**Applying Roles**

**Demonstrate how the various roles on your Scrum-Agile Team specifically contributed to the success of a project.**

The product owner developed clear user stories for the product backlog, each with its own priority, effort level, and completion criteria. This made development more streamlined, and it also further elucidated what the client was looking for. The product owner also communicated with the client to glean the requirements in the first place—when requirements changed, the product owner relayed these changes to the team. This is how the team knew to pivot during development, and the subsequent changes made to the product backlog by the product owner made this pivot easier to understand.

The scrum master handled the specifics of the sprints. He facilitated the team’s sprint planning, review, and retrospective so that each phase was valuable to development. He also led the daily standups to make sure each member hit the key points and to mediate conflicts or obstacles. Especially when the client’s requirements changed suddenly, the scrum master was an ambassador for the team as the primary communication point with the product owner. Team woes about the changes were vented to the master, then the master would tactfully translate concerns to the product owner.

The developer worked on the product itself, producing working iterations of the system as the client’s requirements changed. By following the product owner’s guidance via user stories, the developer tweaked the code and content of the system to fulfill client expectations. Upon receiving feedback from the tester, the developer also edited existing code to make it function as intended. Finally, she communicated with the product owner to clear up confusion on the changed project requirements, and all of these actions combined led to a client-approved product.

The tester made sure each code implementation worked as intended using specific criteria to check. Each user story was checked to make sure it contained the visual and functional aspects it was intended to, and if it didn’t, this feedback was provided for developers to address. The tester prevented a broken product by finding errors early and often. He also worked with the product owner via email to ensure his full understanding of the test cases.

**Completing User Stories**

**Describe how a Scrum-Agile approach to the software development life cycle (SDLC) helped user stories come to completion.**

After each batch of user stories was completed for each sprint, the code was tested by trained testers. This meant each user story was investigated for completion using specific criteria. While testing still happens using a waterfall structure, it doesn’t happen as early or frequently as with Agile. Frequent testing of user story implementations meant problems were found and addressed much sooner, and time was budgeted into the subsequent sprint(s) to fix them. This prevented a scramble or “death march” toward the project’s deadline to fix all issues at once.

Another way the Scrum-Agile approach aided the completion of user stories was through daily standups. Just like with frequent testing, obstacles are identified and addressed quickly when each team member can share their challenges daily. This prevented delays in development, so it contributed to user stories getting finished on time.

**Handling Interruptions**

**Describe how a Scrum-Agile approach supported project completion when the project was interrupted and changed direction.**

When the project was interrupted and changed direction, already having a Scrum-Agile plan in place was useful. Agile expects change; this is one of its principles. Through guided collaboration during sprint planning, the team was able to regroup and figure out how to move forward together. It was a matter of the product owner writing new user stories or tweaking completed ones to suit the new project epics, then the team determining how the work would get done to reach the deadline successfully.

The daily standups were also useful for handling the project interruption. For example, the developer, Nicole, seemed particularly upset about the sudden changes—this was an obstacle she brought up during a daily standup, and the team determined how to address her needs. She sent an email to the product owner (see next section) that asked key questions about the changes which, when answered, would enable her to move forward with her work. This collaborative approach to problem-solving meant her issue got fixed sooner, but it also provided a supportive, sustainable work environment for her, which fulfills another Agile principle.

**Communication**

**Demonstrate your ability to communicate effectively with your team by providing samples of your communication.**

***Email from Tester to Product Owner***

To: Christy

Subject: Further User Story Clarifications

Dear Christy,

I have looked at the slide show you provided, and I thank you for the mockup which provided guidance on the top destinations lists (popular and personalized). I’m revising my test cases based on this feedback, but I still need guidance to clearly define the necessary tests. Can you answer the following questions for me?

User Story #5 – Personalized List

* Is the slide show meant to display in-browser or as a downloadable?
* If in-browser, will the slide show open on the same page or a new one?
* How will the user’s preferences and history be used to personalize the list?

User Story #4 – Filter Button Features

* What are all the vacation types available for users to choose from?

