**Sprint Review and Retrospective**

**CS250: Software Development Life Cycle**

**By Derek Bamford**

A Scrum-agile team is made up of four distinct but equally important divisions; the Scrum Master, Product-Owner, Development Team, and Product Testers. I will briefly describe their role here and give specific examples of how each contributed to the success of the SNHU Travel™ project.

The Scrum Master is much like the ringleader of a circus, they are there to guide the project and keep everyone on task, but also know when to stand back and let the team do their jobs. A good Scrum Master has excellent people skills and can use those skills to motivate their team members accordingly. For the SNHU Travel™ project, the Scrum Master helped to organize design changes that the stakeholders had relayed to the Product-Owner. When goals suddenly change it can be detrimental to the morale of the team, especially if it is late in the development cycle, however, the Scrum Master was able to refocus our team to achieve the new goals we had been assigned.

The Product Owner has three key tasks to perform in Scrum. The first task is to be a leader by overseeing product goals and objectives, they must set up the sprint goals, and user stories, and perform product backlogging. For this project, the Product Owner had the unenviable task of disseminating the changes wanted by the stakeholders to the Scrum Master, effectively rewriting the sprint goals, this meant that the Product Owner must amend the user stories so that the Development Team can pivot their work and adjust completion timelines if needed.

The Development Team is the group of people who build the software to the stakeholder’s specifications. For this project, the Development Team took user stories, which is information about what the end user would like to see in the software, and divided the work by priority. Each team member was required to give a brief update on their progress at the daily scrum meeting and discuss any impediments that may be hindering progress. When the stakeholders decided to pivot to relaxation and wellness, the team quickly reprioritized and produced a new action plan. Once each part of the software was in a completed state, it was passed along to the Product Testers.

The Product Tester’s role is to check each completed version of the software for bugs. They design Pass/Fail criteria and build test cases for each aspect of the software. When the Design Team changed the software to meet the stakeholders’ new wishes, the Product Testers needed to revise both the Pass/Fail criteria and adjust the test cases to better suit the revised software.

A key component that drives the development of software in SDLC is user stories. Scrum-agile allows the Development Team to quickly iterate software versions for testing by breaking the user stories down to priority levels for the team to tackle, this approach allows for simultaneous work on all user stories. Fail early and fail often is the motto of the day in Scrum-agile and it is this thought process that allowed to SNHU Travel™ team to pivot swiftly to relaxation and wellness at the behest of the stakeholders completing all tasks on time and to the specification of the stakeholders. Affinity grouping allowed us to quickly reevaluate our timelines and finish within the original deadlines.

With many different teams working on various aspects of the project communication is of the utmost importance to make sure that everyone is on task and no overlapping or unnecessary work is being done. To better express this, I have included a small sample of how effective email communication occurred while working on the SNHU Travel™ project. This communication is between the Software Development Team and the Product Owner and also the Testing Team to the Software Development Team.

To: Product Owner

From: Software Developer Team

Subject: User Stories

Dear Product Owner,

The team and I have looked at the user stories and begun development on the software changes discussed in the user focus groups. A few team members have asked for clarification on a couple of the stories.

How should users traverse from one recommended vacation package to the next?

Should commenting be allowed from those who are verified, package purchasers?

By default, should we sort by recommended packages or by price?

I look forward to your answers. If longer communication is needed, please call, or send a Teams meeting.

Derek Bamford

Senior Software Developer

SNHU Travel ™

---------------------------------------------------------------------------------------------------------------------

To: Product Tester

From: Software Development Team

Subject: Software Bug Analysis

Dear Tester,

I reviewed the weekly bug report and found most of these issues should not take much time to fix, however, some of the issues we have not been able to replicate some on our systems. I would like to schedule a meeting between someone from your department and one of our Junior Developers to go over these bugs, hopefully, if we can watch the bug being replicated can glean a bit more insight into what is causing the issues.

Derek Bamford

Senior Software Developer

SNHU Travel ™

In the first email, the Development team is looking for clarification on some of the user stories provided. It is always better to communicate concerns or ask for additional information than to assume things and spend time working on projects that are not important. In this example, the Product Owner was able to clarify some of the user stories and the Development Team could then get to work and implement these changes. The second email is an example of when an email is just the initial communication method and additional communication is necessary either by phone, online or in person. In this case, the software developer would like for an in-person meeting to visually see how things were breaking down in the software, sometimes a quick 5-minute face-to-face can be better than hours of back-and-forth emails. The Junior Developer was able to sit with the tester and review the bugs, which he then relayed to the team more effectively than an email chain could have.

Scrum-agile has many built-in organizational tools that help teams become successful during a Sprint. The one tool that was most effective for my team was the Big Visual Charts (BVC), each day at the daily scrum meetings every team member could visually see where each task was in its development, and the board allowed for discussion and easy additions to the user stories or sub-stories. The BVCs lend themselves effectively to the Agile principle of value-based prioritization, the team used Poker Planning to grade the difficulty of each user story, this allowed the team to distribute low-difficulty tasks to a single member, while higher-difficulty tasks could be worked on by multiple people. It was this type of work distribution that allowed us to keep our product timelines.

In the Scrum-agile approach, a Sprint Retrospective is a recurring meeting dedicated to discussing what went well and what did not in the previous sprint, and in that same vein, I would like to take the opportunity to make similarly discuss the results of the SNHU Travel™ project.

Scrum-agile allowed my team to meet our goals quickly and effectively, however, I could see a scenario where the project becomes too unwieldy because of scope creep and the lack of a definitive end date. I liked that all the priorities could easily be visualized by everyone on the team at any time and that discussions were held daily to bring everyone up to speed on the previous day’s work, but again the downside to this is that daily team meetings can be frustrating to its members if they begin to perceive it having no value to them. Finally, the quickness of the development and the turnaround time for testing allowed for quicker feedback to amend the software, this could also be problematic if one falls into an endless cycle of writing and testing without determining an endpoint.

Scrum-agile was highly effective for the SNHU Travel™ project and was ideal for the size and scope of the project. If I were to change anything I may add the waterfall method to the testing team so that we have a more concrete structured way of evaluating the software and so that everyone does not go off and do their own thing when testing.