Software Development - The Product Owner

When one is doing agile type product development, there can be infinite variation in how a team approaches the project. However, these process execution variations usually fall within a well-defined band containing certain elements that are deemed necessary for success. One common category of agile development variation is called Scrum.

Scrum derives its name from rugby, a sport in which members from both teams bunch up around the ball in a sort of reverse tug-o-war. Whereas in rugby, members of the scrum may be trying to move the ball in different directions, in the agile version, there is only one team and they are trying to move the ball in the SAME direction.

It is important to note that even within the Scrum approach to agile development, there are many variations. The most common is to have two teams with one contained within the other. The project team consists of a Scrum Master, Product Owner, Stakeholders, and a Development Team. The Development Team consists of all the technical people required to design and build the product.

If the product is software, the Development Team will need some developers, one or more business analysts and often one user experience (UX) professional. There could be other people on the team depending on the project. For example, there could be a database administrator if the project is database intensive, or there could be an electrical engineer if the product involves interfacing with hardware.

A Quick Review
If you’ve read the other Columns in this series, you will already know that Scrum (an agile practice) differs from traditional approaches to product development in that it is highly iterative. The fundamental principle that underlies this approach is that it is usually extremely difficult to fully understand a functional domain based on pure research and dialog. The best way to understand a domain is to experience it.

Unfortunately, experience takes time. Traditional approaches, the most common of which is called Waterfall, required analysts to study the problem domain and develop a complete design for the product. This design was typically completed prior to presenting it to the project’s sponsor for approval. If you’ve ever written a book review, you know how difficult it is to provide a thorough analysis of a large document. Inevitably, errors and shortcomings are missed.
Scrum sets out to mitigate this problem by NOT attempting to design the product up front. Analysts must still perform research and gain understanding of the problem domain, but rather than write a large design document, the Development Team sets out creating a functional product early on. Not all products can be built incrementally, but software can and there are others too.

**Enter the Product Owner**
In order to develop products in an iterative manner, the Development Team needs a strong guide who is completely familiar with the problem domain. The basic premise behind this role is that the Product Owner can:

- Prioritize required functionality so that the team is always working on the most important things
- Answer problem domain questions when the Development Team needs guidance
- Validate newly added functionality and/or recommend changes to the functionality in order to bring it into alignment with the requirements of the process it supports

As you can imagine, this represents a vast departure from the traditional relationship between the business sponsor of a product. Whereas the traditional business sponsor received periodic updates of the project’s status and a large final review (possibly with a few intermediate reviews), the Scrum product owner, must be an integrated part of the project team.

**The Challenge**
The Product Owner plays a critical role in the success of the project. So, anything that prevents their successful involvement will likely have a negative impact on the product. Let’s take a look at some typical problems with Product Owners:

- **The Absent Product Owner** - assuming that the Product Owner understands the problem domain in great detail, it is quite likely that this knowledge was gained through their intimate involvement in business operations. Quite frankly, the best product owners are usually the busiest people around. Asking them to add extensive involvement in a lengthy project is asking a lot! When it comes time to choose between supporting product development and keeping their business running...I think you know which they’ll choose.

- **The Intermediary Product Owner** - another result of selecting the wrong Product Owner is that the designated person really isn’t capable of making good decisions, but unlike the Underpowered Product Owner, the Intermediary doesn’t attempt decisions and answers that could be wrong. Instead, they simply go get the information they need as they go along. Needless to say, this slows things down tremendously. Scrum teams must be highly interactive. When a question is asked, the answer often spawns more questions. If the embedded individual cannot answer these questions, the process is delayed...and delayed...and delayed...

- **The Agenda-Based Product Owner** - maybe the most insidious of the problematic Product Owners is the one that has his/her own agenda that does not align with that of the rest of the business unit. Scrum attempts to mitigate this issue by having regular “show-n-tell” sessions for other stakeholders. However, even with regular participation from other
stakeholders, an Agenda-Based Product Owner can influence the direction of the product in the wrong direction and drive additional work as the product must be refactored in response to major miscalculations in design.

