## Scrum-Agile Sprint Review / Retrospective

Each member of the Scrum-agile Team has their own tasks that work together to support the project. The product owner is responsible for the backlog and compiling user stories for our project. When creating the product backlog, the product owner conducted interviews to gather stories from clients and customers. Once the interviews were done, the product owner broke the main ideas from the interviewee into user stories. These stories provided a template of the requirements needed for the program to be considered complete. During development, the product owner was most in tune with the shareholders and the owners of the project, helping to share their vision with the development team. One of the ways they did this was by prioritizing the stories on the backlog. The product owner carefully chose which stories were the most impactful and important to complete. They communicated this to the rest of the team and guided them to completion, by answering questions and providing examples.

The Scrum Master provided support to the success of this project by leading and supporting the team. One of the key ways the Scrum master supported the team was by conducting daily stand-ups. The scrum master gathered the team together and led them with concise conversation topics. The three big questions that the scrum master asked each person were, “what did you do, what are you going to do, and what problems or hindrances can you find”. After each member answered these questions, they were each given a better understanding of the entire team’s progress as a whole. During these events, the scrum master made sure the backlog was up to date and current with what everyone shared.

The developer contributed to the team by working on the tasks provided by the backlog. They choose the tasks with the most importance based on the time they had for the sprint and the size of the story. Some sprints allowed for only one large story, but other sprints allowed the developer to complete several small stories. The developer showed great communication skills by asking questions during the daily-stand up and by emailing the product owner. The developer also works with the tester to ensure that the requirements for each story are met.

The tester used the provided user stories and created test cases. These test cases allowed us to document the process of implementing user stories and fulfilling acceptance criteria. Testing involves making sure the code functions as intended and the required content is present. The Test also made sure to ask questions in the daily stand-up and through email about anything they were confused about.

The Scrum-agile approach to the SDLC revolves heavily around user stories and what they bring to the project. User stories are important because they allow the team to gain insight into what customers and shareholders are looking for in a project. The product owner conducts interviews and creates user stories, combining them together into the product backlog. The developers work on the stories in the backlog in order of importance, and what they can reasonably complete in a sprint. The tester makes test cases out of each story, then checks the system to make sure the acceptance criteria are met. The tester made sure that any questions or vagueness from the user stories were cleared up by the product manager, ensuring accuracy.

At one point during development, it was decided that the project would take on a slideshow-like format to display information to the user. The product owner sent out emails to the team with examples of what should be done. The developer then went ahead and made the necessary adjustments, while still maintaining the base requirements and components of the program. Because communication is the cornerstone of agile development, facilitating changes was easy and simple.

Sample communication 1:

To: Product Owner Dave, Tester Dan

Subject: New Plan Clarification.

Hey Dave, Dan,

I am developing a back-end interface for our clients and require some clarification on the new development plan.

* Is the old product backlog still accurate or will there be changes to the order and importance?
* Will the tester have new conditions that must be met for the stories to be considered complete?
* Will the sprints and events keep the same time and schedule or will there be changes?

Thanks,

Tim

Sample communication 2:

To: Product Owner Dave

Subject: User Story Clarification.

Hey Dave,

I am developing test cases for three of your user stories, and I have several questions about

layout and a few other things. I was hoping you could provide an example and feedback, thanks

User Story One

* Will the main page feature destinations that the user can navigate to and find details on?
* Are we offering a few vacation packages per destination or many?
* If we are doing a slideshow format, how will the customer navigate back home?

User Story Two

* Should the destinations be numbered in order?
* Should the buttons be a specific color or look a certain way?
* What color should the banner be?

User Story Three

* Can users get to their profiles from the main page?
* Should the vacation history list be a slideshow or a list?
* Will the list contain all vacation history or just the most recent?

Thanks,

Tim

Here are two samples of communication from an agile development team. The first sample is from the developer to the product owner after a major development change. The developer looks to ask questions directly related to the task he is working on and the events that include him. He checks with the product owner on the backlog to get an idea of other changes that may have stemmed from the overall direction change of the project. He also verifies whether the scrum events that he is required to attend are at the same time or have been changed. In the second sample, The tester is looking for clarification on the acceptance criteria for the test cases. He breaks each user story down and asks individual questions about each. By asking questions now, they prevent mishaps and errors further in development.

Scrum-agile development has many tools for organization and communication to enhance project productivity. One invaluable tool for the team was the product backlog. This board allowed the team to easily tell which tasks were most important and how long they would take. Another scrum tool that was helpful was the daily stand-up event. During this event, the team can start their tasks for the day with a full understanding of where the project currently is and what needs to be done. The product backlog is a large part of the daily stand-up, as each day during the meeting, the board is updated. In addition, the test cases put together by the tester were a great organizational tool. Having clear directives for each user story made it easy for any team member to follow along and understand the work being done.

Agile development has many great features that make it a strong development methodology, but there are a few things that agile lacks. One of these things is a clear picture of the project from the start. Agile planning involves only the base essence of what the project is to accomplish. While this leaves lots of room for change and improvement, it can be harder for developers to see the whole project and scope. In addition to this, the agile development time table is ambiguous and follows the flow of the work instead of following the flow of time like a traditional project. Traditional projects set dates and timeframes far in advance and try to meet them. Again, because of agile ability to be flexible, setting dates becomes impossible. For all the cons however, agile still has many amazing tools and systems such as events and boards that make it an invaluable development method. Agile is a great methodology when applied to the right projects. I believe that because the company wanted customer input and an adaptable system, agile was a great choice for this development cycle.