Scrum-Agile Final

Southern New Hampshire University   
CS-250 Software Development Lifecycle

During the SNHU Travel project I took on all the roles of a scrum agile team member.  My first task was as the Product Owner.  The Product Owner’s main responsibilities include defining the user stories, updating and prioritizing the team backlog, which helps streamline the priorities of the project.  First, I collected a list of features that users would like to see in the SNHU Travel project, this is called user stories.  I gathered the user stories and compiled them into a list called the Product Backlog.  Upon adding user stories to the Product Backlog, I also prioritized each user stories difficulty level, starting from the most difficult, going down to the least difficult.  This was done to help make an estimation on how many resources were needed for completion.  I then took on the role of Developer in the SNHU Travel Project.  Some of the responsibilities of a developer can include reading the product backlog, interpreting the user story to working code, working with tester’s to fine tune the product.  While taking the role of developer there had been some changes to the Product Backlog.  The user’s wanted a feature where they could view the top 5 detox wellness vacations.  The users also requested to have some sort of link that would take them to the top travel package for the vacation they were viewing.  Since agile is very much about adapting to changes in the plan, I got to work on making these new changes.  I first started by researching for the top 5 detox and wellness vacation spots.  Using the Java programming language, I imported my top 5 selection images into the build.  I also added the name and location as well as a short description of each vacation spot.  The next and previous buttons were added so the user could jump between locations easily.  Once I was sure of my locations, I then did research on the best deals for that travel package.  I copied the URL of those deals into string variables in Java.  Finally, I created a Go To Travel Package button and linked each URL to the correct page.  Since documentation is also important, I left comments in my code to explain what it was doing and why.  Once the code was completed, I assumed the role of tester.  My job as a tester was to ensure all of the code did what it was supposed to do, as well as test for possible bugs.  I checked every page multiple times to ensure the correct travel destinations were being displayed.  The developer created an integer variable to track which page was being displayed.  This was to ensure the right travel package website would appear when the user clicked on it.  The code seemed correct, but I went back and forth from iteration 5 to 1 and then through the entire list again to make sure the logic was correct.  It was right as every travel package showed up correctly and every website connected to the right travel package.  Finally, I double checked the user story to ensure all the requested features were implemented, which they had been.  My last role on the scrum team was that of the Scrum Master.  Some of the roles and responsibilities for the Scrum Master are promoting and supporting scrum which include rules, values and theory.  The Scrum Master is also known as a servant leader to the scrum team which means they focus on the growth and wellbeing of the team.  I assumed the role of Scrum Master in our 6-1 Discussion and I replicated the role the best that I could.  In my post I created a list of things our team could do to transition from waterfall to agile.  The list included agile concepts such as user stories, product backlog entries and sprint goals.  The list also included other ideas that could help increase worker satisfaction and development.  These ideas included getting with developers and getting feedback to ensure they weren’t overwhelmed with the work.  Assigning developer’s to jobs outside their immediate knowledge to promote enhanced utility and their personal development, creating an open-door policy to let every team member know whatever problem they are facing can be solved without any fear of retaliation.  I also went to all the other team members' posts and tried to implement the values of scrum in their posts by asking questions.  I also let my team know how I agreed with them when their posts hit on scrum topics.  I also responded when team members would bring up ideas on my post as well.  I think this was a great exercise and I learned a lot from it.  All of these roles helped me better understand the scrum agile process as well as complete each assignment for the SNHU travel project.

When I created my video game, I knew exactly what I wanted to do, it was easy to translate that to code.  I had never interpreted other user’s requests so creating user stories and assigning priorities was a first for me.  Using the Scrum agile approach helped me understand what the user wanted because I had to translate the user story into the product backlog.  This type of data entry was a sort of precursor to the code as I was specifically stating what the user wanted and what I expected as far as inputs and expected outcomes.  I created the expectations for buttons and dropdown menus and wrote down what I expected each thing to do, even down to ascending or descending order.  As the Product Owner it was my job to take that user’s vision and write a plan of action for the developers as well as prioritize those user stories.  This helped put the project into focus as I didn’t just start coding before I understood what it was I was supposed to be coding.  I believe this portion of the project was vital to understanding and relaying what the vision of the product was.

Scrum-agile was created to be more dynamic and responsive than waterfall. An example of how the Scrum agile approach supported the project completion even when something has been changed, would be when the user’s wanted to switch to the detox and wellness vacation option.  This is where the user stories translation into the product backlog is very helpful.  The product backlog was very clear and even though we had already completed a top 5 vacation spots option, it wasn’t too hard to adapt to this new idea.  I simply read the user story and details from the product backlog and started to code.  If the user story isn’t very clear and it isn’t clear what they actually want, this would have been a very hard process, this is where the Scrum agile approach helps the project, which clear concise planning of what to do and what is expected.

