

How to Work with Complex User Stories That Cannot Be Split

by Mike Cohn

How to Work with Complex User Stories That Cannot Be Split

Most user stories can be split. It may be hard to find a good way to split some stories, but most can be split. These are known as *compound* stories—stories that are made up of multiple smaller stories.

There is another type of story: the *complex* story. Complex stories are ones that cannot be split. They are inherently large or complex and there are no subparts to be pulled into separate stories.

Even with a complex story, you don't want to let the story linger open for three, four or more sprints. Doing so

- Makes velocity less predictable from sprint to sprint
- Increases the risk of a developer going astray from user expectations , and
- Allows the team to develop the bad habit of leaving work incomplete at the end of an iteration

Use Progress Points to Identify Accomplishments

So, what is a good, agile team to do?

When it's impossible to find a good, natural split that results in two or more smaller, functional stories look instead for *progress points*.

A progress point is any point during the development (usually coding) of a feature at which the developer feels something has been accomplished.

No developer wants to go two months without the job satisfaction of being able to tell teammates, “I got such-and-such accomplished.”

To identify progress points, ask the developer who is engaged with a complex story, “Where are the points while working on this where you’ll want to tell a tech-savvy teammate, ‘Dude, check it out! I got the this-and-that to work!’”

Examples of Progress Points

As an example, suppose a developer is working on some complex code to interact with a remote system. The developer expects to need two or three iterations to be done.

A progress points on that could be that a first transaction is sent to (or received from) the other system. No error checking or handling is performed but a first exchange with the other system would be a point at which the developer might share the accomplishment with a second tech-savvy teammate who understands the significance of the accomplishment.

A second progress point could be more transaction types being exchanged, errors being detected, or errors being retried.

Once identified, many of these progress points can be quite readily expressed as user stories. So, thinking of progress points helps developers identify natural milestones they’ll reach during the implementation of a large story.

Using a Story’s Progress Points

Having identified one or more progress points within a complex story, use those as milestones. Ask the developer to identify how many of the progress points can be completed during the coming iteration and then use that to track progress just as you would a regular user story.

What’s Your Experience?

Have you tried identifying progress points within your team’s complex stories? How did it work? What else have you tried to make working with complex stories more manageable? Please

share your thoughts in the comments below .

Posted: November 6, 2018

Tagged: user stories, story size, splitting stories
