

What Is a Product?

by Mike Cohn • 41 Comments

What Is a Product?

If you work on a Scrum team, you undoubtedly know you should have a product backlog and product owner. But what, exactly, is a product?

For some teams, this can be a rather fundamental question. After all, an organization cannot identify appropriate product owners, teams and roles without first knowing what its products are. And if there is to be one product backlog per product, we need to know what our products are before creating a product backlog for each.

In most cases, it would seem straightforward to identify an organization's products. For example, a watch manufacturer might think of each item they sell as a product. But even that could be oversimplifying things. And, some organizations have a much harder time identifying products.

What Are an Airline's Products?

Take for example, an airline. A few years back, I was doing some work with an airline when the question of what is a product came up. Some people in the company argued that an airline does only one thing: move people from one location to another. They argued, therefore, that despite the company having over 40,000 employees, there should be only one product.

Others argued that there were many products within the airline. For example, they had a passenger-facing website that could be used to do things like make reservations, check in for a flight or check a flight's status. They also had a system for monitoring and scheduling aircraft maintenance. And another that allowed crew members to select the flights they wished to work based on seniority.

Were these also products?

A Definition of Product and Examples

I define a product as something (physical or not) that is created through a process and that provides benefits to a market.

From that, a chair would be a product. Microsoft Office would be a product. Agile consulting services would be a product. A painting would be a product. A product can be a something physical (the chair). It can be a digital product (Microsoft Office, an ebook or streaming videos). It also can be a service (consulting on how to adopt agile).

A product can even just be an idea (a patentable algorithm or the secret to getting more right swipes on Tinder).

Each of these is created through a process or, more generally, one or more activities. Someone milled and assembled the chair. Microsoft Office was designed, coded and tested. The process that creates a product does not need to be formal or defined. The creators may not even be aware of the process. But some form of activity goes into creating every product.

Products Can Be Defined Recursively

A product can exist within another product. A pen, for example, may have replaceable ink cartridges. The pen is a product. But so are the ink cartridges within the pen.

There are even subproducts within a chair. The company manufacturing and selling the chair may have purchased fully milled legs from another company. Chair legs would then be a product.

Products can be defined recursively. A lumber company harvested trees to make wood—a product in its own right. Next, a milling company used that wood to make chair legs. Another company then took those pre-milled chair legs, and used them to assemble chairs of their own design. So products can exist inside other products.

Products Provide Benefits to a Market

When we identify subproducts within a larger product, we need to be careful that each subproduct provides benefits to a market. The definition above does not say that something must be purchased for it to be a product. But, to be considered a product, the item must satisfy a need or desire.

This is true of pre-milled legs for a chair and replacement ink cartridges for a pen. When defining products--and, therefore, product owners and product backlogs in Scrum--it will be important to define each product such that it provides benefits to a market.

Applying this Definition to the Airline Example

So if products are created through some process and benefit some market, are software components products?

To help answer that question, consider the airline company example from earlier. How would knowing that a product is something that is created through a process and that provides benefits to a market help an airline company identify its products?

First of all, it is easy to see that transporting people from one place to another is a product. That activity provides value to a market of people who gladly pay for the ability to fly to a location.

But what about the system for monitoring and scheduling aircraft maintenance? I claim that, too, is a product.

It is created through a process and is also something that provides benefits to a market. What market?

Well, we could go all the way and say that passengers benefit from well-maintained and safe aircraft. But, even closer to the product's development, we can say that the airline employees benefit from having maintenance monitoring and scheduling computerized rather than needing to do that manually with paper.

The same could be said of the airline's website, which enables passengers to make flight

reservations. There is a market for that.

So, yes our airline has one big product—and it also has many subproducts. Anything within that company that can be thought of as delivering value to a market is a product.

Applying this Definition to Software Components

With that in mind, consider a software component. If one team develops software used by other teams, can that be thought of as a product? Consider the example of a team building a calendar widget. It provides value to a market: the other teams that will use the widget. I would say, then, that a component built for multiple teams is a product.

I would draw the line, though, with a calendar widget (or any component) that is *used by only one team*. Yes, technically a market can exist with only one customer. A painting, for example, is sold to one person.

But, when talking about product development (as with agile), it can be dangerous to think of something as a product if it is used by only one person or group. It could lead, for example, to thinking of code as a product that is delivered to a *market* of testers. That is not only a step backward into sequential (or phased) development, it's also a form of suboptimization.

Avoiding Suboptimizing Products

According to San Jose State University [Economics professor Thayer Watkins](#), suboptimization “refers to the practice of focusing on one component of a total and making changes intended to improve that one component ... ignoring the effects on the other components.”

While organizations do want to define all of their products in order to best manage the work, they do not want to narrow their focus to the degree that they fail to see the whole because they are fixated on the individual parts. As such, organizations want to define each product as broadly as possible.

That being said, as we saw earlier with the airline company, there is such a thing as too broad when it comes to identifying products. When a product is so large that it serves multiple markets,

I often prefer to think of it as multiple products each serving a unique market.

For example, it would be quite easy to think of Microsoft Office as a single product. Yet, Office is a suite of products, each providing different features and serving slightly different (but overlapping) markets.

Because of that, I'd want to think of Word, Excel, PowerPoint and so on each as its own product. Each would have its own product owner and product backlog.

With a product as large as Office, it is quite likely there also would be products based on shared functionality across things like Word, Excel and PowerPoint. Thinking of the spell checker, for example, as a product could make sense. The spell checker provides benefits to a market (the other teams who do not need to write their own spell checker from scratch).

I would not, however, define Excel's Function widget (sum, average, count, etc.) as a product. It is unique to Excel and as such, serves only one customer. To give it a product backlog and product owner outside the context of Excel would be too risky.

In addition to avoiding one-customer products, another way to reduce suboptimal thinking when working with multiple products is to appoint a chief product owner. The chief product owner is a strategic role, responsible for establishing vision across all subproducts. On the Office product, for example, the chief product owner would be looking at how each individual product affects the suite as a whole.

What Do You Think?

Correctly identifying the products in your organization can help you avoid problems related to team structure, product backlog management and having people in the wrong roles. How do you define "product" in your organization? Are there guidelines you can extrapolate to create a general definition of product? Please share your thoughts in the comments below.

Posted: September 6, 2016

Tagged:

product backlog, product owner, product management, software, product, service, hardware
