

Sprint retrospectives

Sprint 1

We were overly ambitious in our estimation of how much we would get done during the first sprint. One important factor was the half term exam in mathematics, which we thought we had accounted for, but still ended up with stories that remained in progress by the end of the sprint.

We experienced issues with distributing work between team members. Partly in one user story that was way too big, and partly because we were sloppy in ensuring each team member had a clear understanding of what their assigned tasks entailed.

No tests were written during the first sprint. Thus we had to manually test each story and hope that we did not miss any bugs while doing so.

Sprint 2

In the second sprint, we added a first task for each story for writing tests, before getting to work on implementing. However, since the majority of our code interacts with Firebase and Facebook, we came to the conclusion that it would be too time-consuming to write automated tests for these dependencies. We still tried to test all local functions, but in the end, we felt that the purpose of test driven design was lost.

Prior to the second sprint, we took some time to reevaluate some of the larger user stories, and divided them into smaller ones, which really has paid off. It is much easier to assign work this time around, so we believe we have made substantial progress. On the other hand, we still need to keep reevaluating stories as the work progresses as we gain more insight.

We were overly cautious with grabbing too many stories, which led to almost all stories being finished halfway through the sprint. Team members then haphazardly added more user stories from the backlog to be able to keep working, without the group discussing which stories should be brought in.

Sprint 3

We got some valuable feedback from the interaction designers which made us change to priority of a few user stories. Since we only got one week to go we had to remove some of our stories and make sure to get a working product ready before the demo.

Test driven development was abandoned. It took way too much time and felt limiting. Since most of our app depends on multi-user interaction and needs to communicate to a database

it was also difficult to get things working properly. We decided to write unit tests whenever possible instead.

We did a full rewrite of the interfaces for interacting with the database. This took a lot of effort but now the database specific code is much easier to interact with.

Too much time were put into the refactoring of the code, less time on real stakeholder value. At the same time we had to refactor certain parts of the application to get things working properly.

Sprint 4

Looking back at the sprint we agree that all would have benefited from following the Scrum workflow better. Got into the mindset of “just getting things done” instead of planning and communicating properly.

We didn't write any tests since we focused on bug fixes and getting everything working properly before the upcoming demo.

Work environment haven't been the best, it was difficult to find available group rooms.