**Software Development Cycle Self Reflection**

I wanted to include some more documentation showing the things I have learned in this course. The start of any project should be the user’s needs, Companies may ask users several questions in order to gauge what they expect from the program. This is called user stories, which are very important to beginning development. The user stories are interpreted by the product owner and added into the product backlog. The product backlog contains the user’s story as well as are rated on difficulty level and may also be prioritized. The product owner will plan out expected inputs and functionality for every specific user story. This helps the developer understand exactly what kind of feature is needed and is shown the expectations so they can focus mostly on coding. User stories help move the project forward and allow the team develop on user centered program.

Taking this program has opened my eyes to different types of development and processes. When I developed my game, I did almost everything. I did collaborate with my friend on design, he would push me to develop more features and I did so because I knew he was right. I did make a mistake while developing though I didn’t have anyone play test my game except my friend. This led to an issue where someone who didn’t have a video game guitar thought the game was broken. It took me a week to get the patch to go through and I lost a decent amount of sales and received criticism from reviewers. It was a hard lesson, if I had implemented user stories, I could have improved the game and possibly avoided this pitfall. I think the incorporation of the product backlog is important as well not only to prioritize features, but it is a detailed planning process that can save time in the long run. I also see the value in information radiators or big visible charts. These are large graphical representations of the project kept in plain site for development teams. This sort of in your face approach can help the team stay focused on not only what has been accomplished, but what yet needs to be done. Big visual charts may also help show the stakeholders of progress made which may also increase investor confidence. The sprints are also important, sprints are planned time frames for completion of tasks. This is important because it allows the team to plan their tasks and track progression. If there are issues with progression from a certain team it will become apparent quickly, allowing the team to react and avoid bottlenecks. I am also a fan of the two-pizza rule, allowing meetings only to a size of what two pizzas will feed. This allows meetings to be and stay productive, meetings are important, but they should be focused and only have needed team members in the meeting. I believe all these agile practices are important and will use these practices in future development projects.

Team members are important as they are who you will be working with to help develop the product. Since you will be working with different developers, you want to have a good repour with your team. There are some traits that you can exhibit to increase that repour, which will not only help your team but the project as well. Some characteristics of a good team member include a person who is goal oriented instead of self-oriented, can compromise, are respectful and polite. Being able to listen and convey ideas is also important as most collaborations will require communication. There are more traits that good team members have these are just a few. I think it is important for everyone to ask themselves this question and analyze how many of these traits they themselves exhibit. This sort of self-reflection can not only help you as a team member but may rub off on other people as well. This will help the projects progress and help maintain employee satisfaction with their jobs, everyone wins in that scenario and I think it’s the best way to go.