

Team Electric

Final reflection

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 initially created 4 epics, but were only able to deliver the first two. However, early on in the project we realised that we would not be able to deliver all four epics so we adjusted our minimum viable product accordingly. We were able to easily evaluate the priority of the features and kept that throughout the whole project. Our main focus was on having all the bathroom locations appear on the map which we managed to implement. So in that way we created value to our customer.

B: Of course we would have liked to be able to deliver all four epics. By realising epic 3 and 4 we would have been able to deliver more value to the end user. Right now all they can do is see bathroom locations on the map with some small information about the location. By implementing other features such as a filter, being able to save locations, and add reviews our application would have been an even more unique experience from what is already available on the market.

A->B: While we are happy with what we prioritised in terms of scope, we did underestimate the amount of time it would take to set up the development environments, to understand the tools we were using, and for researching how things should be implemented. In the future we will plan with these obstacles in mind, as well as build a better ground work so we are able to focus on implementing features instead of spending so much effort on setting up.

- 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: In terms of the application we are pleased with what we managed to accomplish. Our team had good and consistent communication throughout the project which made the process much more efficient. In terms of effort, some teammates have spent more effort on the project than others. However this can also be due to external factors (personnal, technical problems). At times we may have lost sight of the use of Scrum and focused too heavily on creating our product.

B: While we did communicate throughout the sprints, certain kinds of communication could have been improved. We should have spent more time explaining our code to each other and ensuring that we all understood. As well as communicating and working together on solving the technical issues some teammates had that hindered their work. In the future we want to better implement all of Scrums features.

A->B: We could spend more time working in the same space, or doing some kind of team building exercise. We could spend time looking back on what was delivered and what is expected instead of rushing ahead. Ensuring that the whole team is on the same page is an important element to keep the initial goal in mind. We could have also used the trello board more extensively throughout the week which would have led us to better understanding Scrum. By keeping the user stories and the acceptance criteria in mind and also holding each other accountable to this.

- 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 kept our initial user stories through the whole project since we made sure to follow every rule given while creating them. Initially we created tasks that were too big and required too much effort. The tasks were also not independent and didn't follow good "slicing" practice. Creating smaller tasks made the project feel less overwhelming and the team was more encouraged since we were improving faster. We did not follow good effort estimation practices. Some sprints we did the effort estimation, others we did not. Due to our lack of knowledge on the programs, effort estimation was difficult.

B: We would have liked to spend more time breaking down tasks in order to have more of a discussion on how to resolve these tasks. We should have also been more consistent in our effort estimation since it did help us on certain sprints. Our acceptance criteria could also have been more detailed and more thought about.

A->B: Because of our lack of consistency with effort estimation, we felt like it wasn't helpful to our project. However, in future by being more consistent with this we would have a better understanding of our own abilities and what we would be able to achieve. As well as a better understanding of each other's abilities and in this way we could have broken down our user stories more effectively. At times focusing too much on implementation and not enough on scrum and the trello board meant that we did not use it to its full potential. Next time this could be improved by properly referring to our board throughout the sprint.

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

A: Our acceptance tests were written quite late in the project. Our application is really easy to use. There is only one way to use it so writing the tests seemed useless. We still created them since it is part of the exercise but we tested the application without even realising it every time we opened it.

B: We should have written the tests as soon as possible since different team members might have had a different version of the application on their device (all of them working but with a different look for example.). Those tests should have been realised by team members who had no part in the development of these functionalities.

A->B: We feel like it is once again important to sometimes slow down to take some time and look back to make sure the product we realise is of great quality instead of rushing ahead because of stress and the want to implement more features. We noticed the technical debt of this decision as some teammates were able to get certain features to work (such as user location) while on other devices it did not work. In the future this could be improved by implementing tests for every task.

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

A: We created three KPIs: evaluation of stress levels, estimation of the percentage of tasks achieved based on what the member was supposed to do during the week and estimation of efficiency. Those KPIs were great conversation starters to understand the struggles some teammates were going through : technical, personnal, lack of knowledge.