User Story #3 – Vacation Type List in Profile Settings

* Is the development of a mobile app expected to be part of this project or a later one?
* Will the mobile app include identical functionality as the website?
* How will changes (such as profile settings) made in the app reflect on desktop, and vice versa?

Thank you in advance for your response.

Brian

***Email from Developer to Product Owner***

To: Christy

Subject: Development Changes

Christy,

Thank you for bringing the recent project changes to my attention! I need to review some items with you to move forward with the new development plan.

1. Will the regular and personalized lists remain with a new detox/wellness focus?
2. Will the profile preferences list need to be narrowed to just detox/wellness types?
3. Will the user be able to book *only* detox/wellness travel packages, or other types as well?

Essentially, I need to know what aspects of the previous user stories can be kept and which ones need to be scrapped. My overarching question is: **Is the new detox/wellness focus the all-encompassing focus of the booking software?**

I look forward to your prompt response and the updated backlog. Thank you in advance for your time!

Sincerely,

Nicole

**Explain why your examples were effective in their context and how they encouraged collaboration among team members.**

These examples were effective because they showed how an email can be professional, polite, and to-the-point. Lists were used to visually highlight the most important information—the questions that needed answering. Since brief and direct communication can sometimes be misconstrued as rude, these emails take care to provide some positive feedback without being excessive, such as by using “Sincerely,” to sign off, or by beginning with a “thank you for…” remark. Collaboration flourishes when team members feel respected by their colleagues, and thoughtful word choices like these can go a long way to make that happen. Listing key questions also made it easier for the product owner to respond in the best way possible, by answering each question item by item.

**Organizational Tools**

**Evaluate the organizational tools and Scrum-Agile principles that helped your team be successful.**

One of the principles of Agile is providing client satisfaction early and often with frequent product implementations. Through the team’s one-week sprints during this five-week process, the client was repeatedly able to see and interact with working prototypes of their vision. This is thanks to the collective efforts of the team, but also to good organization. Using VersionOne as the primary means of viewing the backlog, tasks, goals, issues, and more was crucial to keeping each team member in the loop beyond what’s brought up in the daily scrum or weekly planning/review. Having a highly visual tool like this to organize work meant daily scrums were held with less catch-up conversation, since everyone could see who was working on or finished with what tasks. Instead, the conversations focused more on obstacles and future development.

Another principle that was important for the team to remember is welcoming changing requirements. This meant the team resisted morale loss when requirements changed late in development by remembering it as part of the process. Another principle, working together daily throughout the project, meant the team could regroup and strategize new development tasks almost as soon as the changes were made.

**Evaluating Agile Process**

**Describe the pros and cons that the Scrum-Agile approach presented during the SNHU Travel project.**

The Scrum-Agile approach allowed for quick adaptation to changes, constant collaboration, and multiple working iterations of the product. It gave the team freedom to produce less-than-perfect work so long as the code functioned as intended, and this “just barely good enough” ideal led to the realization of a project that had been made, tested, and reviewed multiple times before its final draft. With the team working so closely together daily, no one member was left to shoulder all the work or find themselves with little to do.

A bottleneck of the project’s work was relying on the product owner for information. Since she wasn’t expected to attend every or any daily scrum, this meant having to repeatedly ask her for clarification or backlog updates. Scheduling a face-to-face or virtual meeting with a busy co-worker isn’t always possible in a timely manner—similarly, it couldn’t be predicted how soon the product owner would reply to a crucial email, yet timeliness was so important for such a short project. Needing her for so much information to move forward with development, especially when late-development changes were made, was difficult and a con of using the Scrum-Agile approach.

**Determine whether a Scrum-Agile approach was the best approach for the SNHU Travel development project.**

Given the short time frame for this project (five weeks), it could be that waterfall would’ve been a perfectly reasonable approach to take. However, with the sudden switch up of client requirements mid-project, the team might’ve suffered more had they not been using Agile. Despite the short timeframe, the client’s requirements were simply not fixed enough to support a rigid structure like waterfall, so it’s clear that the Scrum-Agile approach was the right option, especially because it made the pivot point much less painful.