Final report

Julia Giaro, Victor Hui, Love Lindqvist
Hampus Rhedin Stam, Arvid Svedberg

DAT257, Informationsteknik, Chalmers Tekniska Högskola

Contents

1 Customer Value And Scope

The chosen scope of the application under development including the priority of features and for whom you are creating value

A

We were given an idea by a representative of EWF ECO, a company in Sweden that distributes and maintains "Big Belly Wastebins" (The big green wastebins that you can see spread around the city). Their pitch to us was that the current system of emptying these wastebins didn't make use of the fullness detection features these wastebins have. Currently the garbage collectors empty all wastebins regardless if they need to be emptied or not, which they considered to be ineffective and wasteful. They told us that you could get the data for the bins' fullness from an API, so we got the idea that you could make a route planner application that would take into account the bins' fullness when creating a route. We also thought it would be neat if this route application created routes that were somewhat optimized in order to save time for the garbage collectors. The application would provide value to both the garbage collectors since it would make their planning easier, but also to their employers who would benefit from the waste bins being emptied in a more efficient manner.

Throughout the project we had some difficulties regarding the prioritization of features due to a multitude of factors. First we had problems with acquiring a project idea and a stakeholder, since our initial plan and idea fell through. In the end we managed to find a stakeholder that could provide us with a reasonable project idea, but the planning stage ended up being suboptimal since we were unsure what project idea we would end up with. Our first meeting with the stakeholder ended up being during the first sprint, instead of before the first sprint which would have been more ideal. These factors together led to a project scope that wasn't fully fleshed out. We knew what we wanted to do, but all the details weren't concretised which led to some confusion during the latter stages of the project regarding the prioritization of features amongst team members. Another contributing factor which made the prioritization unclear was that the stakeholder had a lot of different ideas which they threw out during the first meeting, and since they

didn't have a clear vision of the application themselves we were unsure which features were essential and which were not. These issues resolved themselves as we continued working with the application and had more meetings with the stakeholder, as they also seemed to gain a clearer understanding of what they actually wanted. Towards the latter stages they even told us what features they considered "must have" and which features were "nice to have", which made the prioritization of features much easier. For example there were a lot of discussion surrounding a feature which would allow the user to input a time and have the application generate a route that would allow you to empty the most amount of wastebins in that time frame, but during the later meetings the stakeholder told us that this was just a "nice to have" feature instead of a "must have" so we deprioritized it and it wasn't a feature that ended up being included in the final version of the application.

B

A problem with this project was that we had a lot of options of what to do due to the amount of data you could get from the API and thus subsequently a lot of ideas, narrowing down these ideas would have helped a lot in regards to prioritization of features. So essentialy we would have liked a more clear and concise project scope, since it would have been easier to know which features should be prioritized and where to concentrate our efforts.

C

While it was nice that our stakeholder gave us such free reign, we should have had more of a dialogue surrounding what was expected of the final application. If we had been more clear that we wanted to find a minimal viable product that both parties agreed upon, they would have probably mentioned which features were most important to them, a lot earlier. Also we should have spent more time during the planning stages so that we could have a more fleshed out project scope, as well as getting everyone on the same page in regards to what we were working toward.

The success criteria for the team in terms of what you want to achieve within the project (this can include the application, but also your learning outcomes, your teamwork, or your effort)

A

We wanted to learn some new frameworks and tools including React.js and Springboot. Trying to learn rather big and advanced tools during this project was however a huge mishit. A lot of the time during the sprints went to research and learn how to do things including syntax. In the end no-one feels that fully understand the tools. If we were to redo the project we would probably use tools we already know. Unfortunately the only graphical tools we learnt so far is JavaFX, which is a developers nightmare. We also defaulted to using a Java-backend, when other backends were probably more suitable for our project.

In the social contract we said that the group should aim for even workload measured in the difficulty of a task. The group agrees that everyone has equally contributed to the project. The teamwork has worked well and everyone have helped each other during the project.

B

We wanted to understand the new tools better than we actually do. If we had used tools we've used before there wouldn't be as much time-effort to research during sprints. At the same time using some new tools was almost unavoidable, so we should have researched these new tools more thoroughly. The situation with the Java-backend would have been avoided then.

