Sprint 3: A Look Back

Introduction

Welcome to our Sprint 3 retrospective. This sprint marked our halfway point into the project. This is an important phase for our developers as we are in the thick of the programming aspect of the project. Many tasks were focused on back-end and user acceptance tests. The challenge of this sprint was finishing the many issues assigned, as this can affect the stress and pace of the second half of the sprints. As we gather to review this phase, let's take a measured look at the achievements, obstacles, and insights gained during this phase of our project development. The following is an overview of the important components of our project.

Version Control: GitHub for collaborative development and version control.
IDE: Visual Studio Code (VS Code) for code development.
Diagram Creation: Draw.io for creating diagrams.
Prototyping: Figma for designing and prototyping user interfaces.
Testing: Cypress for end-to-end testing, along with Angular's spec tests for unit tests using the Jasmine/Karma testing framework.
Continuous Integration/Continuous Deployment (CI/CD): Utilized GitHub Actions for automated CI/CD pipelines to ensure a smooth and efficient transition to the customer.
Architecture: Client-Server with Angular and Firebase.

What went wrong

1 - Ambitious Workload

For this sprint, I believe our team was too ambitious in the number of tasks that could be accomplished in such a short timeframe and many tasks were pushed to the backlog. To be more specific, our team pushed a lot of the user acceptance tests to sprint 4 because of lack of time or because the tests required back-end functionality that hasn’t been implemented yet.

2 - Merge Conflicts

For this sprint, we had a lot of changes and pull requests being made at around the same time and that caused a lot of merge conflicts which took a long time to solve. To resolve this issue, we had team meetings and carefully reviewed the code before merging and tested the code after merging.

3 - Code Management Document

It was unclear that we had to produce documentation for code management. This was communicated to our team later on in the sprint which was last minute. Thankfully, in our last sprint we had already gone over and collected information for most of the points to put in our PowerPoint. Yet, separating these sections was still difficult considering the lack of time while finishing up some remaining tasks.

What went right

1 - Good task distribution system

In sprint 2, we realized too many tasks were assigned at once on the same user story. This resulted in people having to wait on each other's work to commence the task since there are a lot of dependencies. We changed this up for sprint 3 and focused on front-end components first. We also focused on back-end components, but only on issues with the front-end already previously completed. This turned out much better as people had to wait less. Furthermore, it was less daunting having to complete a full user story in one sprint.

2 - Good team collaboration and support

Everyone on the team worked very well and completed their tasks to the best of their ability. We had several meetings to discuss our plan of action with regards to the code and frequently updated each other on our progress. Additionally, when someone needed help with the code, members of the team would gladly offer to help and provide valuable insight into how to solve issues/bugs.

Conclusion

As we conclude our retrospective on Sprint 3, we look back for important insights. Sprint 3 was an important milestone in our project as it determines how well the developers work with the current system in place. We had changed our tasks distribution system last sprint and it seemed to work a lot better in terms of productivity. For example, a lot more issues were finalized this sprint. We continued to be patient with each other’s mistakes in this sprint and held up with detailed communication. In all, Sprint 3 was a success for many reasons such as: good communication, good productivity, maintained good coding practices as well as good feature

outcomes. We look forward to building upon the work done during Sprint 3 and hope that the last 2 sprints allow us time to properly finalize and deploy our system.