B: We realised that maybe we should have used our KPIs more often, two times a week for exemple. Indeed, we would realise at the end of the week that a team member was too stressed to focus. Realising this earlier could have meant helping this person with their

tasks. Moreover we did create a KPI for stress but we did not differentiate stress from the project, for personal reasons or other school projects. It sometimes made it difficult to understand where the problem came from thus making it difficult to know if we could help the person. We also should have created a KPI that took into account the time spent on research or technical difficulties since some team members spend a lot of time on those elements which was not described by any of the KPIs.

A->B: In the future, we may have changed or adjusted our KPIs after realizing that they were not descriptive enough to represent the entire situation. We realised our KPIs were not perfect thanks to the openness of some team members. So good communication is key and in the future this would be focused on even more.

Social contract and effort

- your social contract, 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)

A: We barely changed our social contract. We were very thorough during our first draft as a few of us have worked in teams many times and were aware of what to watch out for in a team. The whole team followed the social contract closely which really helped with the teamwork.

B: We initially had problems forming a team which meant that one of our team members was not present when we wrote the social contract. It would have been much better to write it all together and may have helped the team if everything was talked about all together. In the future, it would have been good to add some kind of additional point about clarity on what everyone was working on, as well as having a clear understanding of what every team member understands and does not understand about the code. Due to working from home, it is hard to know if everyone is on the same page, and the social contract could have reflected that better.

A->B: By updating the social contract more frequently to reflect the changes of the team's dynamic. As well as ensuring that the social contract reflects the features of working from home such as not having a clear understanding of what every team mate is doing, and not being able to view each other's screens to know what works and what doesn't. All this would have helped in openness and communication.

- 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)

Every team member spent a large amount of time on researching, installing, testing, and simply understanding the technology we were using. Now that did not always reflect in what and how much we were able to deliver to the customer and in terms of the value the final product has.

B: We should have kept a better track of how many hours we spent on the project. This could have also contributed to our ability to better estimate effort for each task.

A->B: In the future, the time spent on each task could be added to the trello board. As well as more openly discussed during the sprint reviews. This would also help in doing better effort estimation for the upcoming sprints as well as better understanding how each team member works.

Design decisions and product structure

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

A: We used an API for the map that uses Google maps in order to make the app as easy as possible to use since most people have already used this map. Google maps also had a Places API which allowed us to implement a search feature. We used MySQL as the database which was not hosted on an external server but was locally installed on our own computers. We created a restful API in order to get the bathroom locations from the database to appear on the map. We did not however intentionally implement any kind of architecture pattern as our main focus was simply getting some kind of feature to work.

B: We would like to have followed some kind of architecture pattern which would have helped give our project structure. Ideally our database would have been hosted somewhere. Now that it is just a locally installed file, every change made in one person's database has to be changed everywhere (which also means good communication is key).

A->B: Now with a better understanding of React Native, we would begin the project by discussing potential architecture patterns. This would have helped give our programming more structure and direction already from the first sprint. Next time it would have been good to host the database somewhere instead of having a shared file. Having to change from a local database to a hosted one means that we have to make many changes to our code. For what we implemented so far, these issues did not affect the customer value so greatly. However if we wanted to go further and implement possible user changes to the information in the database, our database would have to be hosted somewhere. Otherwise a user would only be able to see their own changes to the database and not the changes made by another user.

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

A: We did not use any technical documentation. This was because our implementation of both the front end and the back end are quite simplistic and we did not feel it necessary to express with any documentation.

B: We should have kept better documentation on what dependencies and libraries we installed, that would have made delivering the product to the customer easier. While documentation of the code is not essential at the moment, if we were to implement more features to the product, implementing some kind of documentation would be good to communicate how everything works.

A->B: We did not even realise how many dependencies we would have by the end of the project, so keeping better track of this is essential in future projects.

- how you use and update your documentation throughout the sprints

A: We did not have any documentation so we did not update it. It was also not something we really thought about until the final few sprints. We did however update the team on the new elements to install and how to install them through discord.

B: We would have liked to have better documentation, and records of the dependencies to install, what new classes have been created and such. Next time we would like to have already implemented some kind of system of documentation already from the first sprint.

