

Final Report

Team Abacus

27 October 2021

Q	A	B	A -> B
1	<p>We chose to make a complete android application which focused on displaying values for air quality in Gothenburg. The priority was weighed and the number one was to get an app working and after that display the values on a map.</p> <p>The project was divided into four epics where the former two priorities were the first and second. Another one for statistics and also one for a user-friendly user interface was added. A definition of done was set up for our own sake telling us when we were done with our user stories. Every week we split user stories into Todos that we weighted and assigned. We made sure each week that every todo's card included some kind of end value. If accomplished this would create value for everything we did in the project towards the product owner. Some minor cards were very internal and were left out of this idea though.</p> <p>No particular acceptance tests were made, we simply decided that code should be tested by the coder and later on inspected and/or tested by another one in our group, this was made as points in our DoD.</p> <p>Our three KPIs consisted of:</p> <ul style="list-style-type: none">- Firstly the combined "work" on the week's finished To-Do cards, based on their assigned weights (XS-XL).- Secondly we measured how many User Stories were completed for each week to provide more of an overview picture of how the project was moving along.- As the third KPI we compared the amount of finished work to the work left on the To-Do's that were taken on for the week but not yet finished, as a way of estimating how well we balanced the	<p>Obviously we would have started using all of our KPIS earlier, they made every sprint so much more clear. They worked really well for us so probably would have wanted the same ones again just implemented them directly next time.</p> <p>Perhaps in testing we should have had some specifications of what is tested and not to make this part clearer. To have proper testings seems a bit overkill for a project of this size.</p> <p>Next time our user-stories should be smaller. We had some stories in the beginning that we later figured were too large for any sprint. Also we should have sliced them better.</p>	<p>With our current knowledge of Agile we would have set up the KPIs right at the start before any sprint to get this going as soon as possible. Also next time when initially weighing a card we should let the person to perform the task weigh it since a weight can be personal meaning various people can have unequal knowledge thus someone might need more time to perform the same amount of work.</p> <p>Instead of just having "Should have completed testing" in our DoD we would have specified some simple statements of what has been ensured via the tests. An example could be as easy as "Code does what it is supposed to" or "Has been stress tested with inaccurate parameters".</p> <p>Again with our current knowledge and experience we would from the beginning make sure to now compress too much into single stories but instead divide them up and instead focus on creating value for both</p>

	amount of work we accepted for each week. This was the last KPI we settled on and started using for week four.		product owner and us as a team.
2	<p>Social Contract In the first week we established our social contract and set our running schedule with a longer meeting on Mondays for ending the previous week and starting the next one. In the beginning we had trouble containing our shorter meetings to 15 minutes (Wednesdays and Fridays) which was our stated goal for the Scrums and instead would spend over an hour on meetings. In week 4 we switched to summarizing the week on the Friday meeting instead, prolonging it and reducing the meeting time on Monday.</p> <p>Effort In the beginning progress was slow since everyone was new to both Kotlin and Android Studio. Most time was spent on learning and researching how an app like our idea could be made. It was also hard to split up the tasks in smaller parts since we didn't have a big base project to split up yet. Later on this was much easier because some people could work on the statistics tab and some on the map tab for example. So the first week many people did the same task but this wasn't all bad since we needed the learning. At week 4 we started to use more KPIs to measure how much effort we estimate to take on for the sprint. We give all tasks an effort rating of either XS: 0.25, S: 0.5, M: 1, L: 1.5, XL: 2. Where a number of 1 is the amount of work 1 person is expected to be able to do. This made it easier to plan how much work we expected to be able to do in a sprint but it was hard in the beginning to know what effort rating a task should have.</p>	<p>Social Contract We often didn't keep our meetings to the decided lengths, especially in the beginning.</p> <p>Due to summing up the week on Mondays we got into a desync of sorts towards our supervisor meetings with Felix Naredi which were also on Mondays - The supervisor would check our project Trello on Sunday expecting the week to be over, and feedback would not be as good as it could have been. In week 4 we decided to start ending the week on Fridays instead, which moved some time from Monday's long meeting to Friday's previously short meeting. We often ran longer on Fridays to summarize the week anyway, and only moved Trello cards to finally summarize the week on Mondays.</p> <p>Effort We should have used KPI's earlier to better know how much effort was done every sprint. It would also have been useful to keep track of how much time was spent working in a sprint to better know how much time a medium task takes for example.</p> <p>Better use of teamwork from the start to get the project up as soon as possible.</p>	<p>Social Contract It would be a good idea to have set proper meeting times to stick to and plan their day. At the same time perhaps right at the start of a project there could be issues that need discussing which takes longer time than planned (although perhaps 4-6x as long meetings than planned is perhaps not ideal).</p> <p>As for the weekly schedule overall, looking back it would have been a better idea to spend more time discussing and coordinating the schedule with the supervisor so supervisor guidance fits well with the weekly planning.</p> <p>Effort From what we know now we should have implemented more use of scrum from the start. This would have helped the effort estimation.</p> <p>We also did the same thing in our own project in the beginning, this was helpful for learning but maybe it could have been more efficient to work together to get a project up and running. And also work together to split it up in different parts so we then can distribute tasks more in parallel.</p> <p>Next time it could be</p>

