Selik Samai
10/22/2023
MS 501

MS 501 Post Mortem

To my surprise, this class has showcased the importance of Agile development both inside and outside of the game development space. Prior to this class I thought of agile as just a type of approach to software development but having gone through the course, I now realize that much of the principles behind agile are actually a necessity for a lot of software development in general. It's easy to overlook and minimize the tools that we use every day, but this course puts into perspective just how useful those tools are and why and where they are vital in the industry.

Though I've had some familiarity with scrum in my work as a software engineer, this course has formalized a lot of the concepts that I had used in my day-to-day work. In particular, I learned that agile has become extremely popular when it comes to video game development. This course also showcased why it's become so popular and why and where it's likely to increase. As a gamer myself who's current favorite game is Destiny 2 – a live service game – I see how the live service game industry is dependent on being able to rely on scrum and the development processes that come with it.

Finding a single thing that was most helpful from this course is difficult to pinpoint. I believe that having a greater understanding of the complexity of video game development would probably be what I consider to be the most helpful concept I gained from this course. I think for a lot of gamers, we're used to games coming out late and also filled with bugs but this course along with the readings, lectures and videos gives insight as to how this happens. A particular story that jumped out to me was that a lot of teams dedicate sprints to refining a game. Another insight gained is that there has been a shift in the dependence and importance of having too much documentation. In our Agile Game Development book, the author recounts a story of how a revision to their design document stated that the game needed some additional features. This was missed by the designers as they never looked at the changes to the documentation after their initial look through.

What was most confusing was the chapter on video game project management. Though I understood the notion that some games require a fixed ship date, and that date can't be moved, there seems to be a very large misunderstanding that the released game is not the actual final product which I would think is what would truly matter to investors. I think a good example would be what's happened with Cyberpunk 2077. It's very clear that the game should not have been released in its original form—given the backlash and its eventual removal from the PlayStation store, I think most people would agree. At present though, the game has had a huge resurgence after more development and bug fixes have been made. Still, it doesn't seem like these issues matter to investors. I get the sense that even if they knew the game would have to be pulled from the PlayStation store, investors still would not budge on the fixed ship date. It seems like there needs to be more exceptions to this rule. Luckily, it seems the live service video game model finds this a little more appealing.

This class will make me a better programmer in that I have more engagement with other stakeholders. In my current job, we have more ownership in the design, implementation and overall schedule for the majority of our work. I think this will make me better in that I'm not too accustomed to a certain structure. In working with the STEM for Kiddos as well as the class readings and lectures, I've learned to pivot my priorities when needed and also become better at time management and communication with the various teams. Most importantly though, I had to get information as to what the design approach should be for this project. This was the major difference for me as usually I have more of a say in how things turn out.

During my time as a member for STEM for Kiddos, there were definitely a lot of things that went right. First and foremost was integrating me into a team that was already in existence. The process was fairly seamless, and we found a role that needed someone's attention but was previously unassigned. Second was the coordination with the team, we used a wide variety of ways to communicate with our team. Lastly, I'd say the overall makeup of our team was something that went extremely right. Alexis was a great lead and was very communicative throughout the course.

There were things that did go wrong. My first task was to seek grants. I ran into a lot of roadblocks with this due to many organizations having a 501c3 requirement for receiving grants. I also ran into issues regarding advertising. My main priority was on grants and at the same time, finding ways to advertise for free was very difficult.

This course has been pivotal in giving me a greater understanding in how vast scrum and agile are. Previously I had a slight understanding and used it in practice on a day-to-day but after this course I now understand that much of the games that I love probably wouldn't exist or would be drastically different without it. Though never explicitly stated at my current job, I don't think I've had any work items that were larger than a given sprint. This course really helped in understanding how and why we break up large work items into smaller ones to ensure that every sprint releases with a working build. This is definitely crucial in iterative development. As well, having a clear understanding of what is considered done has been more formally explained. It doesn't necessarily mean the best implementation or even best version but it does mean it meets the expectations of stakeholders—who could also include investors-- and designers.