

## **Project Reflection Essay**

Throughout this entire quarter of working in a group on a specific product, I have begun to notice that there existed many aspects that can either make or break teamwork and how those aspects can affect the progress of the product. One of those aspects that I have seen affect my product in a negative way was disorganization. Although my group, including myself, have been taught about the Agile approach to software engineering in order to avoid such disorganization, the approach was somewhat difficult to follow for us during the first Sprint when I was the scrum master. One of my earliest mistakes as the scrum master was not reigning everyone back whenever the discussions were turning into tangents. If anything, I also contributed to the tangents created, and as a result, time was wasted for the first week of the Sprint. Because time was wasted, important aspects outside of our user stories were not discussed, which resulted in disorganization and stalling for our product.

For example, because we only had created our user stories for the functionality of our app, we did not really focus on the UI aspect of our product during our Sprint planning and standup meetings. As a result, I did not really know what to do when I was doing one of my assigned tasks, which was to create a working camera for the app, because I did not know where to place the camera button. From this situation, I texted the group about the issue at hand, and had to wait until our next meeting until I can work on my tasks. Realizing that the issue was caused due to bad time management and bad organization in our meetings, I had made sure to correct myself as the scrum master and thought up of topics to talk about for the next group meeting. Through planning and organizing topics to discuss during the group meetings, tasks

became more clear for everyone and tangents were less of a thing. As a result, a basic expectation was created for future Sprints to meet; however, issues regarding our Agile practice did not stop at this point.

My team did not notice our next mistake until the end of our second Sprint where we noticed that our tasks were not being finished on time. The reason behind the situation was because the tasks that were assigned to each member were too broad, and the story points assigned to each of those tasks were not consistent at all. For example, a fairly easy task had the same amount of points as the task of setting up firebase for the product's database. As we all noticed the error, we decided to make a couple of changes, and one of them was to hold a weekly 2 hour long meeting reviewing each tasks for the Sprint and try to break them down into smaller and more specific tasks. Afterwards, we implemented our second change of playing planning poker on those smaller tasks that we had created from the larger and more broader tasks. With these two changes, more tasks were being completed compared to our previous two Sprints and the basis for the story points has been developed.

One last issue that I have seen was the existence of merge issues. Although this problem is not related to the Agile approach, it still had impacted Sprint 4 negatively. Because we had run into merge conflicts at the second week of Sprint 4, we had to spend the rest of the entire Sprint trying to figure out how to resolve merge conflicts. As a result, a couple of tasks had to be backlogged including the button to report. Additionally, bugs that existed in previous features

were found during the product evaluation as we had spent most of our time resolving the merge conflicts for the product release instead of focusing on testing our features and debugging them. Although we could have built unit tests during our third Sprint for the functionalities of our product, I believe that those tests would have taken a lot more time than anticipated as everyone would have to look up on how to build unit tests. With that amount of time spent on trying to write unit tests, we would probably run into the merge conflicts before we can start to debug the issues found from the tests. Despite the fact that the merge conflict had taken a toll on our time, it was still a good learning experience. Not only did I learn more about git and how to resolve merge conflicts, but because of that situation, I have also learned to expect unexpected situations like these to happen where the issue at hand was not caused by anyone, but rather by our lack of knowledge and experience.

With the changes implemented from the first and third Sprint, the product began to take its shape and progress began to go smoothly. Everyone had gone through a great amount of issues during the first couple of Sprints as we were all inexperienced with SCRUM and did not really know what we were doing. However, after noticing our mistakes, we all began to take notes of what to do for the next Sprint, and the one after that. Through resolving our mistakes by organizing our topics for our meetings, creating a communication link with everyone in the group, and holding longer meetings, I believe that we had not only become more experienced with the Agile approach, but also more knowledgeable in what to expect in the software engineering industry.