Navigating Hybrid Scrum Environments

Understanding the Essentials, Avoiding the Pitfalls

—

Frederik M. Fowler

Apress®

CHAPTER

7

The Scrum Master

The third role defined by the Scrum Framework is perhaps the most misunderstood one of all. The Scrum Master role is neither a "business" role nor a "technical" role. The day-to-day work of a Scrum Master is hard to describe in terms of specific tasks to be done. When asked what a Scrum Master does on a day-to-day basis, the proper answer is: It depends. At best, the detailed work of a Scrum Master is vaguely defined and hard to pin down. This causes some organizations to attempt to use the Scrum Framework without the benefit of qualified Scrum Masters, usually with less-than-desirable results.

Once again, the role of Scrum Master on a Scrum Team is defined in terms of *responsibilities* as opposed to tasks. The Scrum Master has significant responsibilities that no one else bears. Addressing those responsibilities can take many shapes and forms. The Scrum Master adds value by accepting and acting on those responsibilities. The value is usually easy to see and measure when comparing the results of a team with an effective Scrum Master with those of a team with no Scrum Master.

The word *master* in the name Scrum Master implies a kind of authority. Some people think a Scrum Master is some kind of task master, someone with a whip who drives slaves to work hard.

In fact, a Scrum Master is the master of *no one*. A Scrum Master is a *master of the Scrum Framework*. A Scrum Master is a master of Scrum in the same way that a Kung Fu master is a master of the art of Kung Fu.

F. M. Fowler, *Navigating Hybrid Scrum Environments*,
https://doi.org/10.1007/978-1-4842-4164-6_7

Like a Kung Fu master, the role of a Scrum Master is to be a teacher, mentor, and coach. The Scrum Master teaches and coaches the Product Owner and Development Team how to use the Scrum Framework properly. More specifically, the Scrum Master's responsibility is to make sure the Product Owner and the Development Team understand and live up to *their* responsibilities.

To function properly, a Scrum Team must have a Product Owner who focuses on maximizing value and a Development Team that focuses on technical execution. Both must work together on the basis of mutual respect and trust. In short, the Scrum Team must truly function as a high-performance team.

The Scrum Master is responsible for the Scrum Team's ability to function as a team. The Scrum Master owns the team's "teamwork." Just as a coach in the world of sports teaches players how to play together effectively, a Scrum Master teaches developers and Product Owners how to work together effectively.

Product Owners tend to be people in positions of power within a company, and they make decisions involving large amounts of the company's resources. They accept considerable responsibility and are held accountable for results. For this reason, there is a natural tendency for these people to become "bosses" and to give orders about what is to be done.

Developers tend to be individual contributors who feel they have little or no power within an organization. They tend to play a passive role and often resort to passive-aggressive tactics when confronted with unpleasant situations. For these reasons, there is a natural tendency for these people to become "peons" and to ask to be told what to do.

The Scrum Master's main work is to teach and coach everyone to combat these natural tendencies. The Scrum Master teaches the developers to take an active role, set expectations, and live up to them. The Scrum Master teaches the Product Owner to set priorities, negotiate with the developers to get them addressed, and then trust the developers to live up to their end of the negotiated bargain.

When qualified and able people play their proper roles on a Scrum Team, the results can be spectacular. Unfortunately, there are often organizational, cultural, and personality-based pressures that prevent people from playing these roles correctly.

The Scrum Master's job is to teach and coach the team to resist these pressures. The specific challenges faced by the Scrum Master change every day. The goal, however, remains the same: to take a group of professional individuals and help them to form an identity in which the whole is greater than the sum of its parts.

Teaching People to Solve Their Own Problems

The roles on a Scrum Team can be compared to the roles on a professional sports team, such as an NBA basketball or NHL hockey team. On such a team, the players are seen as similar to the developers, and the team owner is recognized as playing the Product Owner role. The analog for the Scrum Master is the head coach on the sidelines.

Head coaches do not play the game, nor do they make financial decisions. The know their players know how to play their various positions (otherwise, they would not be on the team at all.) They know the team owner will keep the finances in order. The head coach job is to (gently) keep the owner from interfering with the players, and to get the players to work together effectively. Needless to say, it is a full-time job, and no professional sports team operates without a coach.

The Scrum Master's job is quite similar to that of a sports coach, but in this case, to keep the Product Owner from interfering with the developers, and to get the developers to work together effectively. It is also a full-time job, and no professional Scrum Team should operate without a Scrum Master.

Just as a professional basketball coach watches the action from the sidelines, a Scrum Master watches the team without becoming part of the work being done. The Scrum Master's job is to get everyone else to do the work necessary to create the product. Another way of putting it is: The Scrum Master's job is to make sure team members do not have an excuse for not fulfilling their responsibility.

Scrum Masters must make sure the Development Team owns every part of delivering the product features prioritized by the Product Owner. Scrum Masters must also make sure the Product Owner owns every part of the product's content and value. Scrum Masters can give advice and help out with details if asked, but in the end they must make sure all technical decisions are made by the Development Team and all business decisions are made by the Product Owner.

Servant Leadership

The Scrum Guide states that a Scrum Master is a "servant leader" for the team. Servant leadership is a slippery concept that confuses many people. After all, how can a servant lead the way?

The best way that I have found to describe servant leadership is to point to a historical example of a very effective servant leader. If the role played by this man is considered and understood, the concept of servant leadership becomes clear.

This man's name was Tenzing Norgay. He was a Sherpa and lived in the Himalayan Mountains of Nepal. In 1953, a British mountain climber named Edmund Hillary arrived in Norgay's village and hired him to lead the way up the slopes of Mount Everest—the tallest mountain in the world.

