Technical Documentation

A: What was

Our technical documentation consisted of the comments in the code, both JavaDoc and "normal" comments, and our tests and acceptance criteria on all the user story cards in Trello. There are also some documentation related to the product in the common group chat and on google drive.

B: What might, or should, be

We could have benefited from a design document and domain model which would have helped to get a better overview of the project.

C (A $\rightarrow$ B): Feedback designed to reduce the gap

We could have created a design document and a domain model.

3.3 Usage and Updating of Documentation

A: What was

Our documentation had a high priority and was updated after our meetings on Mondays and Fridays. We either update during or after the meeting depending on what the team decides on. One important tool to help us use documentation properly was Google Drive and there we created several documents with important information that we in the group needed. We also knew that we had to write descriptive comments on every time we were to commit something on Git.

B: What might, or should, be

One of our main goals within the group is to have a good communication and collaboration and therefore we of course want the documentation to be of very good quality in order for us to be an effective working team. Our documentation should be accessible, structured and well-defined on all tools that we use for the project. This means that we can not only focus on one tool but we have to use the same documentation on all tools. For instance, if one person makes a good documentation of comments in a file and then continues to commit that to Git then the description for that commit has to be well-structured as well.

Therefore every team member has to be responsible for creating a good documentation of their own work. We believe that all Git commits must be well-defined for all team members to understand what changes have been made and also future members can benefit greatly from detailed descriptions.

C (A $\rightarrow$ B): Feedback designed to reduce the gap

Our documentation has been very good during the entire project but of course there is room for improvement. To increase a better documentation one important thing to do is to establish which working tools the group shall be using and how they work. For example, we used Trello as well as Overleaf and a few people in the group were unfamiliar with those tools. Sometimes when we worked with reflections or had meetings people would be confused where exactly people are working because they were inexperienced with the tools. In the future it would be good to explain to everyone in the group what tools you will use and how.

In addition, since Drive is such a simple tool it could be good to create a definitive main document where the group writes down the most significant information. By doing this people can go to that document to get a reminder of what they missed, for instance that the team reflections will all be done using only Overleaf. By doing this, we reduce confusion and we do not have to spend time explaining to others who are late to a meeting where we are working. To make better detailed and expressive commits on Git we can take some time during a meeting and look through the previous commits in our repository and note if some commits are poorly detailed. Then we can improve our commits for future sprints and make sure that everyone in the group structure their commits the same way.

3.4 Code Quality and Standards

A: What was

Throughout the project pair programming has been used to keep the code consistent and ensure readability. Since the pairs changed each sprint this also made sure that the team as a whole continuously reviewed code without requiring a formal and cumbersome process. Additionally, since the team strived for readable code its complexity was kept lower allowing for this more integrated and informal version of code reviewing.

Version control was used to allow developers to work in parallel on files and update them together. The team also used branches to extend more complex functionality that would require more work to avoid breaking functionality in the master branch.

Finally, the group discussed how to implement things before working on them and developers tried to be flexible in their implementations while also thinking ahead.

B: What might, or should, be

It would maybe be useful to use the code reviewing tool in GitHub to ensure code quality when the project gets bigger or when larger parts are added from other branches. Adhering to best practices and standards for web-development is also a good idea to write higher quality code.

Additionally, a higher use of branches could make the development process prettier and allow the team to work on separate features over multiple sprints in subgroups.

C (A $\rightarrow$ B): Feedback designed to reduce the gap

To achieve the improvements mentioned above a more formal code reviewing process could be used to ensure that only completed user stories are pushed to the main branch.

To reduce the gap the team can make more use of Git's features like branching and code reviewing. The project would also follow better practices by researching similar projects and reading their code base and looking up best practices.

Making use of design documents would also help get a better overview and create a guide to help write consistent code.

