Navigating Hybrid Scrum Environments

Understanding the Essentials, Avoiding the Pitfalls

—

Frederik M. Fowler

Apress®

CHAPTER

5

Scrum Team Roles

The Product Owner

The Scrum Framework organizes people by defining three distinct sets of responsibilities. Each set of responsibilities is called a *role*, and individuals play different roles when they participate in Scrum teams.

It is important to realize that these roles do not necessarily correspond to actual individuals. It is quite possible for people to play more than one role. It is not uncommon, for instance, for a developer also to play the role of Scrum Master (provided they have the necessary skills and experience to do so). Roles are sets of responsibilities; so, if an individual is capable of fulfilling more than one set, that's fine.

The three roles in the Scrum Framework address the fact that products always have both business and technical dimensions. There is a role—the Product Owner—that takes responsibility for all the business decisions that must be made. There is a second role—the Development Team—that takes responsibility for all of the technical decisions that must be made.

Finally, there is a third role—the Scrum Master—that takes responsibility for making sure the other team members understand and live up to their responsibilities. The Scrum Master makes sure the team uses the Scrum Framework properly.

F. M. Fowler, *Navigating Hybrid Scrum Environments*,
https://doi.org/10.1007/978-1-4842-4164-6_5

The Product Owner

The first role to consider is that of the Product Owner The Scrum Framework's Product Owner role involves guiding the team so the value of the product delivered is the highest possible for the team to produce. In the words of *The Scrum Guide*, the Product Owner's role is to maximize the business value of the product delivered by the team.

The role of Product Owner is *not* a technical role. Although knowledge of a product's underlying technology can be helpful, Product Owners are not responsible for any technical decisions regarding the product.

Product Owners are businesspeople. They must have expertise in understanding the marketplace in which the product will compete. Their job is to make sure the developers are always working on those parts of the product that will be of most value to the organization when they are completed.

In many ways, Product Owners function as investors. Product Owners invest the company's time and money in a team of developers. The goal of Product Owners is to earn a profit on that investment. They earn that return by guiding the developers to deliver valuable additions or changes to the product. The relative success of a Scrum Team Product Owner can be measured by the value of the product in the marketplace.

As stated in Chapter 1, the Scrum Framework focuses on products rather than projects. The reason for this is that the value of products can be measured directly in the marketplace, whereas the value of projects cannot.

Although one Product Owner can own many products, each product must be owned by one and only one Product Owner. That Product Owner must accept accountability for the value of the product.

Product Owner Characteristics

The role of Product Owner involves a considerable amount of responsibility. The individual who plays this role is measured on a "profit and loss" basis and makes decisions about how to spend significant amounts of the organization's resources. This role is usually filled by someone who has the respect and trust of the highest levels of the organization's management team.

Much of the day-to-day work of Product Owners involves negotiation. They negotiate with upper management to get the resources necessary to put together the Scrum Team. They negotiate with customers and stakeholders to identify and prioritize the development of product features. They negotiate with the Development Team to reach agreement about the features and functions that will be developed during each sprint. Product Owners should be people who are skilled at negotiating.

One more crucial characteristic of good Product Owners is their ability to make decisions. Product Owners make choices every day and they have to make them quickly and efficiently. After a decision has been made, it should "stick" and not be revisited lightly.

About ten years ago I was involved in a project to create an e-commerce web site for a distribution company. I was lucky enough to have the perfect Product Owner for the initiative.

The company was very old-fashioned (even for the business climate of 2007). It produced a thick catalog containing all 14,000 or so products it carried. Once a year it would print and mail a copy of this catalog to every one of their 20,000 or so customers. The customers would then leaf through the pages until they found the items they wanted.

After the customers had made a list of their desired items, they would call a toll-free telephone number printed on the cover of the catalog. They would be connected to a customer service person in a call center who would take their order over the phone. The customer service person would then key in the ordered items on an old-fashioned "green screen" terminal and into the mainframe-based order entry system.

This way of taking orders was both expensive and error prone. The costs of printing and mailing the catalog were considerable. There were also accuracy problems caused by taking orders orally ("I asked for ten 20-inch brackets, but you sent me 20 ten-inch brackets."). Something had to be done about the way the company did its business.

