Matthew Cohen

The following review is a retrospective on a pilot for shifting development models at ChadaTech, a custom software design company for domestic and international clients, from its prior waterfall model to the agile approach. ChadaTech believed that such a shift would enhance their products and build a more cohesive corporate culture and in order to predict the efficacy of such a shift with greater accuracy they decided to test it out with a pilot team that I was a part of. Over the course of my team’s shift to agile we worked on an application for SNHU Travel and the following will draw from examples that occurred during said development.

Demonstrate how the various roles on your Scrum-agile Team specifically contributed to the success of the SNHU Travel project. Be sure to use specific examples from your experiences.

To begin with, our scrum team consisted of our Product Owner, Christy, our Scrum Master, Ron, our Developer, Nicky, and our Tester, Brian. The main goal of our application for SNHU Travel was to develop a vacation booking system for them which would leverage trendy and unique travel packages in order to enhance their reach and reputation. To this end, Christy, as product owner, was responsible for directing product vision and prioritizing backlog while also communicating with the client; Ron, as Scrum Master, was responsible for facilitating scrum practices on the team, such as daily stand up meetings, bi-weekly sprint planning, and sprint retrospectives; Nicky, as Developer, was responsible for writing the actual code which went into the application, she had to design and develop it so that it met the key project objectives such as an intuitive user interface, seamless booking process, and integration with existing systems; Finally, Brian, as Tester, was responsible for ensuring quality control, effectively making sure that everything that Nicky did worked right and looked good.

One of the key areas in which agile immediately began to prove itself in development was through the collection, distillment, and ultimate action taken of User Stories. User Stories are essentially direct feedback from some early customers of a product which are analyzed by the team in order to determine what should be changed about the product. In our case, Christy, our product owner, held a focus group with several real customers in order to collect their user stories which mostly related specific desires for the product, such as being able to access a list of top destinations or to exclude destinations based on specific criteria. Christy then analyzed said user stories by identifying the size and priority of what they were asking for and finally placed them into a Product Backlog. Brian, our Tester, then took over by creating test cases, identifying ambiguity in the user stories, developing a set of acceptance criteria, indicating pass or fail measures for each identified step that a user must take for their user story, and ultimately revising the test cases. Finally, Nicky, our Developer, actually implemented the needed changes to the software. Ron, our Scrum Master, served as a sort of glue for this aspect of development by essentially making sure that everyone was on the same page about what needed to be done since this process relied on all parts of the team to be in good communication with each other.

Another way in which agile was beneficial was when it became apparent that we would have to change the direction of the project a bit. At one point in development Christy met with SNHU Travel management and, in a subsequent meeting between her and the rest of the team, revealed that said management wants to shift the focus of the booking tool to detox and wellness travel due to their emergence as a growing travel sector. Agile was useful here in several ways; Firstly, agile promotes transparency and open communication between all team members which allowed everyone to be on the same page about what the client wanted, how they would accommodate that want, and what exactly every other team member was doing. Secondly, because agile is all about being iterative it wasn’t a major deal for the codebase and test cases to be updated. Finally, agile even allowed the team to maintain its same schedule and deadlines by way of Christy deprioritizing stories in the Product Backlog which were deemed no longer as important to the product.

As is evident by now, one of the cornerstones of agile development is good communication between all members of the team. For our team in this case we were consistent about Daily Stand-up meetings, which were daily meetings whereupon each team member would answer some similar questions, such as ‘What did I accomplish yesterday?”, “What will I do today?”, and “What obstacles are impeding my progress?”; these questions and sharing in this format encouraged frequent communication that focused on problem identification and made sure that everyone was on the same page about what everyone else was doing. Another good example of a communication process we favored as a part of agile was using a Kanban board, which used cards to visually represent our progress on different areas of development and, once again, helped to keep everyone on the same page about where development was at currently and where it was headed. Finally, we utilized Jira, an issue tracking and management program, in order to keep track of specific issues, requirements, and time constraints for specific parts of our product as well as keeping our codebase updated and sharable between team members. Each of these communication processes had an emphasis on keeping everyone aware of what everyone else on the team was doing and where development was headed and that degree of certainty in the present and future greatly enhanced the overall workflow by eliminating any uncertainty about what any team member should be working on.

Now that the development cycle has concluded we can determine with some clarity how effective agile was for this project. On the one hand agile made many things faster, easier, and less confusing; every step of the project was very well organized and, due to the openness and frequency of communication and the distribution of team members, scheduling, assigning tasks, solving problems, and even shifting project focus were all very straightforward problems that could be attacked directly before they became major. However, even given these advantages I don’t think that adopting an agile model was a good idea for a project of this size. To be clear, nearly all of the overarching concepts of agile, such as open communication between team members as well as between the team and the users and clients, an adaptable schedule with tasks of different priorities, and techniques for each team member that enabled them to develop iteratively and adaptably are all great ideas and should be standard in any project; The problem that I have with agile is simply that for a project and team of this size having a person on payroll whose job is essentially just to set up meetings and foster communication between team members seems completely superfluous. Perhaps for a project of several dozen or larger members working on a much larger product organization of meetings and agile communication like this does indeed require people dedicated solely to it, but for a team of this size I cannot justify it. In conclusion, most of agile should be adopted for the company as a whole, however, the need for employees dedicated solely to upholding it should be looked at more closely.