Sprint Review and Retrospective

On our Scrum-Agile team, we had a Product Owner, Tester, Developer, and Scrum Master. The Product Owner’s responsibility was ordering our product backlog, and creating and organizing user stories based on customer feedback. The Scrum Master was responsible for enforcing Scrum-Agile practices while also ensuring that teams were not side tracked with unproductive work. Our Developers were responsible for writing and creating software based on the user stories and backlog. Our Testers were responsible for testing the software that the developers created. Our Scrum-Agile team used these roles and qualities to gather feedback from our customers to create and implement a slide show feature that showcased the top 5 destinations for detoxing and relaxation for our customer SNHU Travel. Our Product Owner was responsible for gathering the feedback and creating the corresponding user stories and product backlog. Our Scrum Master conducted our daily standups and ensured that our teams were all on the same page and able to focus on their work without distractions. The Developers were able to create and develop a software feature that allowed users to view the top 5 detox vacation destinations. Finally, our testers were able to adjust their test cases to handle the request for a slide show.

The Scrum-Agile approach to software development allowed the Product Owner to create user stories, and organize our product backlog based on the created stories. These user stories are created and organized by the amount of work associated with the task, and how critical the completion of the feature is to the final product. These tasks derived and categorized from the user stories are then handed off to the development and testing team. The testing team will use the successful acceptance criteria that the Product Owner created to create their pass cases for their tests while the development team will create the desired software using the user story as their guide.

While developing a feature for the SNHU travel app, we were tasked with changing how we wanted to display the information based on information that our product owner received from customer feedback. While frustrating at first, the Agile-Scrum process allowed for the team to quickly adapt to the requested change. First, the product owner reorganized our backlog to fit the new user story. After that, the team met up to get each other up to speed and coordinate changes that need to be made to the software. Next, our testers adjusted their passing cases to fit the new acceptance criteria. Finally, our developers and testers worked together to develop their final product.

Here is an example of communication that took place between me and other members of our Agile-Scrum team:

Example Email:

Dear [Product Owner] and [Tester],

My team and I have been working on the newest version of our software and we needed some clarification on a few things before moving further with development. If you could answer some questions for me, we could have a better understanding of our development goals.

For [Product Owner]:

* Could you clarify our Vision Statement? I understand that our “Top 5 Destinations” slide show user story fits the scope of our Visions statement, but with more clarification, I feel we can deliver a better performing product.
* Can we reprioritize some of our product backlog items? We have discovered that account creation and mobile application development maybe larger workloads than anticipated. We should get together and run a workload estimation meeting to accomplish this.

For [Tester]

* Have there been any developments in our pass/fail cases for getting users to our recommended vacation packages page? I know our last tests were showing that users with preferences set in their accounts were being recommended items they said they didn’t want to see. We pushed an update this morning that should have fixed that.

Thanks,

Evan

This message was sent to gain clarification, and reprioritize product backlog items, and to gain information on passing/failing test cases. Asking these questions to the Product Owner will allow the team to be on the same page as far as our vision statement goes. At the time, developers were struggling with their end goals because they felt our vision statement was too broad. We also requested that some backlog be reorganized. A few backlog items were more complex than originally thought and need to be reprioritized. The message to the Tester was sent to attempt to collaborate and get on the same page when it came to pass and fail scenarios, as updates were made that hadn’t been followed up on yet.

Our team used Scrum Events like standups to collaborate, coordinate, and inform one another about our work for the day. The Scrum Master would gather all of the teams together every morning and go over what we did the day before, what we were doing today, and what was getting in the way of meeting our goals. Our Scrum Master would use that information to help developers with work they were struggling with and keep our teams focused on their work by taking care of any distractions or work impediments. Our team also participated in Sprints to develop different parts of our software. A Sprint was a short period of time where we focused on developing and delivering an important part of our product. Tools like JIRA allowed us to collaborate in an online environment and turn our work into smaller tasks that were easier to track and work on.

The Scrum-Agile approach to the software development life cycle was useful in our cases. This approach to software development is useful because it encourages teamwork and collaboration, and focuses on consistently delivering a product to our customer at regularly scheduled intervals. Scrum-Agile development also has its drawbacks. Products that are delivered using Scrum-Agile planning are usually changed constantly because planning is done incrementally. While a product is delivered consistently, it may not be exactly what the customer wants because of time constraints when developing.

Overall, the SNHU Travel app benefitted from using a Scrum-Agile approach. It allowed for Chada Tech to plan, develop, test, and deliver a working product to their client on time. Smaller products that are prone to update and change due to competition benefit more from having an adaptable, collaborative approach to planning and development. Larger, complex projects might require something like a Waterfall approach to planning.