**Final Project: Retrospective**

Craig Rowell

CS-250 Software Development Lifecycle Jun. 23. 2022

**Final Project: Retrospective**

**Roles**

Throughout this course I was tasked with taking on each role on a Scrum team: Scrum master, product owner, tester, and developer. Through this experience, it was apparent that the development process is dependent on all team members understanding each role’s purpose and perspective in order to efficiently and dynamically create a quality product. Communication and responsibility are paramount throughout the entire team when using Agile methodology in real-world software development.

***Scrum Master***

When learning how a Scrum team works, it is most beneficial to begin with the Scrum Master role. The overall collaboration of the team is reliant on the Scrum Master’s ability to motivate the rest of the team and provide direction through sprint planning, daily standups, backlog grooming and sprint retrospectives. These are essential organizational structures of development and the Scrum master’s purpose is to facilitate the agile team’s success by providing direction. When properly executed, the Scrum Master’s responsibilities foster team collaboration, efficiency, accountability, focus, and self-organization.

The purpose of Sprint planning and daily stand-up meetings are to discuss what tasks were completed yesterday, what features and bugs should take priority today and who is responsible for completing which tasks. Planning daily goals is more manageable for individuals when the whole team contributes their expertise. Communicating and collaborating reduce or removes roadblocks that could cause team members to get stuck on a task which may delay implementation of features or release of working software. Daily Stand-up meetings also help the team identify and resolve areas of conflict, enable the Scrum Master to curate the sprint and backlog as necessary to correct estimates with consideration for new information.

When assuming the role of Scrum Master, I was tasked with creating an agenda for the daily stand-up meeting.

I learned how to use this practice to develop a structure and flow for the meeting, which demonstrated how important daily stand-up meetings are to keep the team cohesive when starting a Sprint by providing everyone with status updates on which tasks are completed and which to work on next, which gets the day started off on the right foot. Communicating ensures that every software feature is implemented, tested, and documented, as well as holding all team members accountable for their work. Communication is valuable for every member of the development team, but it is most important for the Scrum Master who is responsible for arranging information exchange between all team members and to coordinate all user stories so they are clear to everyone.

***Product Owner***

The Product Owner’s foremost objective was maximizing the value of the project for stakeholders, by integrating their feedback into the product backlog for each epic to steer the direction of feature development. This establishes an overarching vision for development team, so they can better understand user and client perspectives and pivot accordingly. By curating the backlog, product owners make decisions for the project to prioritize the workload. As the intermediary between the client and the rest of the development team, it is the product owner’s responsibility to make sure goals are clear and the vision is aligned with business objectives. This job is pivotal to the success of any agile development project from start to finish. As a Product Owner, it is imperative to understand the distinction between providing direction and micromanaging, which is a delicate balance. The temptation to determine exactly how much work will be completed at each sprint is a major challenge for the Product Owner. Work closely with the Scrum Master can prevent a Product Owner from over-estimating the team’s capabilities, and pushing them beyond a realistic timeline by continuously expanding the scope of the project. It is essential to ensure the development team understands the client’s vision without impeding the collaboration that leads to an agile project’s success.

User stories help delegate tasks and prioritize the workload for the development team. To sreate a product backlog of user stories, the Product owner regularly provides clients and users with the latest enterprise or release implementation and supports a feedback loop between them and developers. Hearing feedback directly from users clarifies project goals and helps the team prioritize feature implementation. Asking development team members to communicate directly with users to understand their perspectives is a highly effective strategy that a Product Owner can use to be effective in their role. This leads the way for the Product Owner to create a cohesive backlog and user stories with a purpose that is familiar to the development team. When learning to create a product backlog and user stories myself, I discovered that visualizing the workload this way made the project much more organized which smoothed progress as I moved on to fill the other roles.

***Tester***

A tester on an Agile team, writes test requirements and test plans, then designs and executes test cases to discover bugs, determine whether the product is implemented as efficiently as possible, and to provide scalability, giving developers confidence in their ability to alter the software. This reduces the number of undetected bugs in released software. At the press of a button, a good test plan allows developers to detect and fix any issues that arise as a result of code changes made during the rest of the development phase, and into the maintenance phase. By frequently interacting with the development team, a tester ensures that everyone is in sync with the tests and that user stories pass muster.

