SNHU TRAVEL PROJECT:

SPRINT REVIEW AND RETROSPECTIVE

Russell k. Chimunya

Review and Retrospective: Applying Roles and Responsibilities

Throughout this course, I have assumed various roles within a Scrum team. The team was transitioning from a waterfall methodology to an Agile approach to develop an application for the customer, SNHU Travel. The team included a Product Owner, a Scrum Master, and a Development and Testing team, consisting of both Developers and Testers. In this paper, I will analyze the Scrum-Agile methods applied to this project and evaluate how these methods did or did not contribute to the final deliverable.

Product Owner

This role was crucial because it connected the customer with the Development team. As the Product Owner, I was responsible for defining the requirements for how the project would be executed. Part of these requirements came directly from the customer, while I gathered other inputs from a focus group with end users. I had the chance to see the software development life cycle (SDLC) from a non-technical perspective. In this role, I made sure that the goals, scenarios, and project requirements were clearly communicated to the team. By providing a well-organized product backlog, prioritizing tasks, and maintaining transparency, the Product Owner role allowed developers to focus on delivering value. This helped keep the project aligned with business goals and client expectations.

Scrum Master

As Scrum Master, I focused on facilitating meetings, organizing sprint planning, and helping the team remove blockers. One of the most important responsibilities in this role was leading the Daily Standup meeting, a brief session held every day that typically lasted no more than 15 minutes. The purpose of these standups was to keep the team aligned by sharing progress, identifying challenges, and planning work for the next 24 hours. During each standup, team members answered three key questions:

What did I accomplish yesterday?

What will I work on today?

What obstacles are blocking my progress?

By ensuring the meetings remained short and structured, I promoted accountability and effective communication within the team. Issues identified during these meetings could be addressed quickly outside of the standup, which helped keep the project on track. Overall, the Scrum Master role was essential in maintaining focus, removing impediments, and fostering collaboration throughout the project.

Development Team

## As part of the Development Team, I had the creative opportunity to approach coding tasks with flexibility while still aligning with team goals. Developers played a central role in transforming user stories into working software features, ensuring that each increment of the application was functional and testable. Because Agile emphasizes collaboration, it was important for developers to understand each other’s strengths and weaknesses. This awareness allowed the team to self-organize effectively during sprint planning, choosing tasks based on individual capabilities and shared goals.

## For example, when building the travel search feature, the team divided the work into smaller user stories such as designing the interface, connecting the backend, and validating inputs. By breaking the work into manageable increments and leveraging team strengths, the developers were able to complete the feature within the sprint timeline while maintaining quality and efficiency.

Testing Team

The Testers, working alongside developers, validated each feature as it was completed. This “test early and often” approach reduced errors that would have been more costly to fix later. Continuous testing ensured that quality was maintained and that customer acceptance criteria were met.

Completing User Stories

User stories were essential in guiding development and ensuring effective communication between the team and stakeholders. They allowed us to break down large requirements into smaller, actionable tasks, making the work more manageable and aligned with user priorities. Instead of attempting to deliver the entire travel booking system at once, we focused on incremental progress by addressing one feature at a time. For example, one sprint concentrated solely on developing the flight search functionality, which included designing the user interface, connecting the database, and testing input fields.

User stories also helped the team prioritize features based on customer preferences. For instance, if users expressed a strong preference for ocean-view accommodations rather than a “sort by city” option, we would focus on delivering the ocean-view feature first. This approach not only gave users immediate value but also allowed the team to plan and prepare for future enhancements without overwhelming the current sprint. Using user stories, the team stayed flexible, satisfied stakeholders, and delivered usable features.

Handling Interruptions

Interruptions are an inevitable part of software development, as it is impossible to predict when new requirements or changes may arise. While they cannot be prevented, Agile practices provide the flexibility needed to manage them effectively. When clients request modifications, such as adding new features or removing existing ones, the Agile framework allows the team to adapt without halting the entire project or starting over.

Communication

Communication is essential in Agile, as the Scrum framework cannot function without consistent and open dialogue. Agile provides multiple opportunities for communication each day, including daily standups where team members share progress and discuss challenges. These short meetings ensured that issues were identified quickly and that the team remained aligned on sprint goals.

In addition, sprint reviews allowed the team to demonstrate completed work to stakeholders, reflect on both successes and shortcomings, and determine what adjustments were needed for future sprints.

Organizational Tools

Several tools helped keep the team organized and efficient:  
• Product Backlog: Ensured transparency and prioritized the most valuable features.  
• Sprint Backlog: Provided a clear focus for each iteration.  
• Burndown Charts: Allowed the team to visualize progress and adjust workload.  
• Retrospectives: Helped the team identify strengths and areas for improvement.

Evaluating the Agile Process

Pros:  
• Flexibility to changing requirements   
• Faster delivery of value through iterative releases.  
• Stronger collaboration and communication.  
  
Cons:  
• Required constant stakeholder engagement.  
• Risk of scope creep if backlog changes were not tightly controlled.

Overall, Agile was the best approach for the SNHU Travel project. It allowed the team to remain flexible while delivering features that directly addressed customer needs. In contrast, a waterfall model would have delayed delivery until the end and would have struggled to accommodate the new booking feature.

Conclusion

The SNHU Travel project showcased how effective the Scrum-Agile methodology can be at ChadaTech. By establishing clear roles and responsibilities, completing user stories in increments, maintaining robust communication, and leveraging dedicated organizational tools, the team was able to deliver successful outcomes. While Agile demanded consistent effort and discipline, its advantages, flexibility, collaboration, and the steady delivery of value significantly outweighed any difficulties encountered.

Agile also proved to be a versatile and effective management strategy, empowering developers to self-organize, rapidly address client requests, and deliver meaningful features within project timelines. Thanks to its adaptability and emphasis on continuous feedback and improvement, Agile stands out as a sustainable approach for ongoing success. This reflection supports ChadaTech’s decision to implement the Scrum-Agile framework across all its development teams.

**References:**

**References**

Cobb, C. G. (2015). *The project manager's guide to mastering Agile: Principles and practices for an adaptive approach*. Wiley.

Cohn, M. (2019). *Succeeding with Agile: Software development using Scrum*. Addison-Wesley.

Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner’s approach* (9th ed.). McGraw-Hill.

Rigby, D. K., Sutherland, J., & Noble, A. (2018). Agile at scale. *Harvard Business Review, 96*(3), 88–96.

Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide*. Scrum.org.