C

If we were to redo or do another project we'd use tools we know or, if there is time, properly research and get used to the tools before starting the project. There is will to continue with the project both within the group and the stakeholder but then we should learn the tools better.

Your user stories in terms of using a standard pattern, acceptance criteria, task breakdown and effort estimation and how this influenced the way you worked and created value

A

We have used Trello as backlog and arranging user stories. Every user story has acceptance criteria, a effort estimation and who are working on the story. When a user story is done it is moved accordingly to the sprint it's done. Trello has been great during the project and has made the scope easier to plan and see. Our goal was to use GitProject instead of Trello to have everything in one. No-one was well experienced with GitProject and since there were already other tools to learn and everybody was well-known with Trello that is what we settled for. One problem has been our effort estimation, we often under-estimated the user stories. Researching and implementing tasks took a lot more effort than we thought. This has improved during the project but even in the end it was hard for us to make accurate estimations.

B

Our estimations have improved since the beginning of the project but we still under-estimate a bit. If we were to continue the project we would get better estimations. We would learn the tools we used which would make tasks take shorter amount of time. We would also get a better feeling for how much effort the tasks takes.

C

By using SCRUM and agile management i future projects and courses we will get better at estimating a user stories effort value. Also by continuing researching and learning the tools we used we will be more prepared before next project. It's a lesson learned to not use to many or complex tools when doing a project with a small time limit.

Your acceptance tests, such as how they were performed, with whom, and which value they provided for you and the other stakeholders

A

The group have not made any acceptance tests.

B

The group agrees that acceptance tests would be nice and completely understand their purpose. Why it haven't been a priority is because the project does not imply any concrete features, it's more of a algorithm and path finder. Time constraint is also a cause why there aren't any acceptance tests.

C

The group felt that the program should be more finished or at least have a more refined version before acceptance tests become relevant. There should be an actual task in the program for the user to test, which there is not in the current state.

The three KPIs you use for monitoring your progress and how you use them to improve your process

A

The KPIs we used were burndown chart, velocity graph and team value. The KPI have been very low prioritised and the group does not feel that they have yield to any positive outcomes or value.

B

In later projects it's a possibility that KPIs should be more prioritised. For this kind of school-project with a small time span the KPIs could be a little redundant but the group agrees that in a larger project they would be nice.

C

It's good to learn and get comfortable in how KPIs work and their purpose before taking on a big project. Maybe this is something to do after the course now that its finished.

2 Social Contract and Effort

your social contract (Links to an external site.) (Links to an external site.), i.e., the rules that define how you work together as a team, how it influenced your work, and how it evolved during the project (this means, of course, you should create one in the first week and continuously update it when the need arrives) There is a survey (Links to an external site.) you can use for evaluating how the team is perceiving the process and if it is used by several teams it will also help you to assess if your team is following a general pattern or not.

A

The social contract made in the beginning of the project looks as following:

1. If you're not able to attend a meeting then you have to accept what the rest of the group has decided upon
2. Notify the rest of the group if you can't attend something/something comes up
3. Be on time!
4. Finish the work that you agreed to take on
5. If you delegate your work to someone else you should trust them to handle it as they see fit
6. Try to balance the workload amongst the members
7. Schedule group related work during school hours
8. Decide meeting times at least a day in advance
9. Cancel meetings at least a day in advance
10. We will try to have meetings in person if possible

The whole group was present in most of the meetings but when someone was not attending the team was still able to make changes and decide new things

without the approval of the members being away. But even though this was something we came up with and agree on we still made sure with everyone that bigger and drastic changes was ok.

We are also happy on how everyone was on time. Of course team member sometimes got a bit late due to traffic etc but the rest was then notified on social platforms about the delay of arrival and everyone was fine with this.

Everyone did in the most cases finish the tasks they accepted to take. In some situations members was not able to solve the given task, but in these cases the team was notified about it and could help out. We were pretty forgiving when it came to this stuff because a lot of the work that had to be done was unknown for us and therefore could take longer time than anticipated.

We made sure that everyone had a user story to work on every week and tried to balance the work evenly. Two of the members was away one week in the beginning of the project but then made sure to work a bit more during the Easter holiday to balance the workload among us.

