


Not Everything Needs to Be a User Story: Using FDD Features

by Mike Cohn •

40 Comments

The following was originally published in Mike Cohn's monthly newsletter. If you like what you're reading, sign up to have this content delivered to your inbox weeks before it's posted on the blog, [here](#).

User stories are great. When you've got users, that is. Sometimes, though, the users of a system or product are so far removed that a team struggles to put users into their stories. A sign that this is happening is when  Blue Box Stands Out from Line of White Boxes teams write stories that begin with "As a developer..." or "as a product owner...."

There are usually better approaches than writing stories like those. And a first step in exploring alternative approaches is realizing that not everything on your product backlog has to be a user story.

A recent look at a product backlog on a product for which I am the product owner revealed that approximately 85% of the items (54 out of 64) were good user stories, approximately 10% (6 of 64) were more technical items, and about 5% (4 of 64) were miscellaneous poorly worded junk.

Since I'm sure you'll want to know about the junk, let's dispense with those first. These are items that I or someone else on the project added while in a hurry. Some will later be rewritten as good stories but were initially just tossed into the backlog so they wouldn't be forgotten. Others are things like "Upgrade Linux server" that could be rewritten as a story. But I find little benefit to doing that. Also, items like that tend to be well understood and are not on the product backlog for long.

My point here: No one should be reading a product backlog and grading it. A *little* bit of junk on a product backlog is totally fine, especially when it won't be there long.

What I really want to focus on are the approximately 10% of the items that were more technical

and were not written as user stories using the canonical “As a <type of user>, I want <some goal> so that <some reason>” syntax.

The product in question here is a user-facing product but not all parts of it are user facing. I find that to be fairly common. Most products have users somewhere in sight but there are often back-end aspects of the product that users are nowhere near. Yes, teams can often write user stories to reflect how users benefit from these system capabilities. For example: *As a user, I want all data backed up so that everything can be fully recovered.*

I’ve written plenty of stories like that, and sometimes those are great. Other times, though, the functionality being described starts to get a little too distant from real users and writing user stories when real users are nowhere to be found feels artificial or even silly.

In situations like these I’m a fan of the syntax from the Feature-Driven Development agile process. Feature-Driven Development (FDD) remains a minor player on the overall agile stage despite having been around since 1997. Originally invented by Jeff De Luca, FDD has much to recommend it in an era of interest in scaling agile.

Wikipedia has a [good description of FDD](#) so I’m only going to describe one small part of it: features. Features are analogous to product backlog items for a Scrum project. And just like many teams find it useful to use the “As a <user>, I want <goal> so that <benefit>” syntax for user stories as product backlog items, FDD has its own recommended syntax for features.

An FDD feature is written in this format:

<action> the <result> <by|for|of|to> <object>

As examples, consider these:

- Estimate the closing price of stock
- Generate a unique identifier for a transaction
- Change the text displayed on a kiosk
- Merge the data for duplicate transactions

In each case, the feature description starts with the action (a verb) and ends with what would be an object within the system. (FDD is particularly well suited for object-oriented development.)

This can be a particularly good syntax when developing something like an Application

Programming Interface (API). But I find it works equally well on other types of back-end functionality. As I said at the beginning, about 10% of my own recent product backlog I examined was in this syntax.

If you find yourself writing product backlog items for parts of a system and are stretching to think of how to write decent user stories for those items, you might want to consider using FDD's features. I think you'll find them as helpful as I do.

Posted: July 21, 2015

Tagged: user stories, product backlog

About the Author

As the founder of Mountain Goat Software, Mike Cohn specializes in helping companies adopt and improve their use of agile processes and techniques to build extremely high-performance teams. He is the author of *User Stories Applied for Agile Software Development*, *Agile Estimating and Planning*, and *Succeeding with Agile*. Mike is a founding member of the Agile Alliance and Scrum Alliance. Mike can be reached at hello@mountaingoatsoftware.com. If you want to succeed with agile, you can also [have Mike email you a short tip](#) each week.