4 Application of Scrum

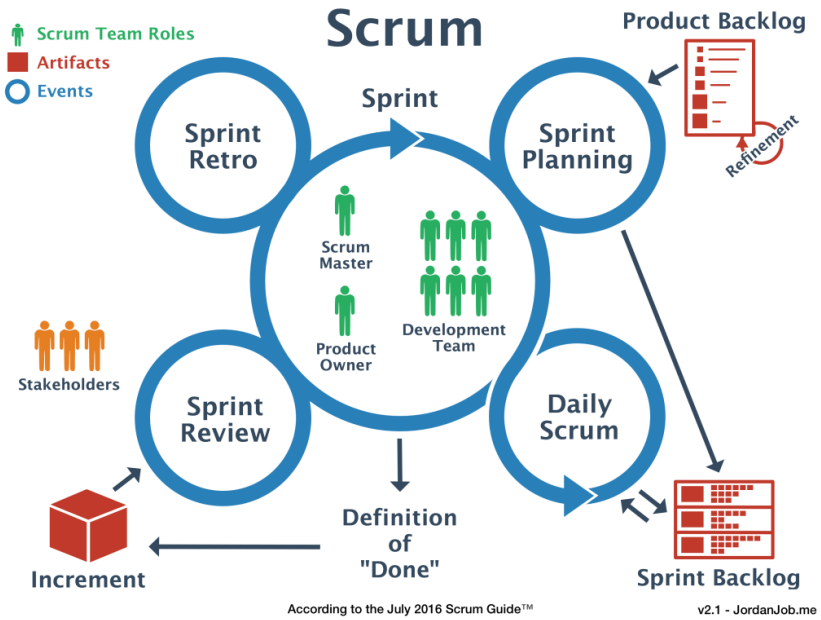

Figure 1: Scrum

4.1 Roles

We used the 3 classic Scrum roles: Scrum master, product owner and the development team. Below, we go into each of their impacts on our work.

4.1.1 Scrum Master

A: What was

Each week, we had a new Scrum master, to let everyone in the group test how it was to be one. The last 2 weeks we had 2 Scrum masters, since we were 7 group members but there were only 5 sprints.

1. Johan Nilsson
2. Thomas Jinton
3. Antonia Welzel
4. Ludvig Lindell and Jesper Lundgren
5. Emma Pettersson and Jennifer Krogh

B: What might, or should, be

We found that, for our project, having a Scrum master was very redundant, since we were all developers. Also, since none of us knew what being a Scrum master really entailed in the context of this course, we did not feel like we were very good at it. Changing Scrum masters every week might not have been a very good idea, since none of us really got a good feeling of what it meant to be one. As soon as we started to feel like we knew how it worked the role was given to someone else.

C (A $\rightarrow$ B): Feedback designed to reduce the gap

If we ever do a similar agile project, we would instead give the role of "Scrum master" to every member simultaneously. What we did as Scrum masters were basically keep track of meeting breaks, and that is something we do not need an entire role for.

Either that, or give the role to one person and let them keep it for the entire duration of the project, so that we have one experienced Scrum master instead of 7 that feel as if they are doing nothing at all.

4.1.2 Product Owner

A: What was

At first, we were unsure of who we should choose to be our product owner. The stakeholder? One of the group's members? In the end, we decided split the role in 7 pieces and give it to everyone in the group, since everyone was already doing the work of the product owner: creating the backlog items, ensuring that they were all understood, and optimizing the value of the work done by the development team. It felt unnecessary to give the role to one person when it was a lot easier for everyone to share the work.

B: What might, or should, be

We felt in general that, for this project, our approach worked well. However, it does not reflect reality. Usually, the product owner is just one person, and their main responsibility would be to keep track of the backlog. Since our group only consisted of seven members, it felt like a waste to have one person only focus on this specific thing. We needed everyone to also be a part of the development team. However, there were issues mentioned earlier in this report that could have been avoided with somebody specifically assigned to the product owner role.

If we had done this project again, we would have probably had a similar group composition, on the other hand, if we had more members, changing the way our roles worked would enable everyone to focus more on their specific role. This would be preferable, of course, but we had to make do with the resources we had.

C (A $\rightarrow$ B): Feedback designed to reduce the gap

If we had more members, we would simply dole out the roles in a different way. One Scrum master, one product owner, and one development team (consisting of several members).

4.1.3 Development Team

A: What was

Our development team consisted of everyone in the group. We needed every brain we could get to maximize the value of our product, and everyone had prior experience being a developer. Everyone had to juggle being both a product owner and a developer, and sometimes also a Scrum master. This meant we could not focus all our 20 hours on being developers, and instead spent only around 10 hours each week.

B: What might, or should, be

See the previous section, ”4.1.2 Product Owner”.

C (A $\rightarrow$ B): Feedback designed to reduce the gap

See the previous section, ”4.1.2 Product Owner”.

4.2 Agile Practices

A: What was

We used several agile practices during our work. Our framework was Scrum.

