

Week 1 Reflection

I want to learn and have a good understanding of scrum, sprints, and other important concepts of agile development. During the exercise session on Wednesday, it felt hard to both concentrate on keeping to the allotted time, as well as making sure we really focused on each step, the planning, execution and reflection. For example, instead of truly reflecting on the sprints, we focused more on planning the next sprint. Instead of the 3 sprints feeling like an iterative process, it felt a bit more like 1 long extended sprint. I am also really interested in seeing how using this style of development affects and benefits the software that I create.

I am excited about working with the team to create something interesting and creative. I hope to be able to contribute to the group as much and as best I can. I want to be able to create an environment in our group where everyone feels like their voices can be heard and they don't feel afraid to make suggestions. Group dynamic can really affect how well a project is carried out and so I hope to do my part in making it as successful as possible.

Week 2 Reflection

Week 2 was really about establishing ourselves as a team. We struggled a lot with members both leaving and joining our group which felt a bit stressful to not know how many members we would end up with, and ultimately how much we would be able to accomplish with the people that did choose to stay in the course. Despite those problems we were able to start on our project. I think one thing that I found interesting was the focus on the value that our product is giving the customer. It feels obvious but it is new to me to think of the value proposition in every step of the development process.

We also started creating out project backlog and I had initially thought the vertical slicing would be a lot more challenging as it is definitely a new way of thinking for me. In most of my other programming classes I have thought only in terms of methods, functions, and the entire code being like layers. However this understanding of vertical slices actually came quite naturally to our team and we managed to create user stories that give value in themselves. While each slice builds on the functionality of the other, each slice has value in itself.

For next week my focus is trying to continue to create a good structure in our sprints, and in the team in general. As well as of course start with the coding of our project.

Week 3 Reflection

Week 3 went a lot less smoothly than we had originally planned and maybe hoped for. We had divided the project into about 4 epics that we felt covered the entirety of the functionality that we would like to implement in our app. We decided to focus on the user stories that related to the 1st epic which was about creating a basic foundation of the app, implementing the map, creating a database and connecting the database to the frontend. In the end we only ended up finishing less than a half of the tasks we had originally planned to finish. We overestimated how long “setting up” would take as well as how long it would take us to begin to understand all the tools that we had chosen to use (in this case javascript, react native, Expo, MySQL). At the beginning of the sprint we also did not spend enough time with poker planning and really evaluating the amount of time/effort each task would take. We had a loose understanding of “oh maybe this task will take the more time than this task”. In the next sprint we hope to spend more time with poker planning. We will also try to estimate the effort each task will take not based on our wishful thinking but more based in actual reality. By that I mean the amount of knowledge we have on the topic (how much will we have to research) and the amount of time we have in the week to work on the project.

I was also the scrum master this week which gave me the responsibility of leading the sprint and ensuring good communication and teamwork in the group. For a first time I think I did an alright job however I think there are definitely areas of improvement. I think I had to remind myself that it doesn't matter if just I have managed to finish the tasks that I was supposed to finish, if the rest of the team is stuck on something or needs help. I think I should also have established more and better communication throughout the week and not just on the days we decided to have a meeting. I think the team would definitely benefit from small, daily updates so we know where everyone is at with their tasks. If someone in the group is struggling with a task, it is better to know as quickly as possible and not wait till the sprint is almost finished.

Ultimately for the next sprint I feel it is important to do a better job estimating our own capabilities as well as establishing better communication amongst the group.

Week 4 Reflection

In week 4 our sprint went a lot better than in week 3. We made lots of improvements to the structure and format of our sprint starting with the first meeting we had at the beginning of the week. In our discussion with the supervisor we gained a better understanding of the meaning of value. It was very helpful in understanding that even though we may think we have not done enough or progressed enough, the tasks we have completed can be valuable depending on how they are sliced. If you slice tasks strategically each one can give value to the product owner and ultimately the customer as well. We realised that in order for our tasks to deliver more value we would have to rewrite them and give them a greater focus on the value that they deliver to the project once they are completed.

This definitely helped in our sprint going smoother and ultimately being more effective. We divided the tasks even more. For example, we changed the task “get markers to show up on map”, we sliced it further, “have one hard-coded marker appear on the map” and then “have one marker appear on the map from information from the database” and then “have multiple markers appear on the map from information in the database”.

For myself thinking in terms of incremental value has had a really great impact in my own understanding of agile. In my previous experience of programming and creating projects it has just been about trying to get something done or something to work, a kind of brute-force programming style. However this format of agile development allows for a lot less stressful programming style. As a fairly new programmer, creating something like this project felt extremely over-whelming but it has really helped being able to focus on one small slice at a time whilst also being able to see a progression. For example if we had just spent 3 weeks focusing on the database we would not have really had anything to show for it. Our product may have some kind of value to us but not to our customer.

This week we managed to accomplish many more of our tasks. We were also better at communicating and updating each other on the progress throughout the week. We also did the poker planning which helped in our ability to plan effectively. For the next week I hope to continue to progress, and follow the sprint structure even better.

