

Job Stories Offer a Viable Alternative to User Stories

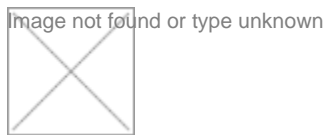
by Mike Cohn • 37 Comments

Job Stories Offer a Viable Alternative to User Stories

As useful as [user stories](#) can be, they've never been right for every team. An exciting alternative for some teams is the *job story*. A job story is focused less on the user performing some function than on the job to be done by that story. Job stories originated at [Intercom](#) and were best explained by [Alan Klement](#).

A Job Story Template

To see how a job story shifts emphasis from the user to a job to be done, let's take a look at the recommended job story template:



As with the common [user story template](#), there are three parts to complete in the job story template. The first is known as the *situation*. This follows the word *when* in the template and provides context on when the story is being performed or perhaps what triggered the story. Examples could be:

- When an order is submitted...
- When searching by postal code...
- When no matches are found...
- When looking at recent orders...

The second element of the job story template follows the *I want to* and provides the *motivation*

for the story. Think of the motivation as a user's stated or first-order goal.

As an example, I want pizza for dinner tonight. Why do I want pizza tonight? Because I am meeting some friends tonight to watch a football game, and it's easy to feed a group of us with different dietary needs and preferences with pizza. In the job story world, easily feeding a group is referred to as the *expected outcome* and it follows the *so I can* portion of the template.

Putting my desire for pizza into a job story would lead to

- When it's dinner time tonight, I want to have pizza so I can easily feed my friends.

This isn't a particularly perfect job story, but it illustrates the difference between a user's motivation and their expected outcome.

Contrasting Job and User Stories

To see the times when job stories may be better than user stories, let's look at some sample job stories and their corresponding user stories.

When an Order Is Submitted...

Let's start with this job story:

Job Story:

When an order is submitted, I want to see a warning message so I can avoid resubmitting the order.

This story describes the behavior seen on most eCommerce sites warning a user not to submit an order multiple times.

The user story equivalent of this might have been

User Story:

As a customer, I want to be shown a message telling me not to submit an order twice so that I don't place a duplicate order.

The job story is superior in this case for two reasons. First, this story applies to everyone making a purchase on the site. So it's not important to know the person doing this is a customer. (In fact, calling the person a customer could be misleading because the person may not be a customer until this order has been placed.)

Second, the job story is better because it provides more context about when this is happening. It is happening "when an order is submitted," as the job story tells us. Look carefully at the user story and you'll notice that it never tells us when this message is displayed. The team could "successfully" implement the user story by adding an item on an FAQ page warning against double submitting orders. That is almost certainly not what the product owner wants.

When an Order Is Submitted...

Let's look next at a job story about searching for an address by United States ZIP (postal) code.

Job Story:

When searching by US ZIP code, I want to be required to enter a 5- or 9-digit code so I don't waste time searching for a clearly invalid postal code.

This story is about making sure a user enters something reasonable for a postal code before allowing a search. The United States postal, or ZIP, codes are either 5 or 9 digits. This story says users should be prevented from clicking search with only entering two letters entered in the postcode field.

The equivalent user story would be:

User Story:

As a user, I want to be required to enter a 5- or 9-digit postal code so I don't waste time searching for a clearly invalid postal code

These two stories highlight that the difference between user and job stories exists in the first part of the templates. The when... and As a ... clauses differ but in this example, the remainder of each story is identical in both user and job story format.

As in the first example, the job story is better here because of the additional context it provides around *when* the story is being performed. Who is performing the action (the search in this case) is not important, which is why the user story is written with the generic, "As a user..."

When to Use Job Stories

In deciding when to use job stories, I think it's important to acknowledge that both user and job stories have unique strengths.

I still find user stories most helpful for products that have users who vary significantly and deeply understanding those users is important. This is why user stories start with As a... We start user stories that way because that puts the user right upfront. With user stories, *who* will be doing the story is perhaps as important as *what* they'll be doing.

In contrast for a job story, it is not necessarily important who is doing the story. This makes job stories the better option when your product has users but their needs are not very distinct.

If you've ever written a lengthy set of user stories and started each with "As a user...", you've encountered this problem. When a large set of user stories all begin with "As a user..." you've got a set of stories for whom the user is not very important.

Writing those as job stories rather than user stories would be helpful. Doing so would allow the story to include the additional context of when the story is being performed. In some cases, knowing when a story might happen is more important than knowing who will perform it.

Combining the Strengths of Job and User Stories

So, while job and user stories each have their own strengths, it's possible to merge them and get both benefits in one story. Let's revisit our postal code stories and see how to do this. First, we had the job story:

Job Story:

When searching by postal code, I want to be required to enter a valid code so I don't waste time searching for a clearly invalid postal code.

It's never clear who is performing this story. Is it a normal user? A site admin? Someone else? We're never told. If we think knowing who is doing this is important, we can augment the job story by adding a role to the story in place of I. This changes our story to be the following:

Job Story:

*When searching by postal code, **a buyer** wants to be required to enter a valid code so **the buyer doesn't** waste time searching for a clearly invalid postal code.*

The changes are in bold. You can see that I just changed from "I want..." to "a buyer wants..." and then made the corresponding change later in the story.

We can do something similar to a user story. Our initial, unmodified story was:

User Story:

As a user, I want to be required to enter a valid postal code so I don't waste time searching for a clearly invalid postal code

To provide additional context, we add a phrase telling us when the user is doing this. Our modified user story then becomes “

User Story:

*As a user **who is searching by postal code**, I want to be required to enter a valid postal code so I don't waste time searching for a clearly invalid postal code*

The modified user and job stories are semantically the same. Which you choose is entirely up to you. I personally prefer the modified user story over the modified job story because it keeps the story in first person. I've written elsewhere about the [benefits of stories being in the first person](#).

When to Use Job Stories

So when should you prefer user or job stories over the other?

First, each is great and has its own advantages. During the course of any week, I will write some user stories and some job stories. The two techniques are quite compatible and there's no reason to view them as mutually exclusive.

If your product has users and those users needs differ in important ways, I suggest user stories. The additional emphasis a user story puts on who is performing the action can lead to insights about user behavior.

If, however, your product's users do not differ in significant ways, job stories are likely the better approach.

A good starting point is to mix user and job stories in the same product backlog. Start by writing job stories any time you are tempted to write a batch of stories all beginning with "As a user..."

What's Your Experience?

What do you think of job stories? Would they be helpful on your product backlog? Have you worked with job stories before? Were they helpful? Please share your thoughts in the comments section below.

Posted: October 29, 2019

Tagged: product backlog, user stories, job stories
