Jhariel Almonte Ventura

February 19, 2023

Software Development Lifecycle

Sprint Review and Retrospective

Throughout multiple the hypothetical cycles of the SNHU Travel project I took part in most of the roles provided by the Scrum framework. During module four as a tester, and module five as a developer. As a developer, I implemented features proposed in the product backlog and as asked by the Product Owner, specifically, the “detox and wellness” destinations update, which completely changed the fundamental structure of the program from a traveling history and preferences-based offers and advertisements, to a very particular type of destination and vacation type. Additionally, as a tester, I developed acceptance-criteria-based test cases to ensure that the delivery of the final product during the given iteration followed the team’s definition of “done”, as every cased or user story had a specific acceptance criteria and expected results.

As the scrum-agile approach allow us to break down user stories into features, which then broken down into smaller tasks and implemented in a priority-based sequence, it is significantly easier to implement these features, make progress, resolve any code-related issues, and even implement new features during the same iteration depending on the complexity of the feature which is relative to the cycle.

The best example would be the last feature implementation, as the stakeholders suddenly requested an overhaul of the program, which raised some concerns among the team members as implementing these changes meant that we would have to go back to the drawing board and start from scratch with new features. Additionally, the deadline for the final sprint iteration and delivery date was unchanged, which could have been the leading cause of “crunch”. However, as the framework specializes in sudden changes, we were able to restructure the user stories, prioritize the new features and changes in product backlog, and eventually deliver the requested features, hence, the expected results.

I strongly believe that the core and one of the strongest aspects/factors of the Scrum framework is the communication workflows it facilitates. Additionally, I personally aim to create or encourage concise and straightforward communication in every situation I take part on, school and this course in particular are included. Furthermore, as we are encourage to engage with other students in a weekly basis I was able to get in touch with most, if not all members of the class during the course during the discussions. I was intrigued by the insight and feedback provided by my peers, and most importantly invested in their response to the topic of the discussion, as I often found myself not only providing clear, concise, and non-biased feedback or alternatives in my responses to their posts, but also learning new things and observing the framework from a different perspective. Once again, I strongly believe creating communication workflows in any fast-phased environment or scenario is required, regardless of the level of professionalism and expectations involved in the process, which takes us back to the communication workflow facilitated by the framework in the form of daily scrum meetings, and even project management tools.

In addition to the organizational tools covered during the course such as GitHub’s project management feature, Microsoft Azure Boards, and Jira, we also have the agile twelve agile principles. The project management tools essentially facilitate the management of user stories, where we can easily prioritize, track, and document for implementation or testing, which were simulated by the test case spreadsheet during module four where we created user stories and test cases for each feature requested by the product owner. Additionally, the twelve agile principles served as a handbook or a “bible” for the software development cycle and iterations, as among the principles we could find the following which were present during the development process of the SNHU Travel project; sustainable development, welcome changing requirements which is a perfect example of this practice considering the last changes requested by the stakeholders, and deliver working solutions frequently.

Scrum is essentially a software development tool ( or framework) that facilitates communication between client and every member of the development team to deliver and meet the needs of the client. The pros of this framework are the internal communication workflows and the delivery of product and feedback it facilitates, and its flexibility to the stakeholder’s request and new features, although the latest is a double-edge sword, as the framework emphasizes heavily the delivery of the client’s request, which can potentially place the development team in a disadvantage depending on the requirements, criteria and requests made by the clients. In short words, I strongly believe that if we had decided to implement a Waterfall approach for this project, it would have been significantly more difficult to deliver a final product, as in contrary to Scrum, there is no space for sudden changes in Waterfall as a strictly plan-based framework/structure.