That "something" turned out to be the creation of an e-commerce web site. The goal was to create an "electronic catalog" in which customers could search through the company's products and load up a "shopping cart" of the items they wanted. They would then "check out" with their shopping cart, which would feed the orders directly into the mainframe-based order entry system.

As I said, the Product Owner for this "electronic catalog" was the perfect person for the job. He was the chief operating officer (COO) of the organization. As such, he had three very valuable traits:

1. He understood the business at a very fundamental level. He had been with the company for the better part of 30 years and knew just about everything there was to know about it, its products, and its customers.
2. He knew what he wanted. He understood the problems and drawbacks of the existing telephone-based system and he wanted those problems to go away. He also knew what kind of people the customers were and the amount of change they could accept.

3. He could make decisions that no one would question. As the COO, he was the no. 2 ranking person in the entire company and had the authority to shape its operations as he saw fit. If a developer asked him a question about how some particular feature should work, he would give an answer on the spot and that would be that.

Serving on a Scrum Team with a bunch of technical people turned out to be very appealing to the COO. He had a great interest in how technology could help the business, but he also had very naive ideas about what modern technology could and could not do. Working with the developers on the team gave him a way to explore some of his ideas and learn both the capabilities and the limitations of the web site. In the end, he learned to make business decisions while taking technical realities into account.

Having the COO on the team was equally beneficial for the technical people. Their interaction taught the developers about the business and its customers. Among other things, the COO taught the developers that the company's customers were even more old-fashioned than the company itself. Many of the customers were still exploring the use of the "modern wonder" called a *fax machine*. The developers learned the e-commerce web site had to be very simple and needed extensive step-by-step on-screen instructions for how to use it. In the end, the developers learned how to make technical decisions while taking business realities into account.

The fact that the e-commerce web site had the COO as its Product Owner produced wonderful results. Within six months of going live, the web site was doing 30% of the company's business. The number of incorrect shipments and returns dropped, order accuracy improved, and customer satisfaction rose as well. It was a success by any measure.

Common Product Ownership Dysfunctions

Although that COO was the best Product Owner I ever worked with, there have been many I've coached throughout the years who have had trouble filling their role. The dysfunctions I've encountered tend to fall into a few basic categories.

The Product Owner as PM

Quite often Product Owners feel quite a bit of pressure to "deliver" and can then fall into a trap. The trap is to become the de facto manager of the Development Team.

In the Scrum Framework, developers accept accountability for the delivery of the product and then take responsibility for organizing themselves to do so. If they can get someone else to organize them, the developers can then avoid any accountability for what they are doing. They can then say, "Just tell us what you want us to do and we'll do our best." As noted earlier, this statement really boils down to a complete abdication of responsibility on the part of the developers.

I've consulted with clients whose Product Owners have fallen into this trap. One of them had Product Owners who had been "taught" by the developers to write very detailed requirements, including details of both *what* to do and *how* to do it. When I asked one of the Product Owners why his requirements included actual sample code for the developers to use, his answer was, "I have to do it. Whenever the developers do not deliver what I want, the excuse is always 'you didn't put that in the requirements.'"

The most egregious example of a Product Owner acting as a PM involved adding an "e-mail address" column to an existing screen that already contained columns for name and phone number. The developers added a third column, but it was not wide enough to contain a full e-mail address without cutting off the last few characters it contained. Product Owner had asked for the column but had neglected to specify that the column be wide enough to accommodate e-mail addresses.

The developers had done what the Product Owner told them to do, and they had done "their best." The Product Owner let them get away with it by starting to tell them what to do. The Product Owner gave them tasks to do rather than problems to solve. The results were laughable and of little value to anyone.

The Product Owner "Committee"

Another common dysfunction is to have Product Owners who cannot make decisions without consulting others. The easiest way to find out if this is the case is to ask Product Owners to make a decision. If they actually make a decision, then all is well. If their answer to the question is "Let me go check," then that person is a messenger, not a Product Owner.

Product Owners have to negotiate constantly. They negotiate with developers, stakeholders, clients, and third-party partners. Negotiators have to be able to make decisions and stand by them. If negotiators cannot say "yes" to a proposal, then negotiating with them is a waste of time.