Throughout the design process, it is the Tester’s responsibility to identify which project features are working as intended and which need more work before release. Tester failure is an expected part of the development process. When a test fails, it is an indication that the Tester is doing their job well, and they are encouraged to share what they learned from those failures with the team. User stories are used to write requirements that are referenced to develop a test plan, which in turn is used to write test cases which verify the requirements. Acceptance criteria determines whether a test plan passes or fails. In order to properly develop requirements and a test plan, the Tester should communicate with the rest of the team for clarifications when necessary. See the example email below for how a tester might effectively communicate:

To: Dennis (Product Owner/ Scrum Master), George (Lead Developer), Kris (Technical Lead), Suganya (Lead Tester), Loren (Independent), Bill (QA Auditor), Morrigan (Alternate Independent)

From: Craig

5/29/2022

Subject: Updating Our Test Procedure

Hey Team,

I would like to suggest a slight modification to our test procedure. Although it may seem like we’ll be doing more work up front, I believe this change will result in more refined code and better test methods, reducing the amount of revising the test team will need to do in the long run.

If we write requirements based off our user stories instead of writing test cases directly from them, our testers will have an easier time writing clear and concise test cases. Breaking up the process into two steps, interpreting user’s intent into requirements and developing a procedure to test features in test cases will result in both a clearer vision of user intent and better test procedures by creating a singular focus on each task, rather than trying to do both at once. I went ahead and created a list of sample requirements and a test case based on them for (current task) to show how this process will result in better unit testing (see attached). Please feel free to respond with your thoughts once you’ve had a chance to review my work.

Regards,

Craig Rowell

MFIOS-SQA Automation Engineer

DEKA Research & Development Corp.

***Developer***

Developers on an Agile Team play a pivotal role. On a waterfall team, which uses a predetermined plan-driven method, a developer may write code in isolation. On an Agile team their role is much more collaborative. Developers practicing test-driven development design and test software, sometimes pair programming with Testers. A Developer must communicate with the Product Owner as necessary to clarify requirements as they are accountable for the software they create. Working together and communicating effectively with the rest of the team, they need to be aware of any updates in the development process to efficiently update their software.

Making changes while developing code can be very stressful. When an Agile team works together to communicate changes, Developers will have a much easier time meeting deadlines. Any changes must be updated at every level from production code to product backlog to user stories and test cases. The developer will continue updating their code, editing their documentation simultaneously where necessary. Developer might need clarification from the Tester or Product Owner while making changes. To ensure they get a response necessary to move forward their questions about functionality and desired outcomes should be specific. To understand how to communicate effectively, see this example email :

To: Product Owner; Tester

Cc: Scrum Master; Other Software Team Members

Subject: Development Strategy Requests

Hi Product Owner and Tester,

I have completed the Top 5 Detox/ Wellness Vacations Slides task as requested, and it occurred to me that (Tester’s Name) should have been involved if we are planning to practice test-driven development. (Tester’s Name), I’d appreciate it if you’d be willing to try pair program with me on tasks going forward so that we can ensure that we’re both on the same page about what tests our code needs to pass and how we’re implementing new features. I think working together will prevent interruptions that would inevitably arise if we were to work on separate tasks.

(Product Owner’s Name), I wanted to run this by you as well, in case you had any questions about allocating two resources to a single task. Most experienced programmers agree that these agile practices are more efficient methods and will save us time in the long run. Also, when I was completing the detox/wellness vacation task SNHU Travel management requested, I noticed a couple features that could use improvement. The background and text colors on the slides make the text very difficult to read and the text size for the location titles on slides 2 through 5 is smaller than the text on slide 1. Will you please add a task to the backlog to fix text formatting on the Top 5 Detox/Wellness Vacations slides?

I have cc’d the entire team on this email, so please use reply all when you respond, that way we can keep everyone in the loop about how we’re planning to work together.

Thanks,

Craig

Due to its adaptable nature, Agile practices provide more flexibility to a Developer’s development approach. Proper estimation of time allotted for changes in features or other modifications the client or product owner requests is primarily the Developer’s responsibility. User stories are utilized as a conversation starter for the Developer to interact with the Product Owner, Testers and Users as necessary, particularly when significant changes are made to the project goals.

In real world applications there is always a balance between Agile practices and waterfall planning. “The typical discussion that compares agile and waterfall as if they were two discrete, mutually exclusive, binary choices oversimplifies what should be more accurately thought of as a range of adaptive and plan-driven approaches.” (Cobb 13) The roles I assumed and how I approached each task using Agile would not have been effective using the Waterfall method. There are benefits and drawback to using either method. The Agile method is a better choice when you anticipate changes in scope throughout the project, the product cannot be clearly defined, or when working software needs to be implemented quickly and new features will be added over time after initial release. Waterfall is recommended on projects that has a clearly defined, predictable, and immutable end product.

**References**

Cobb, Charles G. *The Project Manager's Guide to Mastering Agile: Principles and Practices for an Adaptive Approach*. John Wiley & Sons P&T, 2015-01-26. Vit.