Sprint 3 Retrospective - Team 14

Introduction

Our team developed a COVID-19 tracker web app using the React framework. The app allows us to keep track of a patient's status in terms of infections, what were their previous symptoms, and who was the doctor that they are/were assigned to? We worked so far on the admin part of the app, which allowed us to modify not only the patient's data but also the data of the different health professionals.

Based on our progress from Sprint 2, we were able to get more progress done. In Sprint 2, we implemented the general outline of the client application. This included the creation of the client dashboard, a simple profile page, as well as a symptoms page. In this sprint, we expanded more functions in the client application. We added integration with the backend for some functions such as editing the user profile, as well showing a person chat history. In terms of only the frontend, we added display for the diary, and the update status pages as well.

In terms of project management, it was very similar as to how we functioned in Sprint 1we divided the tasks using GitHub issues and rated them based on importance and difficulty using the Kanban Board. Our team also followed the agile methodology, where we had two weekly meetings to discuss our progress and assign teammates to different tasks, however, for this sprint there was some hiccups to that strategy.

What went wrong

1 – Lack of commenting on the code early

During the sprint, we had a problem where we did not comment our code regularly as we should have. This sometimes created a bit confusion amongst the team where we could look at another team member’s code and would have to message them to understand it.

This had no impact at all as we fixed this as soon as we did our first team meeting.

2 – Struggle to adhere to consistent agile methodology

During the sprint, a lot of team members had many other exams and assignments for other courses. This meant that many of them could not attend the two weekly meetings that we had amongst ourselves to discuss our progress of the project. Another reason for this was also that the sprint coincided during our one-week break, and a lot of the team was studying/relaxing during that time.

Overall, while this did have an impact on the overall scrum meetings, and to see what the overall progress of the team was, we did have a written description of what we did, and we will be doing during the meetings in our team server.

3 – Application appearing different on different machines

During the sprint, sometimes the application when we would run it, especially the client application looked different for different people. The code would be similar for everyone since we mostly dealt with pull requests and needed approval before making big changes, despite that however, the layout sometimes and some functions would be different for some people. To fix this, we usually ended up having a call with other members of the team and went through step-by-step to solve the problems.

Overall, this could have had a large impact on the project, but the issue was still solved due to teamwork we displayed.

What went right

1 – Separate branches to reduce merge conflicts

After sprint 2, we knew that we would be dealing a lot more with the backend in sprint 3. Because of this, we decided after the last sprint, that it would be good idea to have a develop branch, where we could create other different branches from, and then later create pull requests to merge the code in. This was easier for the team to handle, instead of always merging pull requests to the main branch and then merging it to their specific branch.

Overall, this had a big positive aspect on the workflow since nobody had to wait until something was merged from the main branch to continue working on their own branch.

2 – Everyone doing more on the project

In sprint 2, we only assigned one tester to write most of the tests. This meant one teammate had to write many of the tests, which also meant that he had to understand how the other code worked to do it. While this was still not a bad thing, as we were able to do the tasks very well for the last sprint, in this sprint, everyone was able to write tests based on their own code, a code they would understand, and could hence write better tests for it.

Overall, this also had a good impact on the team since we were able to learn more stuff to do, which at the end of the day is the goal of this project.

3 – Team communication

As in the other sprints, our team communication was again on point for this sprint. Multiple times in the sprint, we encountered bugs on the application which one teammate could not figure out how to solve. This prompted someone else from the team who could fix the issue to step in, and either call the person, or even screen share to help figure out the issue at hand. Another example of this would the implementation of the backend. As most of the team are more comfortable with the backend, there was always bound to be some hiccups transitioning to that aspect, but once again, if someone was stuck in the team, another teammate who could help always stepped in to help the person in need.

Overall, this had a very good impact on the project, as it did not derail anyone in the team, as well as letting us work on our tasks in a timely manner, which allowed to finish the sprint stories in a timely manner as well.

Conclusion

As like the previous sprints, we found out of how important it is to document stuff as we go and not have to scramble after to make everything work properly. While we did do a great job of communicating as team, some of it, such as multiple midterms from team members, made some aspects out of control. We did also learn, however, that despite these small problems our team was able to persevere and still do the tasks on time, and it’s an important confidence boost and something we will taking with us in Sprint 4 and beyond