Week 5 Reflection

In sprint 5, we focused on finishing all the user stories that were connected to epic one. This week I was the product owner which meant I had to keep more of an eye on the trello board, the user stories, and on the value each of our tasks was going to eventually give the customer. Playing at the product owner gave me the opportunity to take a step back and view the project from a different perspective, from the customer but also from the eventual users of the product. This week we accomplished quite a lot in terms of value. After the first sprint, the team realised that our core functionality (and the main thing we wanted to ensure we were able to implement) was for our bathroom markers from our database to appear on the map in the app. In this sprint, we were finally able to implement this functionality in our project. Now instead of having hard-coded marker locations showing on the map, all our markers come from the coordinates that are stored in our database. As a product owner I was very happy with this development, as this means the project is one step closer to being able to implement other information that pertains to each bathroom location, which is also stored in the database. It also means that it will be simpler to update the information stored in the database, no longer having to deal with the source code in order to change the marker data.

We didn't have as much or as frequent communication this week since our schedules were more varied. However we did still update each other on what we were doing, writing on our discord and sending screenshots of updates in the app which still allowed for some idea of what everyone in the group was working on. This week we also slightly overestimated the effort needed for each task which meant that we ran out of tasks to do and could have probably assigned more tasks during the beginning of the sprint. I think for myself this is one of the hardest things about the sprint structure. It feels hard to get the effort estimation correct. You have to know everyone in the team pretty well to be able to understand how each person works and how much they will be able to get down as well as understanding yourself. Each week has been a bit of a surprise in terms of what tasks took more or less time. However I do notice improvements in our effort estimation skills. I definitely see how you can get really spot on with the effort estimation and knowing how many tasks the team can potentially accomplish during the sprint.

Week 6 Reflection

During Sprint 6, my main task was to make each marker clickable so that when a user clicks on one of the markers a window will popup giving them all the detailed information of that specific bathroom. We started the sprint by splitting this big tasks into smaller tasks, which fulfilled the 'small' characteristic that makes up one of the INVEST criteria. For example, one tasks was the simply click on one marker and get an empty window to popup. The next task was to click on one marker and get some kind of information to appear on the window. The next task was to click on any of the markers on the map and have a window popup with more information pertaining to that marker. I feel these tasks where sliced in such a way that they still gave value to the customer, although the 2nd tasks of course gives less value than the customer being able to click on any of the markers and learn more about that location. However, we did notice that these tasks couldn't exactly be seen as completely independent as they did build on each other. Through the various sprints over the past few weeks I have learnt that their is kind of a balancing act to creating valuable tasks that are also independent enough from each other so that they can be worked on simultaneously. We did struggle a bit during this sprint in knowing how to best divide the tasks seeing as how they were often built on the other tasks.

We also ran into some trouble due to the lack of knowledge with these programs and programming language. For example, we would have loved to create smaller popup windows where the map was still visible behind it, instead of windows that take up the entire screen and obscure the map. However we were struggling to find a solution to the problem and decided it's better to just do it the simpler way considering how that would still add value to our project, and give value to the customer. I consulted the product owner and discussed with them that for this weeks sprint it was more productive to just complete the task doing it in the way that was less desirable. We discussed that in the next sprint this could be improved upon instead of just moving on to add more functionality. I think this sprint showed that it is not just about how much functionality your app or program has, it also has to be well made, well coded, well tested. So if we have to spend another sprint improving on certain aspects to what we have already done, that still creates value. If the customer gets a better made product with less features, it is better than a product with lots of features but none of them work properly. In a way I forgot that this is the whole point of agile development. The developers are supposed to be able to make changes and fix the things that they have already done if the product owner or the customer is not satisfied. Because the development style is so incremental, I do see how easy it is to make these changes, we don't have to go back all the way to the beginning to change how the marker descriptions will look to the user. Actually what we did this sprint, although not ideal, makes it easier to create windows that only popup on the map.

Week 7 Reflection

The teams focus during this week's sprint was less on adding new functionalities to the app, instead we focused on ensuring that our project was deliverable to our customer. This for example includes adding a package manager to resolve all the dependencies as well as fix the bug we were having with localhost which meant we had to change the IP address in the code whenever we connected to a different network. So in terms of value for the user, not much new and fun functionality was added. However there is still value in what we did because if software is not scalable, meaning it cannot be launched for multiple users to have access to and use then there is not really any point in creating that software. The project can't just be useable on the devices that we have in the team, it must be useable to everyone.

With this shift in focus, we came realise a lot of technical debt that we have acquired throughout the weeks of development. With the stress of the sprints and the want to just have something implemented and be able to deliver something, it has meant that at times we took some short cuts in order to finish that week's tasks. For example, the issue we were having with the IP and connecting to the API. If we had initially spent more time trying to resolve that bug throughout instead of simply putting "tape" on the situation, then we would not have had to come back to it and now at the end of our project sort that out. We noticed this problem with the database too. Our database is not hosted anywhere, instead we have simply exported and the imported it to our own devices. However of course this is not ideal when delivering the final product. If we had sorted this out originally when first implementing and creating the database, then we may have saved some time. Now we may have to go into our actual code and change the connection between the database and the API.