As you can see, having the right person fully engaged in the project is a huge factor in driving good product. Iterative development mandates highly interactive teams. Creating the product in this manner requires that the team always be working on the right things. Development teams are not generally equipped to know what the “right” things are, so they rely almost exclusively on the Product Owner. Furthermore, since there is no major design document to work from, questions inevitably arise during development. If developers receive bad or untimely answers, the product will suffer for it.

The Absent Product Owner: A Real Life Story
I work at the University of Michigan Medical School on a software development team. The mission of our team is to support medical education. Students are admitted, enrolled, scheduled, gain learning experiences, are assessed and graded, and eventually graduate. If the need arises to either build custom software or integrate existing solutions, our team gets the call.

One area that suffered from an antiquated solution was assessment. The third and fourth year of a medical education requires students to begin practicing within a real clinical setting. The University of Michigan is blessed with a teaching hospital, but even with our large facility, we don’t have enough capacity for all of our students to gain clinical experiences in all the different areas of medicine (e.g. surgery, emergency, pediatrics). So, we send them to other medical facilities in the area. Because they are students, they must be very closely monitored. The doctors and residents in the clinics where students learn all become faculty and must provide feedback.

Each clinical area or clerkship as we call them has a coordinator and a director. The directors are very important people within the hospital. Getting any one of them to participate as a Product Owner was unlikely. Furthermore, each director has their own way of doing things, so even if we could get one of them, the likelihood of them being Agenda-Based would be extremely high.

Fortunately, there is an assessment office that manages the process across all the clerkships. The assessment office does not have as much political power as the clerkship directors, but at least they have broad and deep understanding of the problem domain. Unfortunately, they are some of the busiest people around. The head of this organization regularly works 12 and 13 hour days and weekends are not necessarily time off. Even without involvement in a major project, his work-life balance is in serious jeopardy.

In the end, we could not find a good Product Owner. Faced with the need to execute the project anyway, we settled for an Intermediary Product Owner. We leaned on an individual within our own organization who had been involved with medical education for some time and had established strong relationships with many of the key process stakeholders.

This compromise definitely took its toll on the project. Despite the deficit, the team did remarkably well. The biggest effect was in the speed early in the project. As the project
progressed, the team’s own domain knowledge expanded to allow them to make more decisions with a higher degree of confidence. The amount of refactoring to recover from bad decisions was relatively minimal.

The Intermediary Product Owner combined with the Development Team’s analyst and UX professional to interact extensively with stakeholders. This triumvirate essentially became a pseudo Product Owner. The situation was suboptimal, but creativity prevailed to fill the void left by the absence of a real Product Owner.

**Conclusion**

There’s no good substitute for a fully engaged Product Owner. That doesn’t mean that you’ll always get one. It is important to recognize when you don’t have one so you can take steps to mitigate the effects. Scrum teams practice extreme agility. In this highly interactive environment, having a subpar member will cause negative effects. There are no exceptions. Because the Product Owner directs the activities of everyone else on the team, their role is supercritical. Ignore Product Owner problems at the product’s peril.

Happy holidays! See you next year.

**Author**

**Tom Bellinson**

**Mr. Bellinson** has been working in information technology positions for 30 years. His diverse background has allowed him to gain intimate working knowledge in technical, marketing, sales and executive roles. Most recently, Mr. Bellinson finds himself serving as President of a BPM related software start-up company called UnaPage that provides solutions based on Microsoft SharePoint. From 2008 to 2011 Bellinson worked with at risk businesses in Michigan through a State funded program which was administered by the University of Michigan. Prior to working for the University of Michigan, Mr. Bellinson served as Vice President of an ERP software company, an independent business and IT consultant, as chief information officer of an automotive engineering services company and as founder and President of a systems integration firm that was a pioneer in Internet services marketplace. Bellinson holds a degree in Communications with a Minor in Management from Oakland University in Rochester, MI and has a variety of technical certifications including APICS CPIM and CSCP.