- Backlogs
- Daily Stand-up/Daily Scrum
- Pair Programming
- Retrospective
- Scrum Events (sprint planning, sprint review and retrospective)
- User Story
- Velocity Tracking

We started off with a structured planning and with a lot of focus on prioritization. To collect all tasks and user stories a Scrum board was used. These were then used during the sprint planning each week which finished with a sprint review. This aimed to collect and process the teams shared experience to help plan the next sprint and improve the overall team work. This process is also meant to help us implement the agile methodology, especially the steps of evaluating current state, tasks to be done that create value and a review of the results and methods that brought us there at the end of the week. Also at the start of the project we had a clear stakeholder and so we had a lot of ideas of how to make the application satisfy our stakeholder.

We had our daily Scrums bi-daily since all team members had at least one course course at the same time and we figured that there was no point in having a meeting if no one has been able to work after the previous meeting. Another reason for having it bi-daily was that other courses often took up a whole day, as some of the team worked on their *Bachelor's Thesis* and needed to spend a great amount of concentrated time with their group to work on that.

B: What might, or should, be

Of course we wanted to achieve all of the agile practices the best we could and so we had a lot of focus on reaching those goals. We believed that having planned meetings and weekly talks with each other would increase our understanding and experience with the agile practices. In addition, one thing that might benefit the whole group's understanding of the agile practices would be to choose one person in the group to be a sort of "agile master". This person would make sure that the team stays on track with the principles during the project.

Our goal was to have a smaller meeting every Wednesday to give people the opportunity to show how far they have come. In addition, our goal was to take finished functions of the application and show it to our stakeholder and receive some feedback in order for us to improve our tasks and add new functionality to the website.

C (A $\rightarrow$ B): Feedback designed to reduce the gap

Perhaps it would have been better to have the daily stand-ups actually be literally daily (instead of bi-daily), which is something other groups did. This is something we plan on doing in the future. Because the stand-ups will be daily then the group can make shorter meetings(sprint plannings and sprint reviews will have to be longer of course). In order for the group to be flexible and effective the shorter meetings can be via telephone voice-chat and the longer meetings can be on Zoom. This is something that the team has to determine early in the project so that people for example have their phones ready for the short stand-ups as well as microphones fully charged for the Zoom meetings.

Another thing we wanted was to stay in touch with our stakeholder as much as possible and this was something that we could have improved on a lot. For example, our stakeholder was not available all the time and we sometimes contacted her directly to plan time for her to join our meetings to show her our progress. Our meetings with the stakeholder were not properly scheduled and this has to be improved for the future. To accomplish this we have to be a bit forward with our stakeholder and agree on what times are most suited for a longer meeting and what times are better for a shorter meeting. This means that the group has to be flexible and adaptable, to make sure that all important meeting happen. People have to be responsible and prioritize their schedules so that if the stakeholder can only have a late meeting one a specific day, then the group's members have to make sure that they do have any conflicts in their timetable.

For a future project or a future update of this website, we would like to plan weekly meetings with our stakeholder and make sure that every week is convenient for her. By doing that we would increase our communication with our stakeholder and also we think that our final product would be better with more feedback and suggestions from the stakeholder. In addition, we would discuss different roles that some of the people in the group can have, for example that one person is the master of the agile parts. In the case of confusion or if some people get stuck when working with the project then the agile expert can advise and help everyone so that the project stays on the agile path and that the entire process does not turn into a disorganized mess. Of course the person who is chosen to know more about agile has to take the responsibility to learn and study about that subject so that person can help the entire group for the remainder of the project. This applies to the other members in the group who are assigned a role.

4.3 Sprint Review

A: What was

Our sprint reviews took place on Fridays at 13.00. The idea was to have all tasks fulfilling our **Definition of Done** (DoD) and all code ready and pushed to the master branch. Our DoD criteria were as follows:

- ☑ The code is well written and properly formatted
- ☑ All the acceptance criteria have been met
- ☑ Thorough tests are written
- ☑ All the tests are passed
- ☑ The code is commented
- ☑ Code is pushed into master

We usually did not meet all of the criteria mentioned above. Usually there were some minor last minute fix left to be done which we usually did during the meeting. The fix needed was not very time consuming since there were usually someone in the group that knew how to do it. In that manner it was a quite efficient way to solve problems since there were usually only 1-2 problems needed to be solved per meeting. However that was not the intended way to do it.