All meetings were during school hours as decided. A majority of the future meetings were also planned about a week in advance and no meeting was cancelled.

A clear majority of the meetings was also in person as we together decided. But in some circumstances we choose to have meetings online.

B

It's arguably better to update the social contract during the project which we did not do. Like specifying some things like, "If you can't finish the work let the others know". Also change, "Be on time" to, "It is important to be on time but if it is not possible, notify the others". But we choose to not update the contract since we did not feel like we had a reason to. If there for

example was a problem with someone coming late to meetings frequently we would consider changing the social contract and especially remind everyone about it. But in our case we would much rather spend time and effort on other things.

C

A short meeting each sprint discussing how everyone feels about the social contract and how it is followed would remind the group about everything that's been decided. But during such a meeting there are also room to discuss if something needs to be added or changed.

the time you have spent on the course and how it relates to what you delivered (so keep track of your hours so you can describe the current situation)

A

We have on average spent a bit less than 20 hours a week. We are pretty satisfied with the final result of the project since we achieved the MVP and even added some additional features. We expected a bit more but there was a lot of new things to learn and research about, and that process took a up a big portion of the work on this project. So having this in mind, we are satisfied with the final result.

B

The delivered product would have been a lot better with the same amount of time spent by the team if the research process wasn't as long for so many tasks.

C

We would have done a bit more research in the beginning when deciding what code languages and programs to use. We chose google maps without any hesitance for example. This was both because of our lack of knowledge that

similar services existed and because google maps was used by the stakeholders company. A few weeks after working with google we encountered some problems. If we took a short discussion before every similar decision and done some research we would maybe have noticed that google maps wasn't the best map API for us and that it had it's flaws.This type of research would maybe have led us onto other paths that could save us many hours or researching and having to redo stuff because it didn't work as expected.

3 Design decisions and product structure

How your design decisions (e.g., choice of APIs, architecture patterns, behaviour) support customer value

A

Our project is based on the Google Maps API as it was the most developed alternative of the ones available. We felt as this was the easiest one to implement as there is a lot of information publicly available on how to implement and customise it. We also agreed on that this was the option the stakeholder was the most comfortable with. The teams prior knowledge on architectural patterns doesn't extend past the Objected oriented programming project course. Therefore our use of architectural patterns isn't surpassing that. We were not consciously adapting the program to fit a specified persona, but the stakeholder had ideas of what features and design they wanted so that was what we provided. It started off with some designs in FIGMA and then turned to applying features (customised way-points, information popups) to the google map component. During the course we had a continuous dialogue with the stakeholder on their opinions on what behaviours the finished product should have. The team also provided demos of the programme every stakeholder meeting to ensure we all were on the same page.

B

We discovered that there are other map API alternatives which may have worked better as google maps has been found lacking in some areas. One of these alternatives is Mapbox (our mentor recommended this as she used it for her project), an easy map API to implement in react that displays everything we needed for this project. Other than that it would be beneficial for the stakeholder if another map API was used as google maps for businesses requires a license. The knowledge of architectural patterns would provide a better base for development of the program as it provides more structure and understanding of the project as a whole. Behaviourally a persona would provide a base understanding between the developers and the stakeholder on what is expected. Our stakeholder didn't have clear guidelines of the product from the beginning, therefore a persona was not an option.

C

A better understanding between the stakeholder and developers to ensure a guideline of what the stakeholder is expecting in the beginning.

Which technical documentation you use and why (e.g. use cases, interaction diagrams, class diagrams, domain models or component diagrams, text documents)

A

We have some JavaDoc documentation of the project and some sequence diagrams for the project. As our team consisted only of IT students we had a good understanding of the program structure, and documentation other than the one we have was not considered as a priority.

B

The team agreed that technical documentation may have had impact on the project, unfortunately as the time spent on learning new tools exceeded our expectation the documentation was put on hold.

C

Incorporating better documentation to improve understanding of the whole project for everyone involved.

How you use and update your documentation throughout the sprints

A

The team had "Document code" as an acceptance criteria. We also found it necessary as errors due to lack of imported packages occurred.

B

A better understanding of how documentation should be written would provide a better structure of what needs to be documented.

