Daniel Schween

04/18/2024

CS-250

Scrum Master: Retrospective and Sprint Review

I have been a project manager at ChadaTech for several years now. ChadaTech specializes in custom software design, and develops products using a traditional approach, such as the Waterfall Method. Typically, we make an initial plan with the Product Owner and senior management to publish the product within a fixed time frame and a fixed cost. However, this traditional method creates restrictions, discourages creativity, and doesn’t account for unsuspected problems. Traditionally, the Product Owner creates a development schedule deadline and fixed-price commitment based on what they thought was needed to deliver a final, functional product.

From my personal experience, projects don’t always go as planned; problems arise during testing, user stories change, and brute force development is used to meet deadlines, compromising the quality of the product. When this happens, “[ChadaTech] is pressured to make a fixed-price commitment to complete the project by the scheduled delivery date with performance penalties for missing the deadline based only on a very high-level understanding of the product requirements” (Cobb, 2023). As a result, ChadaTech had asked my team to deliver SNHU Travel an improved website and brand-new mobile app using agile methodology. ChadaTech wants to adopt a new method of development that embraces continuous integration, transparency, and openly accepts challenges without overcomplication. I assembled a Scrum Team to work on this product for SNHU Travel including Testers, Developers, a Product Owner, and myself as the Scrum Master.

I believe without fixed price restrictions and one-track minded planning, my team can focus on a quality deliverable and will resolve problems much earlier before snowballing into an unmanageable task. To embrace the new agile methodology, my superiors gave me permission to contract an agile coach who worked from the bottom up. The agile coach assisted my development team teaching them new agile practices such as sprint-planning and introduced agile tools like the information radiator he set up in the common room. Our developers and testers can view the information radiator daily and implement this information into their sprints that week. The agile coach also spent time with the Product Owner and senior management stressing the importance of their participation on the agile team. Now the Product Owner surveys focus groups to improve user stories and provide the agile team with the latest feedback from clients and stakeholders. The Scrum Team found this feedback very useful so the know their work is relevant.

As the Scrum Master, continuous integration from the Product Owner provides an opportunity for me to create or update sprints and prioritize stories. The developers and testers appreciated this a lot because they know their work is being prioritized and have the freedom to adjust plans without fear of penalty. The Scrum team now focuses on quality instead of pushing work through to meet deadlines which used to compromise their deliverable.

Agile development helped identify additional retrospectives – testing and developing. When the team was confined to the Waterfall Method, most testing was done in the later stages of the software development life cycle. Often, testing became nearly unmanageable since errors that occurred early in the product life cycle affected later increments of code published afterward. “[Waterfall] testing is a more reactive approach to finding and correcting defects and responsibility for the quality of the software is perceived as on a [third party] QA organization. Agile testing is a more proactive approach to preventing defects. In contrast, the agile team includes a dedicated tester, but the overall team shares responsibility for the quality of the software” (Cobb, 2023). The agile method embraces transparency and doesn’t view errors as failures. Instead, the Scrum Team works together embracing openness to fix the problem as soon as it is identified.

The Product Owner identified user stories the Scrum Team used to improve their product. Originally, the developers programed code for the website to showcase the top 5 destinations the general public found interesting. Additionally, the website was coded to display search results as a slide show, opposed to a list with headers and key information. The Product Owner also surveyed user feedback and found the user would like a mobile application. User stories are important because they guide the scrum team toward success and helps deliver the best product; however, initial feedback may not be the “end all, be all”. Stories change over time and affect the overall deliverable the Scrum Team is working on.

A traditional waterfall method wouldn’t consider this continuous integration and would proceed based off pre-determined planning, time, and budget. An Agile Scrum Team embraces change to deliver the best product for the user. The Product Owner consulted with other stakeholders and agreed user stories were important enough to implement into the design. After the Product Owners survey, changes were necessary to deliver a better product and the feedback was embedded into the team’s plan going forward. Agile methodology factors in scenarios as such, differentiating agile methodology from the Waterfall method; agile development embraces change and transparency for a better deliverable.

As the Scrum Team developers finish the latest sprint implementing the original user stories, I asked the tester to send the Product Owner an e-mail to clarify the user stories and be more specific before delivering the product. The Product Owner wound up specifying different details, adding more test cases. The Product owner has updated feedback and clarifies that the users want a top 10 destination list more specific to the individuals travel preferences, rather than a list based off popularity. Good thing the Tester inquired, because the team must also include functionality that allows the user to specify a price range and enable search results to be sortable by user rating. Finally, the users want a full fledge mobile app that includes every feature the website has. This is achievable but introduces a lot more test cases and story points for the developers.

The developers must make changes to existing code and add brand-new code to include this new functionality. I, as the Scrum Master, make immediate changes to the information radiator in the common room and update Microsoft Azure Board to communicate these changes to the entire Scrum Team. Azure Board also allows me to distribute the work accordingly, and update priorities, sprints and test cases. Azure Boards worked wonders communicating updates to the team; after all we embrace changes to our work if the result is delivering a better product.

Our Scrum team has reached the final sprint and is combing through the finer details with the tester. The development team finished their code and is now working with the tester with quality assurance. The major bugs have been fixed and The Scrum Team is testing the functionality, so the end user has a wonderful user-interface using SNHU Travel’s new software. The whole team sharing responsibility and changing roles is what makes agile development superior to the Waterfall Method. When we used to use the Waterfall Method, developers would wait around as the QA testing team finishes their work. In retrospect this was a huge waste of time and resources.

In Conclusion, the Agile Methodology was outstanding and seems superior to the Waterfall method. Daily stand-ups, weekly Scrum Events, our brand-new information radiator whiteboard, face-to-face communication, and software tools such as Azure Boards are all fantastic additions to our new philosophy. When we worked using the Waterfall Method, senior management focused on fixed-price planning and time constraints. The traditional method wasn’t as open to change and transparency, since the team had to deliver the product set within the initial boundaries. Often, time was wasted waiting for other teams to finish their portion of work, and other teams couldn’t move forward with progress.

Agile development and Sprint-planning avoided such redundancy and opened a door to continuous integration and constant development. Breaking the product down into smaller sprints ensured the Scrum team was always working on a piece of the product and testing was done during and after each sprint. Integrated testing saved a lot of time and unnecessary headaches that I have encountered before in my traditional experiences. For example, after the SNHU Travel product was complete, the Product Owner still approached the Scrum Team with a new focus on detox and wellness; SNHU Travel believes it will be the trending topic in 2024. The Scrum Team was able to handle this last-minute change using their new agile tools and scrum roles to successfully adjust the product and prioritize wellness and detox destinations.

**Citation:**

Cobb, C. G. (2023). *The Project Manager’s Guide to Mastering Agile: Principles and practices for an adaptive approach*. Wiley.