B: What might, or should, be

We should have made sure all code was checked, finished and pushed to the master as well as meeting all the DoD criteria. This should have been done before the sprint review starts so that we officially meet the DoD. The documentation on Trello has to be properly done as well so that only finished tasks goes into the DoD section.

C (A $\rightarrow$ B): Feedback designed to reduce the gap

Focus on the above mentioned improvement areas during every meeting and trying to improve over time. If we had received a few more weeks to work with those challenges we would probably overcome them. One important thing to reach the DoD would be to have meetings where we discuss that topic and help each other to finish tasks that we have during sprints. For example, we discussed that one good thing would be to have a morning meeting before the sprint review where we look at conflicts in Git, code errors, solution suggestions etc. During the project we noticed that we solved a lot of problems when we worked together on Zoom and so we realized that we could use our meetings to solve problems and not only discuss what everyone is doing.

4.4 Best Practices

This section will go over the tools and technologies we used when developing our website. Were they helpful to our work, or did they just make it harder? What would we change in the future? These are some of the questions that will be brought up, together with their answers.

4.4.1 Integrated Development Environment

A: What was

We used IntelliJ IDEA as their IDE. It has great integration with Git and GitHub, our chosen VCS. Since we get the Ultimate edition by using our student emails, and since most of the members have used it in previous courses, it felt like a comfortable choice.

B: What might, or should, be

We had no problems with the IDE that could not be solved with a simple Google search. It was very simple to use, yet robust, and had all the features we needed. We would use it again.

C (A $\rightarrow$ B): Feedback designed to reduce the gap

Not applicable.

4.4.2 Version Control

A: What was

For version control we used git through GitHub. Many in our group had already used it before, but some were new to it. For the ones that were new, the experienced group members simply walked them through the process of using it.

B: What might, or should, be

For a future project, we think that using more of GitHub's tools would be beneficial to the work process. Code reviewing in GitHub is one of the tools we only looked at briefly but which we think would be a good practice since it would ensure code quality in a bigger extent. This would depend on what kind of project it would be as well as it's size. For the project we did, we didn't feel like the extra time it would take to do code review would justify doing it.

C (A $\rightarrow$ B): Feedback designed to reduce the gap

To use the code reviewing function you would have to work on branches which our group didn't do a lot in this project. So in order to make this change, the group would have to decide from the start that we are going to work on branches, and that the code must be reviewed before being merged with the master branch.

Doing code reviewing would take time. In the beginning of the project we didn't have much leftover time and there is nothing we would want to drop in order to have time for that. However, as we got used to sprint plannings and got more efficient at that, we got more time to spare which could've been used for code reviewing.

4.4.3 Scrum Board

A: What was

We used **Trello** for our Scrum board, since several of the members were already familiar with it. The free version also had all the functionality that we felt we needed. It was easy to use and

gave us a great overview of our project. Some people had difficulties understanding it at first but as soon as we got further with the project everyone learned to use it properly.

B: What might, or should, be

For our type of project, we would use Trello again. It was not overly complicated and the tool let us do everything we wanted to do. Something like Jira would just make it harder to do what we need to do. Using something that is familiar would also make us more effective and that would be beneficial for the group and the outcome of the final product.

C (A → B): Feedback designed to reduce the gap

The current solution was very good and we do not see the need for any changes. However, if a new Scrum board tool is created in the future and it exceeds what Trello has to offer then we have to reconsider our choice.

4.4.4 Server

A: What was

Since we had a server to run, we decided to use NodeJS for that. Most in our group was new to that but like with GitHub, some people had used it before and could therefor explain to the rest how to use and run it. We also added a short explanation to our ”read me” file in our repository in case anyone needed to see the information again.

B: What might, or should, be

NodeJS was simple to use, so we would have no issues using it in the future. In the beginning however, there were only some group members who knew how to use it. For those members who worked on functionality that wasn't dependant on the server, they felt no need to learn how to run it since they could check their work regardless. This became a hindrance when these persons later were going to work on functionality requiring the server and didn't know how to use NodeJS. It was solved fairly quick by them getting to learn how to use it too but it would've been best if everyone knew from the start.