C

As our group consisted of IT students, documentation was not our top priority. However in future projects this may not be the case as we meet people from different backgrounds and then documentation is crucial to ensure everyone is on the same page when it comes to the code structure.

How you ensure code quality and enforce coding standards

A

As we didn't have any tests because we found it hard to implement for our project, the team reviewed code by creating pull requests on GitHub whenever we had a new feature to add. This ensured that we avoided merge conflicts and had an overlook on what the others were working on and if it met our established coding standard.

B

Due to us being unfamiliar with JavaScript and the react framework, a document for coding standards was difficult to establish.

C

Before the next project, research about how coding standards look for the different languages should be done. The team should establish coding structure from the beginning, much like the social contract is written.

4 Application of Scrum

The roles you have used within the team and their impact on your work

A

We were not good at utilising the traditional scrum roles in or project. Since we started a bit on the back foot, a lot of time was spent on catching up in terms of coding and planning, which resulted in us neglecting properly incorporating the roles of scrum master and PO. Assigning a scrum master was something that we briefly discussed and actually ended up doing but since we didn't properly define our expectations on the scrum master, the scrum member ended up being pretty much the same as a normal team member. The lack of a scrum master wasn't too bad since we still managed to have productive sprints throughout the project, but the absence of a PO was definitely noticeable at times where we struggled with prioritisation of user stories.

B

Since it didn't feel like we got to have a proper go at utilising these roles, it would be nice to utilise them in a future project. The project ended up being fine without these roles as it was fairly small, but it definitely could have been a lot easier if we had utilised them more.

C

The main problem with utilising the roles was that we never really had a conversation about the responsibilities and expectations placed upon these roles and how the roles should be assigned(whether a single person keeps the role throughout the project, or if it rotates on a weekly basis). This is something that should have been done but we just missed it, due to there being other issues with the project that we at the time thought were more important. We realise that the roles are important and that for our next project we should not neglect them just because we have a lot on our plate.

The agile practices you have used and their impact on your work

A

We utilised sprints in order to develop the project in short cycles. This forced us to work on the project through small incremental changes instead of one big change. This was very helpful when we needed to change the direction of the project due to the stakeholders feedback. It also forced us to break down larger user stories into smaller ones that were more manageable, so that we could finish them within the allotted sprint time frame(one week). Our way of working was flexible due to the agile practices. For example if the stakeholder wanted a certain feature that they hadn't mentioned before in the application, we could write a user story and start working on the feature during the next sprint. It was nice to get continuous feedback from the stakeholder since we couldn't stray too far from their vision without them telling us, which resulted in an application that the stakeholder was happy with.

B

We had some troubles writing good user stories. There were some that ended up being not completely independent, which forced us to coordinate the merges more than we would like. Some user stories we didn't finish in the span of a week, a lot of the time due to us underestimating the amount of time it would take to learn how to implement a certain feature in programming languages that we were unfamiliar with(e.g JavaScript for react) or using tools/technologies that we were unfamiliar with (e.g google maps API, Big Belly API(API for the smart waste bins). We did try to take into account the time it would take for us to learn how to implement the feature instead of just the time it would take to implement the feature. However this proved to be quite difficult since sometime you would get stuck at something unexpected or the feature would be harder to implement than expected. For a future project like this we would want user stories that are better estimated(although this might be difficult due to the reasons mentioned above)

C

We should have spent more time ironing out our user stories, this would have helped with having more independent user stories. When it comes to our underestimation of the time it takes to finish the user stories, it feels like a lot of our mistakes on that front were due to inexperience. If we were more familiar with the language and tools/technologies we were working with, it would be a lot easier to estimate how long it would take to finish a story. Perhaps we should have done more research on the things we need to learn in order to complete an user story, in order to gain better insight into how long it would take us to learn these things.

The sprint review and how it relates to your scope and customer value (Did you have a PO, if yes, who?, if no, how did you carry out the review? Did the review result in a re-prioritisation of user stories? How did the reviews relate to your DoD? Did the feedback change your way of working?)

