

**Sprint review and Retrospective**

**By: Alexis Mcvey**

In the software development life cycle, the Agile methodology approach has proven to be a more efficient method of implementation. Within agile scrum, team collaboration is the key to a successful project development, this is largely due to the distribution of responsibilities across the team. In this review I will discuss these roles, their responsibilities and the way agile scrum contributes to SDLC. As mentioned earlier, roles and responsibilities on the scrum team are distributed, these roles include Product Owner, Scrum Master, and Development Team. Each role carries its own weight on the team, and each role is just as important as the other. The Product Owner is essentially the visionary of the team, the product owner often collaborates with stakeholders at all levels to form a collective vision of what the software should be, they brainstorm features and relay data and research they have done, via the backlog, on what consumers are wanting out of an application. Clear and effective communication is vital from the Product Owner as it is they that can either contribute to or delay competition goals. The Scrum Master acts as a liaison between the Product Owner and the Development Team. The Scrum master is responsible for facilitating daily scrum meetings where progress and impediments are discussed, and solutions are strategized. The Scrum Master helps the team stay on task with goals and strategizes effective ways the team can collaborate better, they help the product owner maintain the backlog, and the prioritization of tasks, and as well relay any changes or updates to the development team. The scrum master plays an important role in leading and coaching the organization of the effectiveness of Agile scum and how to conduct a methodology transition. The development team are responsible for creating the useable part of the software project, they implement and revise user stories and apply them to test cases. They create the base code if the project that can be built on and added to as updates and changes are needed, they also test the functionality of the code, and the features applied. The development team requires the most clear and detailed communication for them to be successful in their roles, one such example can be with projects throughout this course that required change to a developed code. The requested change needs to provide specifications that are as detailed as possible for quality assurance, the request should include details such as desired functions and their desired outputs, layouts, filters, and appearances.

A unique way Scrum-Agile operates is through the utilization of user stories, user stories are an informal, general explanation of a software feature written from the perspective of the end user or customer. Completion of these end user requests are possible through the prioritization of the product backlog, this is done by the Product Owner who in turn collaborates with stakeholders to receive specification of desire results, it is important that the backlog is always up to date and changes are added as soon as possible. User stories are focused on sprints that are worked on in increments thus encouraging collaboration throughout Scrum events, this is done during sprint planning, daily standups, sprint reviews and sprint retrospective. Collaboration, adaptability, transparency, and visibility are what make scum-agile so receptive to the end user needs and are what contributes most to their completion. User stories were used in the SNHU travel project often, they helped my team, and I understand what was expected of our application, we were able to incorporate them in test cases to be refined and implemented as desired.

In agile-scrum effective communication is vital, they are the difference between a smooth transition when it comes to sudden interruptions and change to user stories and chaos, uncertainty and a potentially low- quality application. With the right communication methods applied agile- scrum continues to show why it's so adaptable to change even in the middle of a project. An interruption that required me to apply these effective communication methods happened when the product owner informed us, during development, that the client wanted to add a different niche to our already established travel application plans. At the time I was working as a developer and found the sudden change a little stressful as it could’ve involved scrapping the whole project and starting over. The first thing I did was ask for clarification, if a new project will need to be started or is this just a new feature that can be added to our already developed code base. It turned out that it was just a new feature, still, I needed more information to ensure that what they are asking for is what they will get. This communication came in the form of an email that I had to send to the product owner to clarify certain specifications that needed to be added. The product owner then updated the product log with new prioritizations which helped me in developing further. I believe these clarification requests were effective and encouraged collaboration because it got me and the rest of the team the information that is needed, plus a better idea of how our tasks are to be prioritized so that we aren’t spending too much time working on the wrong thing. Organizational tools such as Jira and Azure were not actively used with this project, however project management tools such as those are great for scaling and the estimation process, they are great for helping the team stay on track with goals and allows adjustments any delays might cause.

Overall, I think the effectiveness of agile- scrum is evident, I think agile methodology makes organization and staying on task relatively easy. Its encouragement of team collaboration and communication makes for a better product overall and continuous delivery of functional software leave feedback for revision more feasible. However, the high dependency on a team's work ethic and discipline along with the limitations of long-term planning, and project- complexity agile scrum does have its drawbacks like any other methodology. Still, it proves to be the most effective choice for the SNHU travel project. Since the project is orientated on the customer experience, feedback on expectations to give the project the highest quality, seem aligned to agile approach and outcomes.

**Citations**

•*Scrum roles – the full breakdown - BigPicture*. (2023, March 2). BigPicture. <https://bigpicture.one/blog/scrum-roles-breakdown/>

•GeeksforGeeks. (2024, October 18). *Waterfall Model Software Engineering*. GeeksforGeeks. <https://www.geeksforgeeks.org/waterfall-model/>

•*Scrum Guide | Scrum guides*. (n.d.). <https://scrumguides.org/scrum-guide.html>

•Radigan, B. D. (n.d.). *Agile vs. waterfall project management | Atlassian*. Atlassian. <https://www.atlassian.com/agile/project-management/project-management-intro>

*Scrum, agile and the art of active listening*. (n.d.). https://resources.scrumalliance.org/Article/scrum-agile-art-active-listening