Chada Tech put together an agile team to work on building and refactoring code for their client, SNHU Travel. As part of their scrum team, and working as a scrum master, I oversaw the progression of code as it went from development to testing and back again during its evolution. This was our first foray into the heart of an agile workforce, and I think overall we did a pretty good job seeing our software production go from start to “done.”

We started out every day with a daily scrum, during which each member of our agile team answered three questions in turn: 1) What did you accomplish yesterday? 2) What will you be working on today? 3) What are your impediments? Daily scrums at Chada Tech have become a godsend to the production of our code and its evolution through refactoring. At our daily scrums, developers chime in about what is going smoothly as well as what hang-ups they may have encountered with the SNHU Travel app. The product owner tells his dealings with the clients, and puts in his two cents about what needs to be done to better serve the customers and the potential users of our app. Our testers weigh in on what bugs they have been able to reveal from within our working codebase. And lastly I as scrum master help to keep the daily scrum in motion by letting each team member in turn talk about their particular impediments and how we might be able to as a group overcome any obstacles hanging in the way of completing our sprint and getting our product to a state of being “done.”

Our product owner has been working hard with both our clients as well as with a number of potential users of our software in order to bring to the product backlog a series of user stories which I have been managing for our development team. As the user stories are prioritized, we then come together as a team to enact a vote on each new story in order to decide how many story points a particular user story should cost to get it to the state of being “done.” We do this by a method called, “planning poker,” where every member of our team holds onto a deck of playing cards and we go over the user story in question. When everyone understands the language of the user story, we vote how many story points we want as an estimate for how difficult this user story will be to complete.

As the development team finishes a user story, they pass it onto the testers who have been following the code throughout its life cycle and already have been running tests on it. The developers then pull a new user story off the product backlog when they have gotten the working user story to a state of being “done.” As scrum master, it is my duty not only to keep the developers encouraged and their progress moving forward, but I also am responsible for managing the user story stack for them, keeping the jobs with highest priority at the top of the product backlog. To this end, I attend all meetings between the clients and the product owner, as well as meetings between him and the selected end-users. All so I have a better understanding of what is needed in our software product so we wind up with a happy client.

In a waterfall model, we would have been planning the entirety of our software product before ever making a single keypress in development of our code. Agile has proven to be a much better system of development, because instead of planning the entirety of a design for our software up front, we rather place importance upon getting things working as fast as possible. Where in a waterfall model we would push our required coding principles upon the developers in a top-down approach, in agile the power of design lies instead in the hands of our developers. We want to stress creativity and innovation among our team members. This is why we try to remove labels as much as possible from those who would normally be called supervisors or managers from the team; instead we want to espouse an air of equality among our staff. Anyone who has an idea should have the right to be heard, and in fact much of what a beginner questions to the more experienced developers has been shown to bring those more experienced developers insights into the coding of our product.

The agile approach to development for our SNHU Travel app has been a good experiment with clear positive results. I think our next step is to approach the higher-ups in Chada Tech and let them see what it is we have accomplished in our finished product, but not only that. We also should be telling them about the lift in team morale through working in such an environment as this. The staff are excited about coming to work. They do not feel pressured to meet a deadline and they are not pulling all-nighters in order to complete a preset goal.

Of course, having an agile workforce in an otherwise waterfall-style bureaucracy will not be enough to shift the company towards a more agile-aware workforce. Implementing one or two agile cells still leaves much of the top-down power and control structure left within an organization. If we want to be true innovators going into the next stage of corporate growth, we should definitely look into freeing more of the developer-level jobs from the chains of a waterfall system and letting them explore the realms of creativity which loose them from the “daily grind.”

During the daily scrums, I often bring up at the beginning of our standup meetings what I had heard from my meetings with the product owner and our clients and potential users. I believe this helps keep our developers and testers focused on what the user needs, rather than what features seem nice to have but may not be essential or even wanted by our clients and their userbase.

Our stories have gone from rough sketches to finished prototypes abundantly over the course of our software development life cycle. This is thanks to all the team, but especially from the developers who had to read, comprehend, innovate, and execute what was desired (or at least what was needed) by the clients.

We communicate outside of the daily scrums by email and in-person chats. Particularly we send a lot of emails to the product owner and the testing team, explaining what is happening with our codebase and with words illustrating our pitfalls and asking for suggested workarounds. At least once a week I find myself engrossed in an email to the product owner, raising any concerns I may have about our code and what it will take to get it to a state of “done,” within the timeframe of our sprint.

One reason I found agile a much better approach for my team than any waterfall project I have previously worked on is that when the needs of the users and the client clashed with what our code was doing, it was easy to let go of the current user story’s progression and start over from scratch if need be. Under a waterfall-style system of development, such an abandonment of code would have become disastrous for us as a development team. With waterfall, we would not only have to abandon the current code, but may have to even go back up the chain and start a much larger section of our product over from scratch. Under agile, it is easy; at most we lose the work of a single user story while we shift from doing one thing to another.

In sum, I think an agile approach to software development for SNHU Travel through Chada Tech was a delightful success. We addressed the needs of the client and its potential users. But not only that. As a team of developers we enjoyed our work and never once felt pressured to meet a deadline. Now, we look forward towards the future of Chada Tech and would like to see the continuation and possible evolution of agile principles and philosophy used throughout the company.

Mark Rossmiller

Scrum Master