C (A → B): Feedback designed to reduce the gap

The moment it's decided to use something like NodeJS, the group should have an ”official” moment dedicated to everyone learning how to work with it, in order to eliminate possible roadblocks. One option could be to spend an initial sprint on defining the project's core scope, libraries and frameworks and then towards the end of that sprint include a check/test to verify that all group members get familiar with this. On the other hand, more specific parts of the scope and frameworks should be pushed into future sprints to allow the group to remain agile and uncommitted to specific implementation details.

4.4.5 Communication

A: What was

We started with using Facebook messenger for text communications and also our first meetings. We decided that it was better to use Zoom for our longer meetings since it was easier to

share screens and show our processes that way. None of us had used Zoom before the distant education started and we learned how to work with as the time went on.

B: What might, or should, be

We think these application worked well for our project. We entertained the idea of using Slack instead of Facebook messenger since it is widely used on Chalmers. We decided that messenger was the better choice since it has voice chat which we used for our short daily stand ups. For this reason we also think that it would be the best choice for a future project, if that too would be done on distance. Slack also takes more effort to use, creating channels and so on. Everyone was already on messenger and therefor it was the easier choice.

C (A $\rightarrow$ B): Feedback designed to reduce the gap

Since we were happy with our choices of communication, there are no changes that needs to be made. What we have to be aware of is when we use what kind of communication. For instance, Zoom will be used for all the longer meetings. In addition, if it so happens that a new modern communication application is created in the future then of course we have to be adaptable and see if we can use that and perhaps that application will improve our communication.

4.4.6 Writing

A: What was

We used Overleaf and Google drive to collect our documents. These sites have been widely used by the majority of the group before and the choice to use them was easy. We used Overleaf for our ”serious” documentations such as Team reflections and Sprint Retrospectives, while we used Google drive for taking notes from meetings and writing our presentation.

B: What might, or should, be

We think that for the future it would be better to have all documents collected in one place. Overleaf would be the better choice because of the cleaner structure. For the stuff that might not feel fitting in Overleaf we argued that those files could be directly uploaded to the GitHub repository.

C (A $\rightarrow$ B): Feedback designed to reduce the gap

Simply deciding from the start which site to use and stick with that until the end. This would lead to less confusion and less time spent looking for the right documents. It also reduces the risk of important information being lost due to being added to an unused document.

4.5 Relation to Literature and Guest Lectures

In this section we reflect upon similarities and differences between our experience and others.

4.5.1 Literature

A: What was

We used several websites to learn about HTML, CSS, and JavaScript. One of those was w3schools. It helped a lot with learning the new languages, something that most members needed.

Stack Overflow was also used a lot, when the team ran into problems and needed a push in the right direction.

A lot of other sites were also used, too many to list here.

B: What might, or should, be

We should have read more about web development and Scrum. We were very new to all of this, making what we did in the beginning still worked but took time. We jumped head-first into something we knew almost nothing about and our project therefore had a slower start. Maybe a sprint dedicated to learning about Scrum and then another sprint that targeted learning about the project field (in this case web development) would have been good to prepare the team for the future sprints.

C (A $\rightarrow$ B): Feedback designed to reduce the gap

The next time we do a similar project, we will prepare ourselves more by spending an initial sprint to prepare the team for the future sprints making use of Scrum as well as the project's technical aspects. This because such an investment will mean less time wasted and a more conceptualised project scope and minimum value product.

4.5.2 Lectures and exercises

A: What was

Several lectures and exercises were held during the course, most of them in the beginning.

- [Lecture] Introduction to software engineering and the course organisation
- [Exercise] Collaborative drawing exercise and lecture on Key Performance Indications
- [Exercise] Project scope and pitches
- [Lecture] Scrum details
- [Exercise] Agile RE and ”Slicing the cake”
- [Guest Lecture] Lecture on distributed team work (Husqvarna)
- [Lecture] Signing off

These sufficiently prepared us for the project, and we could go back to looking at the slides when we needed to refresh our knowledge on Scrum.

The guest lecture from Husqvarna helped especially much since it prepared us for working together from a distance, due to the corona virus.

B: What might, or should, be

The information on the final presentation and the final report (”Signing Off”) would have been good to get a bit earlier (as well as when we got it), since we were very unsure of how these would be handled before the lecture was given.

C (A $\rightarrow$ B): Feedback designed to reduce the gap

We could have tried to find out the information for ourselves or bring the teachers’ attention to potential problems by writing them an e-mail or starting a discussion on Canvas.