A->B: We were so concerned with having something implemented and delivered that we did not take the time to document what was being done. It also was difficult for us to balance ensuring good agile practices, actually developing our application and writing technical

documentation. With our focus on just having some kind of features implemented, our final product is quite surface level. In the future we would like to have maybe spent more time pre-development to set up a good system of documentation which would have ensured that we updated it throughout the sprints.

- how you ensure code quality and enforce coding standards

A: Because of our lack of knowledge on programming it was very difficult for us to apply some set of rules to have a code of great quality. We tried our best to keep the code as clean as possible by getting rid of all useless elements at the end of the project and by keeping only some comments to explain the functions of the difficult parts of the code. We researched some code rules and some team members have an understanding of how the code should be organised but it seemed too difficult to apply those rules since we would have to modify our code a lot.

B: We want our code to be of good quality which would make it easier to read, understand, and to build upon.

A->B: In the future we would focus on code quality already from the first sprint by researching coding standards, looking up examples of other repositories to see what these standards are. Writing code of poor quality that we then have to make huge adjustments to later is unfeasible and a waste of effort. One again technical debt caught up to us.

Application of Scrum

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

A: We made sure that every team had the role of ScrumMaster and Product owner at least once during the project. However we felt like some team members were uncomfortable with those roles which made the communication a bit more difficult during those weeks. We feel like our organisation was great overall.

B: Some team members had a difficult time understanding what those roles were supposed to do for the team so we should have spent more time explaining the roles or the members should have spent more time researching what their role was about. We felt at the time that trying a role is the best way to learn but it probably was too stressful for some and they were just overwhelmed. The team would have helped them more or the role should have been given to someone else during the week with everyone's approval.

A->B: We feel like the KPIs and team building could have helped us understand the situation of some team members and helped them better. Communicating about your struggle is also important in a situation like this but can be difficult when you feel like you have a role of "team leader". It can also be difficult to bring the subject to the table when you are the one who suffers from it.

- the agile practices you have used and their impact on your work

A: We have tried to use as many practices as possible. We focused on breaking down tasks with the help of the INVEST criteria. We kept each other updated by texts or meeting at least three times a week following the examples of "dailys". We always had a Scrum master who led the meetings and ensured everyone had a task. We filled our KPIs and discussed the outcome of it and kept our Trello board as organised as possible. We had sprint reviews where we would discuss the deliverables, what was good about the sprint and what could be improved. We struggled with implementing acceptance criteria for all tasks, as well as effort estimation.

B: We should have had a clear checklist of every element that had to be done every week. We were overwhelmed with all the elements that had to be done outside of the realisation of the project, and a checklist could have helped.

A->B: The use of all the agile practices comes with practice and trial and error. We used many practices considering most people in the team had never used Scrum.

- 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: The person playing the role of Product Owner changed every week. The person tried as best as possible to act as if she had not contributed to the project. It helped us focus on only the value we would add to the product. We modified some parts of our project after those meetings.

B: It would have been great to have a product owner give more constructive criticism and really take on the role of representing the customer's viewpoint more objectively. Our lack of programming knowledge made it hard for the product owner to really demand adjustments to our features or implementations as that person knew we struggled already with the implementation. Their demands couldn't be so high.

A->B: While we wanted every team member to have the chance to play the role of the product owner, it may have helped to only have one person playing this role. This would have allowed the person to better understand the mindset and perspective of the customer.

- 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: We feel like the best way to learn about most of the tools is to simply try them and get better through trial and error. For the Scrum board, one of the team members at the beginning of the project was part of a Scrum team and was able to create the board with us and explain it. We learned a lot through this and it gave us a good starting point to use it correctly. We then realised through supervision meetings that some elements were missing in the scrum board like the effort estimation and the acceptance criteria.

B: We feel like we should have learned better how to use some of the tools like GitHub by following a course on the internet before completely diving in the project.

A->B: Because the development environment was new to all of us, we think it would have been good to have more meetings where we discussed the new information that we had learned through our research. In the future this would have been a great thing as it would mean we are able to learn more from each other.

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

We feel like we learned much more by doing the project than during the lectures. Since there were so many new elements the lectures could be a bit confusing. However by using the elements in our project it was much more understandable and easier to remember. The guest lecture was interesting but we learned more when actually trying to implement and use agile practices.