Matthew Powers

CS 250 – Software Development Lifecycle

Final Project – Sprint Retrospective

August 11, 2021

When analyzing the sprint, we must come to a single conclusion: it could have gone better. When we analyze each person’s input individually, we can understand why. Thankfully, the final product is not yet complete and the scrum framework allows us to work toward a better one than what has been put forth thus far. The change in direction did help with the final completion. Communication was a big part of that. Production of the website will reach proper completion at the due date thanks to the scrum-agile approach.

When I and the product owner met with the client, we were given a set of instructions and a basis for the website they wanted. Unfortunately, because the product owner did not respond to the focus group adequately, the product backlog suffered and development was not focused properly. Through the understanding acquired during the sprint review, the product owner can clean up the product backlog and the sprint backlog can move forward much more in line with what the client and end users would want.

My own lack of focus put us behind schedule making us work harder to meet the expected launch. Because I did not work with specificity during the construction of the charter, the rest of the team lacked focus during the rest of the scrum events. Due to the understanding that I did and through communication to the team about my shortcomings, the team can work more diligently on the rest of the construction of the website.

The testers were not immune to failure. They did not understand properly what to revise when the product owner forwarded an email regarding the adjusted use cases. Once they approached the product owner directly, the testers were able to refine a much better product.

Development did not go as well as it could have. The primary failing was in the product launch. Despite a proper series of tests performed by the testers on our team, the display during the sprint review did not execute correctly. We later found that it was the result of an improperly exported file. This denied them the possibility of launching that file when they were trying to show off the website in its current form.

Each of the members of the team had their own share of failures and shortcomings. Despite this, we can still meet the deadline expected by the client. This would not be possible outside of an agile methodology using the scrum framework. By receiving the feedback from the other members of the team as well as the client and end users of the site, we can truly refine our approach and deliver a better product. Not only that, but we can still meet the original goal of a five-week launch. Most of the team may have had to work a little extra, but it will still be worth it to meet the goal.

The change in direction was unexpected, to say the least. After the product owner met with the focus group, I thought our requirements were fairly cut and dry. Thankfully, through the level of communication we can see as part of scrum. The communication was able to be relayed to the rest of the team during the next daily scrum. Rather than receiving this information in the middle as we are working on an existing product during the waterfall model where we would have been on the testing stage, we received it during a sprint. As a direct result, not only was the team able to move forward with the changes, it did not set us back weeks or even days.

When we examine the communication in its own light, we can see exactly how it was reflected in the work done. One such example was a communique which was sent between the tester and the product owner. In it, he outlined what he believed the team should work on for a login system and how to make it as secure as possible. Another example of communication which was sent through the team was when the change in direction on the site happened. At first, nobody knew about it except the product owner. By the next day, not only did everyone know but they began work on it right away. In a typical waterfall model, this level of communication would not have worked quite as well.

The scrum version of agile worked quite well for us. When I think back to the principles that led us toward success, chief among them was “welcome changing requirements (Cobb, p. 26).” The feedback loop present in scrum allows us to do just that whether it is through emails in the middle of the day or at the sprint review at the end. If we do not allow for change to occur, then we will not be able to meet the expectations laid out by the client. Another such principle was, “Face to face conversation, (Cobb, p. 27).” Had the product owner not met with us during the daily scrum the following day when she knew we would all be together, I doubt the team would have adapted as fluidly and as quickly as they really did. For us to even get started on the project, we still needed to approach everything from the backlog refinement meeting. Through the breakdown of identified requirements, we could lean toward construction of the website for our client.

Were I to evaluate what scrum did for the project, I could honestly say that it did what the waterfall model did not. Through the analysis phase of the waterfall, which is comparable to the backlog and sprint refinement, changes to an existing system are laid out to be programmed as necessary. Only by starting a whole new waterfall, after completing the existing one, would the changes have been able to take place. This is why the launch would have been delayed by days or weeks, something which was not to the client’s wishes. The fluctuating nature of the client’s and end users’ goals allows for the agile methodology to achieve far greater results than those of simply using the waterfall method. It is not entirely perfect for this scenario, however. After reviewing some other options we had for the development of the site, I believe that scrumban would have been better by adding in a work in progress board. With the board, we would see an increased amount of collaboration that happened at each stage of development thereby expediting it even more. There is no doubt in my mind that communication and teamwork is necessary in systems deployment, but how we decide to go about it is what sets up each project for either success or failure.

References

Charles G. Cobb. (2015). *The Project Manager’s Guide to Mastering Agile: Principles and Practices for an Adaptive Approach*. Wiley