Communication is important in anything you are trying to achieve. I think the best demonstration of communicating effectively with my team for this project would be the 6-1 discussion.  In this discussion I took on the role of Scrum Master as it was the only one not taken yet.  This is the first time I did this kind of group assignment and I was not sure if I understood the assignment correctly, I was timid about going in a direction and being wrong.  Since I was assuming the role of the Scrum Master, I decided to throw that aside and laid out around 6 ideas that would help our team go from agile to waterfall.  I didn’t think I would have been a good Scrum Master if I wasn’t willing to take a risk, so I did my best to take the first step.  I communicated with my team members and agreed with their scrum ideas as well as tried to implement scrum ideas through my questions.  I also tried to create a positive collaboration environment, which is also something the Scrum Master is responsible for.  As a team we worked together to help define our roles and responsibilities in an agile environment.  This to me was the best communication and collaboration assignment we’ve had.   It was very effective in not only helping us understand agile roles but increased our communication skills. 

There were many Scrum-agile principles that helped my team achieve success in this project.  Getting customer feedback was important in the beginning as it allowed us to understand what the customer expects to see, these are called user stories.  Once the team has the user stories the Product Owner will read the stories and enter these stories in a product backlog, this is a list of items that need to be completed for the project.  A level of prioritization is given to each user story to estimate the amount of work needed to completion.  The product backlog has the user story, information about which inputs and user options are needed and has more information on what the Product Owner expects from each outcome.  This information helps the developer understand exactly what is needed to code so they can just focus on coding.  Sprints are created in order to work on items in the product backlog.  A sprint is an assignment of tasks for teams to complete in the product backlog.  Sprints usually don’t last more than two months, probably so the team can create short achievable goals and evaluate the progress after that.  Once the developers read and understand their assignments, they start coding their part.  Once they are completed the code is handed off to the testers to ensure the user story has been completed as asked and they check for bugs.  Once that section is complete and correct the person assigned to the task updates their completion.  If a team finishes early, they may be reassigned to another team that may be struggling.  It is important that the Product Owner not only understands the idea of the product but understands his teams and what they are capable of.  Weekly meetings can be common to communicate any issues teams might be having.  While meetings are important it is also important to make sure these meetings are productive, this is where the two-pizza rule comes in.  The two-pizza rule is having your meeting no larger than what two pizzas can feed.  This encourages more planning to keep productivity high.  We used all these practices in order to complete our SNHU project.

I think the Scrum agile approach was very effective in the completion of the SNHU Travel project for a couple of reasons.  If you think about what the waterfall process is, it’s assuming you know every step along the way and nothing is going to change, so you plan everything out for the whole project.  In mathematics this would be similar to a straight line, now straight lines are useful but calculus is even more useful.  Not only can you use calculus on a straight line, but you can use derivatives on any portion of the equation to find the slope at that specific point.  Calculus is more dynamic where a straight line is more static.  I think the same way about waterfall being static and agile being dynamic.  No one can guess the development struggles and changes in requested features throughout a projects lifecycle; therefore waterfall is inferior in my opinion.  Agile is about incremental planning, regular coordination with teams and time specific deadlines.  They are about achieving small goals all in the direction of the overall goal.  Agile is constantly evaluating the product for success and gives itself leeway for changes where waterfall doesn’t.  These are the pros of agile in my opinion.  The only negative thing that I could possibly see in the agile approach is the number of meetings that are needed.  I understand these meetings are important for communication, it would be one the leaders to ensure these meetings are productive and serve a purpose.  The two-pizza rule can help reduce unnecessary people being at a meeting that doesn’t involve them.  This is the only possible negative thing I can think of for a scrum agile approach.  I think the Scrum Agile approach was the best way to work on the SNHU project for a couple of reasons.  The interpretation of user stories into the Product Backlog really helped the developer understand what the user wanted and what the Product Owner expected to see coded.  This helps the Developer focus only on coding these features.  Once the code is complete the testers check the code for bugs and errors and ensure the code represents the features of the Product Backlog.  The Scrum Master helps reiterate the agile practices as well as helps in employee development and satisfaction, which helps with teamwork and employee production.  All of these practices help bring the project from idea to completion, every step had a purpose and those steps lead to our big goals. Communication is important in anything we do, this is why understanding the users and translating those requests clearly is important. Communication is important between teams as well as you always want to be able to help solve any problems they may be facing. I think the agile process was the best option for our SNHU project for those reasons.