

Advice on How to Split Reporting User Stories

by Mike Cohn • 47 Comments

Advice on How to Split Reporting User Stories

I've had a handful of emails lately about the difficulty of completing a complex reporting user story in a sprint.

These emails made the claim that perhaps reports were not something well suited for development with agile because some reports are complicated and take more than a sprint to develop.

I'd argue that the opposite is the case. When something will take a long time to develop, that's exactly when I want to use a process that forces me to get early and frequent feedback. This will counter many developers' tendency (including my own) to retreat into a cave thinking we know what our users want. So in this blog post, I want to look at a few ways that a complicated reporting user story can be split. I think it's a useful example because its lessons can be applied to other types of agile work.

One way to split a complicated report is to completely finish the presentation of the report but to bring in only a subset of its data sources. This resulting report at the end of this initial sprint will show useless data. But a report built in this way can be useful for gathering feedback on the presentation of the data.

In building a report this way, I'll often select one of the easiest data sources first. There is often a great deal of work just to make that happen independent of any source-specific complexities. So I'll save the complex data sources for second or third sprints once I know the report itself is working properly.

A second approach is the exact opposite: Bring in all data sources but prepare an incomplete report. Perhaps only summary values are shown. Or perhaps the itemized rows are displayed but they are not summarized and totaled (whichever is easier). Or perhaps they are shown in detail and total form but only in one way.

For example, suppose we are creating a report showing information on revenue earned at a movie theater. In the initial sprint, the report may show it split and totaled by movie title. But it may not yet show the detailed by day of week or by the start times each day.

Each of these approaches works because it is true to goal of being *done* with something at the end of the iteration. The entire user story is not done. (We are assuming this report is too big to be completed in a single iteration.) Yet, even when the full story cannot be delivered, the team delivers some portion of it to a fully working state. They do not content themselves with delivering a full but untested version.

Instead, they deliver a version that works but uses data from only one source. Or that works fully but only fills in one portion of the eventually complete report.

And this is how successful agile teams approach all of the work, not just reporting user stories. Each iteration, the team fully develops some portion of the user story.

In the comments below, please let me know what you think of the approaches I've described. And let me know about other approaches you've taken to splitting large stories about reports or any other type of functionality.

Posted: March 8, 2016

Tagged: user stories, story splitting, reporting