One client with whom I've worked in the past had one so called Product Owner for each Scrum team. These individuals were business analysts whose job was to write technical specifications from business requirement documents (BRDs). They had no ability to make any decisions at all.

When asked for clarifications about requirements, all these people could do was to repeat what was documented in the BRDs in the first place. It was futile for developers to suggest different approaches or alternative interpretations. After a while, they gave up.

Another one of my clients had a number of otherwise-capable Product Owners whose decisions were being overridden constantly by upper level management. The C-level people assumed that being "Agile" meant priorities could be changed instantly with no advance notification to anyone. The Product Owners had become frustrated because every time they tried to make strategic decisions about their products, their plans were overturned by some new "corporate initiative." In the end, they stopped making any kind of strategic plan at all. The resigned themselves to bend with every new gust of wind blowing down from the C level. They developed no long-term vision for their products at all.

A Product Owner is one person who is personally accountable for the value of the product. They must provide a single, clear voice, and identify opportunities and priorities for the team to pursue.

The Product Owner Without a Product

Many organizations make the mistake of organizing Scrum Teams around "components" and then assigning each team its own Product Owner. The Product Owner then sets the priorities for the team and controls the growth and development of the component.

The trouble with this approach is that it is very difficult to measure the business value of a component. Surely the component is valuable, but *how valuable*? How can Product Owners measure the relative value of one enhancement over another? How can Product Owners calculate whether they are earning a return for the company on the dollars being invested in the team?

The lack of clarity about what *value* means for a component leads to priority conflicts. For instance, if a Product Owner is asked by a customer to add a new feature to a front-end component, they must negotiate with the Product Owner of the back-end component. The front end cannot produce the necessary API needed to support the new feature by itself. The Product Owner for the front end cannot set priorities for the back end team. The front-end Product Owner has to hope and pray that the back-end Product Owner agrees that the new feature should be produced. Otherwise, nothing will get done.

Conflicts such as these can be solved easily if it is possible to measure the value of the team's work in dollars and cents. It is only possible to do this if there is a way to measure the value of the product itself in dollars and cents.

For example, it is easy to measure the value of a discrete product such as a smartphone-based game. If the game is available from the Google Play Store for $1.99 and 1,000,000 people download it, then the value of the game is $1,990,000. If some enhancement causes another 500,000 people to download it, that enhancement is worth $1.99 × 500,000, or $995,000.

In the component-level example, neither the value of the front end nor the value of the back end can be measured in dollars and cents. Only the *combination* of the front-end and back-end components can produce measureable value in the marketplace.

Rather than having one Product Owner for the front end and one for the back end, it should be recognized that the product is the *combination* of the front end and the back end. Each component can have its own team of developers, but there should only be *one* Product Owner setting priorities and maximizing value.

Product Owner Commitments

Product Owners tend to be busy people, especially if they are placed highly in an organization. They can have multiple demands on their time.

Despite this, and as *The Scrum Guide* makes clear, Product Owners are members of a Scrum Team. They must be a "team player" with the others and must work with them on the basis of mutual respect and trust.

If a highly placed executive consents to play the Product Owner role for a team, that person must commit to be available. The Agile Manifesto makes clear that business people and technical people must work together on a daily basis. Product Owners must be available on a daily basis to interact with the developers, answer their questions, and teach them about the business purpose their work is serving.

One of the reasons that some efforts to adopt the Scrum Framework fail is that new Product Owners misunderstand the demands of the role. If they are placed highly in the company (as, ideally, they should be), their day-to-day activities may change dramatically,

Product ownership is not a role that can be played by writing memos. Interacting with developers is key. As the Agile Manifesto states, the best way to convey information within a team is face-to-face. Product Owners must commit enough time in their schedule to interact directly with their Scrum Team to make sure all business priorities, concerns, and realities are understood by everyone involved.

Summary

As *The Scrum Guide* states, Product Owners' responsibilities are to maximize the value of the work performed by the Scrum Team. They do this by bringing a business perspective to the team to complement the technical abilities of the developers. Commercial software is always created to serve a business purpose. Product Owners take responsibility for articulating that purpose and guiding the team to serve it as best it can.