A

As mentioned before, we practically did not have a PO. This made the sprint reviews very lax and vague, making it harder to connect progress to customer value, and holding people accountable to what they had achieved in the sprint. Our sprint reviews became focused on how much value, relative to the expected value, we were able to produce to the stake holder, and how much it furthered our progress to the finished MVP according to our DoD. The main thing this accomplished was giving us motivation for decisions for the next sprint planning. It also calibrated the expected work of tasks, though this effect was more passive, as the misestimation (often overestimation) was only briefly discussed.

B

Customer value should have been a larger part of the sprint reflection, along with a more thorough evaluation of work/effort (both the assignment of them to the completed user stories, and the spent work/effort by each member). This would have helped us stay more focused on the user stories that create

value to the stake holder, and hold people accountable to the work/efforts they do (the increased accountability could then be focused on forcing us to evaluate our time, and how much of it we can spend).

C

Having a PO to help keep us focused on customer value during the sprint review would have been beneficial, to more actively stay focused on relevant user stories. Also, defining and evaluating work/effort expectations would push us to make use of the time that we have. This is perhaps a powerful tool we could have used to make the time we had on the project as efficient as possible.

Best practices for learning and using new tools and technologies (IDEs, version control, scrum boards etc.; do not only describe which tools you used but focus on how you developed the expertise to use them)

A

In this project we worked with a react front end which was unfamiliar for all group members. We mostly utilised different tutorials(YouTube, written ones) in order to learn how to write the code that would achieve what we wanted. This proved to be quite problematic at times when we couldn't find tutorials that matched our specific setup. A lot of trial and error went into getting our code to work. We mostly learned by trying out some code we found online and then tweaking it to suit our use cases. For the backend we chose to work with Java since all members of the group were most familiar with Java from previous courses. However this might have not been the best choice, since there weren't many tutorials for how to connect the Java backend to the react front end. We did manage to find one tutorial that showed us how to do it, and by following that tutorial and then changing the code to fit our use case, we got it to work. Basically the learning process for both backend and front end was very similar, mostly trial and error. As for the non code related stuff(IDEs, version control, scrum boards) we mostly stuck to tools that we were familiar so there wasn't much of a learning process.

B

The learning process of react wasn't the easiest, it took us a lot of time to implement the features that we had envisioned. Ideally it should not take us that long to learn how to implement things, so that we instead could focus on the actually implementation. We should have probably used a backend that was more commonly used with a react front end so that we would have more resources available for us to learn from.

C

We could have made it easier on ourselves by researching the tools we were about to learn more in detail. For example the amount of React.js with Java backend tutorials being lacking would have been discovered quite early on and we probably would have moved towards a different backend. For the next project more thought should be put into the learning process instead of us just jumping right into it. Perhaps some kind of plan for how we should learn to use the new tools would be beneficial.

Relation to literature and guest lectures (how do your reflections relate to what others have to say?)

A

We used as much information from the lectures in the first few weeks as we could. To supplement this, we googled practices and buzzwords to gain a better understanding of how the agile process worked, and which practises were best to implement in our project. However, none of us had a great understanding of agile beforehand, and none of us were used to using the work practices. This made our implementation very improvised and not very refined. We spent a lot of effort trying to figure out how the agile workflow worked, and what that would mean practically for us while working. As an example, we had issues defining a DoD, both as we did not yet know the scope of the project, but also because we did not know the scope of the DoD: what to include; how specific to be, or not to be; how important or big different roles were; and so on.

B

Ideally, the process of starting a project should be defining roles, DoD, Social contract, and other documents, with everyone already understanding their implications and how to create them in a clear and concise way.

C

Through this course, we have had to deal with issues that come up when the agile work flow is not followed. With that we have been able to more easily identify them, and how working agile would have prevented them. This has pushed us to work more agile during the course, to prevent further time loss to for example: working on user stories that do not contribute to stake holder value; dropping (or postponing) user stories because of ideas or issues mid sprint; creating too wide user stories that take too much time to implement. Also, having gained a better understanding of the technical documents involved in the agile work process, we have improved our ability to create and understand them.

5 Last meeting with EWF ECO

The last meeting went well, the stakeholder was happy with what we had accomplished with the application. They mentioned a possibility that the project or at least the project idea would be developed further by people on their side, so there will perhaps be a meeting next friday to share thoughts and ideas we had when we were working with the project. This has been a very fun project and we learned a lot throughout it, seeing the idea being developed further would be awesome!