Hillary and Norgay set out in late spring and made their way up the mountain. Norgay led Hillary up the mountain paths to the ice fields, helped Hillary climb up through the ice and snow, helped him cross crevasses, kept him from falling off several cliffs, and got him to the lower edge of the north face of the mountain. He then helped Hillary climb up the final rocky slopes, cope with the fierce winds, and approach the summit. And, by the way, Norgay carried most of the luggage as well.

No one alive knows what happened when Hillary and Norgay reached the very top of the summit, but rumor has it that Norgay stepped up to the top and then extended a helping hand to Edmund Hillary, who thus became *the first man ever to climb Mount Everest*!

Norgay then did something for Hillary that is arguably more important than getting him to the top of Everest. Norgay got Hillary back down again. He got Hillary back down the north face, back down through the ice fields, back over the crevasses, back down the mountain paths, and finally back to the village where they had started.

This was very significant because, many years earlier, a pair of very experienced climbers had attempted to climb the mountain. They were George Mallory and Andrew Irvine, and they may well have reached the top. No one knows, however, because they never came down again. Mallory's remains were discovered on the mountain by American climber Conrad Anker in 1999.

After Hillary and Norgay reached the village, they said their farewells. Hillary went back to fame and fortune in Europe and America, and Norgay went back to village life. Eventually he made a kind of business out of guiding people up the mountain. He died in May 1986 at the age of 71.

A servant leader's job is similar to that of Tenzing Norgay. The servant leader helps others achieve their goals, keeps them from "falling off the cliffs," keeps them safe, but enables them to do great things. Servant leaders do not take credit for success. Their reward is to enable the success of others.

Scrum Masters would do well to remember the example of Tenzing Norgay and focus on enabling their teams to be successful within the Scrum Framework.

Impediments

The Scrum Guide does list one specific duty the Scrum master must perform. The Scrum master must remove impediments to the development team's progress. This "impediments" requirement is easily misunderstood and

often gets new Scrum Masters in trouble. It is very important to understand *what constitutes an impediment*. Many "blockers" that inexperienced Scrum Masters spend time and effort solving are not impediments at all.

Scrum Masters are teachers and coaches. As such, their job is to *teach and coach people to solve their own problems*. Any problem Scrum Masters solve for the team becomes their responsibility to solve whenever it occurs in the future. If they cannot solve it, the developers have a perfect excuse not to deliver the results they agreed to deliver. "We couldn't get it done because the Scrum Master didn't solve our problem."

The Scrum Master must teach and coach the team to solve its *own* problems whenever and wherever possible. Having said that, there are instances when teaching and coaching the team to solve a problem is *not* possible. There are some problems that are too difficult for the team to solve by itself. That is where the Scrum Master must step in.

An impediment is a problem that team members cannot be expected to solve by themselves. Only if it is beyond the capability of the team to resolve should Scrum Masters take ownership and solve it.

Inexperienced Scrum Masters allow teams to saddle them with piles of miscellaneous work in the name of removing impediments. Team members say, "Scrum Master, I can't move forward until I talk with the lawyers. Please arrange a meeting with them for me." They may also say, "Please book a room for us to talk over a design issue" or "Please look into getting us an upgraded version of our software tools." They may go so far as to say, "Please get me a faster workstation" or even "Please get the lights over my desk fixed."

Scrum Masters must exercise judgment with regard to whether a problem can be solved by the team. The team's ability to deal with problems changes over time. If a team is new and its members don't know much about how to request software upgrades, Scrum Masters may choose to do it for them, but *show them how to do it.* Later, when asked to get further upgrades, Scrum Masters might say, "Why should I do that? I showed you how to do it months ago. Please do it yourself!"

Scrum Masters must push back any problems to the team that, in the Scrum master's opinion, can be addressed by the team or its members. The team should arrange its own meetings, book its own rooms, get its own software upgrades, get its own workstations upgraded, and get its own lights fixed. The team must own all aspects of delivering the product, and this includes all of the miscellaneous tasks that make the delivery possible.

Having said that, there is never a time when true impediments cannot occur. Sometimes Scrum Masters must step in when it is clear the team cannot solve a problem itself. These issues tend to be the nastiest, dirtiest, most unpleasant situations that, unfortunately, can and do occur.

During my career I've had developers come to me and say, "I don't know what to do. I saw Suzy going through Brenda's purse." I've also had people say, "Mark is taking drugs all the time." The issue that always causes the most action is when someone—for example, Carol—says something like, "Pete won't keep his hands off me. I've told him to leave me alone but he keeps following me around."

Some issues go beyond the team's boundaries and raise concerns that the human resources department or even local authorities must deal with. In these situations, Scrum Masters must step up and intervene. They must do whatever is necessary to protect the company and the team, and to help keep the team's focus on delivering the product.

Summary

The role of Scrum Master is crucial for a team to reap the benefits of the Scrum Framework. The effectiveness of Scrum Masters is reflected in the quality of the teams they coach, and can be measured objectively.

If Product Owners are ineffective, that ineffectiveness can be measured by the lack of value produced by the team. If the team works hard, creates features, and delivers working software regularly, there is an ineffective Product Owner if the software turns out to have little value.

If Development Teams are ineffective, that ineffectiveness can be measured by the lack of working software produced. If they work hard but can't seem to get anything done, it may be time to reassess team members and create a more effective Development Team.

If Scrum Masters are ineffective, that ineffectiveness can be measured by a team's inability to work together. If the developers are always squabbling among themselves and the Product owner resorts to barking orders at them, it may be time to get a more effective Scrum Master to fix the situation.

A good Scrum Master's contribution is often hard to see, because when Scrum Masters are effective, everything works as it is supposed to. It is only when Scrum Masters are ineffective or absent that their real value becomes apparent.