From Junior Engineer to Scrum Master in less than a year

3 December 2015 | **Insight Zuhlke, Software Engineering** | **Stefan Benz**, **Arnaud L'Hôte**
Reading time: 6 minutes

Nearly two years ago after graduating from University, I started as a junior engineer in a project with over 40 engineers. I joined a team of seven developers with different seniority levels and an experienced Scrum Master. In the project setup, a developer usually stays for two years. Therefore, the team looks very different today. Some engineers left and new ones joined, but working in that team is still really great fun. In this blog, I want to tell you what I've learned in my first two years at work after university and what process I went through.

First steps as developer

In my first half year, I was a regular developer, learning the art of professional software development and realizing it is quite different from University, but no less fascinating. For the most part, I learned that software development includes way more than just writing code. I would estimate that I didn't usually code more than four hours a day. The other half of the day I was dealing with stakeholders of all kinds, team activities, working on requirements or, due to my flair for numbers, calculating and visualizing the development progress in our team. That's where I also learned that the success or failure of a project is rarely decided by technical Expertise, but rather by human aspects (team-work, communication, collaboration, feedback, etc.).

Transition to Scrum Master

After half a year, my team moved to a different location. The setup at that location consists of three teams, each with its Scrum Master, two experienced coaches who coach the Scrum Masters and help them out if needed and a solution manager who has the commercial responsibility. Our Scrum Master was about to leave our team and take over one of that more general coaching role.

My team needed a new Scrum Master. I had immediate interest. My motivation was manysided. While I still consider coding to be a lot of fun, I particularly loved that, even though a Scrum Master role in my Project was not a traditional Scrum Master role according to the Scrum guide, it included aspects outside of regular coding that made my working day even more varied. For instance, the Scrum Master in my project is also considered a team leader. The team leader has the responsibility that the team delivers the different software packages in time, in budget and with the required quality, but the role also includes human aspects such as making sure that the team members stay happy and motivated.

As we had engineers in our team with much more experience, I was not sure whether I would be accepted as new Scrum Master and team leader, so I decided to *talk* to them first. My fear was unfounded though. Some didn't want that role while others were happy with the situation they were in.

First steps as Scrum Master

The first few months as Scrum Master went well. We already had a good working team and we didn't face any troubles, so I was really happy in my new role even though I couldn't code as much as I used to. But that was about to change slightly.

Around New Year, some developers left and new developers joined. Our team went through a transition phase. Suddenly, I had to face situations where we had developers which were unhappy with their work. My approach was again to just *talk* to them and find out what they did't like about the current work and what would help improve it. We could find a solution. The unhappy developer was put in charge as SPOC (*single point of contact*) for a rather challenging project. The developer was extremely reliable and exceled in that SPOC role. The happy curve went up again. While it is important to have T-shaped engineers, I realized that everyone benefits if people are given the work they love. Usually the consequence is that they excel in doing that work.

Our team faced another challenge when we delivered a rather high software cost estimate for a particular project by email. The receiver of that estimate was rather irritated by it and escalated the situation to higher levels. I was unsure how to handle this situation. But one of the most important things I have learned so far in my career is to ask for help if you need it, to learn from those who have more experience. That's exactly what I did. I went to my old Scrum Master (now working as a coach) and asked him how we should handle the situation. The solution was simple again: *talk* to the people involved. So we set up a meeting on the same evening and were able to de-escalate the situation. It turned out that it was just a misunderstanding of a requirement and after some clarification we were able to lower the estimate. That was a key event for me. I have learned a simple rule the hard way. Always *talk* to people face-to-face, especially if you want to communicate things that could be misunderstood, in order to get immediate feedback, to keep the feedback cycle short. If that's not possible, take your phone. Only if that is not possible either, write an email, but who doesn't have a phone nowadays?

And today?

More than half a year later, our team has changed again, but it is even more fun working in that team. Our team forms a strong bond. We have developed a culture of trust, meet regularly outside working hours and some of us even go to holiday together. We are known as the *crazy team* as we joke a lot and have fun whenever we can, but I consider it the key to success: a well-oiled team, consisting of engineers with different strengths (and also weaknesses) that help each other and make up more than just the sum of its members.

To summarize the key learnings

- Success is rarely decided by technical expertise but rather by human aspects
- Give people the work they love and they will usually excel doing that
- Ask for help if you need it
- Prefer face-to-face communication over writing emails
- A well-oiled team is the key for success

What experiences have you had with your team? What did you learn in your first steps as Scrum Master? Tell me!