Maxwell Mueller

SNHU: CS-250 Software Development Lifecycle

8/27/24

Project: Sprint Review and Retrospective

* **Applying Roles**: Demonstrate how the various roles on your Scrum-Agile Team specifically contributed to the success of a project. Use specific examples from your experiences.
  + The Product Owner (PO) was responsible for defining the product vision and ensuring that the team was building the right product to meet user needs. They worked closely with stakeholders, such as instructors and students, to gather requirements and prioritize features.
  + The Scrum Master facilitated the Scrum ceremonies, removed roadblocks, and ensured that the team followed Agile principles we chose to adapt. They helped the team maintain a sustainable pace and fostered a culture of continuous improvement and successful sprint deliveries.
  + The Developer was responsible for delivering the product in the increments as defined in the sprints by the Scrum Master and Product Owner. They worked collaboratively to turn backlog items into working software, ensuring quality and functionality.
  + The Tester was responsible for working primarily with the Product Owner and Developer to create tests that thoroughly test the increment that the developer produced during the sprint. The Tester must ensure that they create tests to test all the functionality of the sprint item, as described in the User Story.
* **Completing User Stories**: Describe how a Scrum-Agile approach to the software development life cycle (SDLC) helped user stories come to completion. Use specific examples from your experiences.
  + The two biggest Scrum-Agile principles that helped user stories come to completion were the iterative development environment and the communication and collaboration between stakeholders and team members. Specifically, during our project we used email communication and other forms of communication to discuss what deliverables we all were focused on and what we planned to contribute to the discussion. This allowed us to plan our responses and craft our proposal so we could hit every point in the prompt, and we could iterate our plans based on what the other team member contributed.
* **Handling Interruptions**: Describe how a Scrum-Agile approach supported project completion when the project was interrupted and changed direction. Use specific examples from your experiences.
  + A Scrum-Agile approach is inherently designed to handle changes and interruptions, making it an effective methodology for projects that need to pivot or adapt to new circumstances. This was especially helpful for us as we had different team members with different real-life responsibilities. The iterative environment allowed us to contribute when we could, while ensuring that no other team member was held up by the absence of a deliverable from another team member.
* **Communication**: Demonstrate your ability to communicate effectively with your team by providing samples of your communication. Explain why your examples were effective in their context and how they encouraged collaboration among team members.
  + Again, for our experience, communication amongst team members was the most important thing for successful project completion. This communication was concise, focused on the problem at hand, and provided a solution, which helped the team quickly understand the situation and make informed decisions. By raising the issue early, the team was able to discuss it and adjust the sprint plan, accordingly, avoiding last-minute surprises.
  + For example, Jerod emailed us on Wednesday to remind us of the deliverables due later that day, if he had not helped us with communicating deadlines, we may not have had our posts done on time. This was extremely effective as it not only prompted us to finish our deliverables, but it also opened an avenue through which we could communicate throughout the project.
* **Organizational Tools**: Evaluate the organizational tools and Scrum-Agile principles that helped your team be successful. Reference the Scrum events in relation to the effectiveness of the tools.
  + Obviously, we didn’t use too many organizational tools in our small project, but if implemented, they could be useful for the following reasons. The combination of effective organizational tools like JIRA, Confluence, and Slack, along with the foundational Scrum-Agile principles of iterative development, transparency, and collaboration, were instrumental in the success of our project. Each Scrum event, from sprint planning to retrospectives, leveraged these tools and principles to keep the team aligned, iterative, and focused on delivering value to the stakeholders at each individual sprint.
* **Evaluating Agile Process**: Assess the effectiveness of the Scrum-Agile approach for a specific project. Address each of the following:
  + Describe the pros and cons that the Scrum-Agile approach presented during the SNHU Travel project.
    - The Scrum-Agile approach allowed the SNHU Travel project team to adapt to changing requirements and stakeholder feedback. For example, during the project, a major change was requested to include a new feature for booking travel packages directly through the platform. The iterative nature of Scrum made it possible to incorporate this change without derailing the entire project timeline. If we were using the Waterfall method, a small change like choosing to target health and wellness travel opportunities could’ve derailed the whole project and pushed the deadline back to an unacceptable date.
  + Determine whether or not a Scrum-Agile approach was the best approach for the SNHU Travel development project.
    - The travel industry is fast-paced, and requirements can change based on market trends, customer feedback, and competitive offerings. Scrum-Agile’s iterative approach allowed the team to respond to these changes effectively. For example, when DriverPass wanted to switch to health and wellness-based travel, the team was able to easily modify, test, and implement the new request as the development effort was kept Agile and iterative. This allowed for major ethos changes to occur without affecting the core functionality of the program and the work that was already done.