			<p>helpful to track the amount of time spent each sprint to know how much time a medium task takes for example. That would also help to better know the individual workload. One person might be able to do a large task in a sprint while another only can do a medium task in the same amount of time.</p>
3	<p>First the project started without a design plan. We decided that all public functions should have coherent documentation for all team members to easier understand each other's code. We also chose which API to use.</p> <p>Later we researched for a visual prototype of our app and the backend structure (what happens when we press which button etc). We began with a menu structure with 3 tabs. We also implemented our definition of done to among other things satisfy the cards criteria and ensure code quality by code reviews and documented code.</p> <p>Then we implemented a dropdown structure for the first tab, which was functional and fetched the data chosen by the user. Updated DoD requirements.</p> <p>Following a chart was implemented to the statistics tab and a button which changes the chart to display week values.</p> <p>We also implemented google maps to the map tab, it is automatically zoomed in on Gothenburg at app start and displays our chosen air stations.</p> <p>Afterwards we changed the design in startTab to look more user friendly, also implemented a customized spinner which looked more coherent to the app compared to the previous one. Also an app icon and splash screen was implemented when starting the app. to the mapTab we added several extra effects to the map marker such as sensor value dependent marker color</p>	<p>We should have enforced the code commenting criteria or skipped it completely. The team was bad at remembering to comment on the code and as a result we instead worked together more which worked great in the end.</p> <p>The database API we used for air quality values apparently crashed and stopped being updated in the middle of the project course which led us to have to use old numbers from the last update (from 17 September). We have emailed several times and the administrators of the API at Göteborgs Stad said it would be fixed, but as of writing this report the API is still not updating. Our supervisor Felix assured us that this would not fall on us, although it is of course frustrating for the team.</p> <p>UI didn't scale correctly for different phones, where at first the buttons weren't centered, but also that parts of the graph didn't show on smaller phones.</p>	<p>For this project, the team was divided into different menu groups (start, maps and statistics) and had very limited contact with each others code. Although most of the menu groups used some form of contact with the API fetching classes, therefore it would have been more necessary with commented code for those classes.</p> <p>The free APIs available for air quality were few, we should have considered paying for an API and split the cost between team mates so we wouldn't be stuck with an API without any accountability in case it goes down. Or have chosen a different measuring value such as pollen instead of air quality.</p> <p>The UI problems happened because we tested the app on mostly the same phone. We had that the buttons started at x from the left on the screen and had y distance between</p>

	<p>and displaying stats when clicked on and user location to the map.</p> <p>Then there were many changes to statisticsTab, the code was split up into helper methods since many of them were just code reused and or code which would not do anything. These changes made the app run faster. A callback was also implemented, which receives updates from the API.</p> <p>Finally we changed the app color theme, from dark purple to a light blue which better suited the aesthetics. Also xml code was improved to look better and more coherent, as well as code documentation which was added to all classes.</p>		<p>them. W. This is a problem as screens have different sizes. We solved this by adding another constraint that centered the buttons, but there still is the problem that the graph isn't always visible on smaller phones. We should have started earlier to test on different phones to see how it looks on them. Then fix that the graph and other things adjust in size based on the screen and not have a fixed size. We noticed the problem with the graph a bit too late, but this is something that has to work perfectly if you are going to release the app.</p>
4	<p>The first week we didn't have any Scrum master, secretary, or Product owner. Also no estimation of the workload for different tasks. The second week we used a randomizer to assign the different roles and started using t-shirt sizes for todos, to better estimate how much work we took on for the week. In week three we worked on keeping two of our meetings shorter and had a longer meeting on Mondays. In week four we improved our usage of the Scrum board and got better at converting t-shirt sizes to points, to measure how much work we took on for the sprint. In week four we also started working more together as a team and had more pair coding than before. Week five we also worked a lot together and updated the Trello board as we got done with tasks. Some of the t-shirt sizes didn't get the correct size so we adjusted them on Friday. After a suggestion from our supervisor, we changed our meetings a bit. Instead of doing the weekly review and planning the sprint on Mondays, we moved the weekly review to Fridays. The last week we worked together to fix the remaining things and some bugs that we noticed and prepared for a presentation on</p>	<p>Assigning roles for Scrum master, secretary, or Product owner at the start. A more active Scrum master and Product owner. Either follow the DoD better or change it as we did not comment on our methods before moving them to done. A better understanding of how much work we are able to do per week. Better understanding how much work it takes for different todos. Problems with Github and updating to the newest versions of the project.</p>	<p>It would help to spend more time the first week to think more about how to improve our implementation of Scrum. Starting with assigning roles, where the Scrum master is more active and makes sure that things are getting done. The product owner should also be more active and talk more about how he/she wants the user stories to look like, instead of it being more shared in the group, as we started with. In the DoD it said that we should comment on every method, but as we were new to Kotlin, we ended up trying a lot of things that we later removed. This led to that in the UI part we didn't comment until after week 6 and didn't follow the DoD. We think the best thing</p>

	Wednesday.		<p>would be to update the DoD to only comment methods that are going to stay. This was done a lot better on the API part which the ones working on the UI had to look at, to implement their part. We were all new to Kotlin and Android Studio, so that made it harder to know how much time different things took. This got better when we got more experience. Measuring t-shirt sizes and converting to points was very good for getting an understanding of the workload and that is something we should have done from the beginning. However, we've realized that also logging the hours could have helped a lot as a KPI to better connect how well we estimated our shirt sizes - finishing extra "points" in a week is perhaps to be expected if we put in extra time. With Github, it would be good if we had a better understanding of how to use it in Android Studio and how to discard your changes and update the project. There was a problem for one person in the last weeks where he didn't get the last code for some files and instead sent it over Discord for upload. It would have been better to communicate more openly about this as it worked after getting help with reset head.</p>
--	------------	--	--