

Writing User Stories for Back-end Systems

by Mike Cohn • 74 Comments

Writing User Stories for Back-end Systems

I was asked recently how to go about writing user stories for a back-end financial system. This is an interesting example and is a question I get asked a lot, so I thought I should answer it here.

This example brings up a couple of key interesting challenges:

- While there may be users of the system they are not often not direct users (i.e., with hands on the keyboard waiting for something to happen)
- The functionality is often larger than will fit in one iteration

So that we can deal with these challenges, let's make up some more context for the example. Suppose our financial system takes in a lot of daily data and produces flat files that will be sent to banks and other partners at the end of the day. Some of the files are simple formats. Other files are in more involved formats with multiple record types within the file and possibly with multiple lines for some of the transactions. This would be a fairly standard batch-processing type application. I want to write user stories in the template I recommend in [the book](#): *As a , I want so that* . So, let's deal with these challenges in order and first try to figure out who the user will be in our stories? Given the context provided above the user, is probably a bank or business partner. This will let us write stories like *"As a bank, I want..."* It's entirely possible that we will want to get more specific and sometimes write stories for more specific users:

- As a commercial bank, I want....
- As a savings & loan, I want...
- As the Bank of America, I want... (assuming we have some specific business partnership that provides BofA with unique functionality)

Now, what is it these banks want? At a high level we can start out with some epics, perhaps like these:

- As a bank, I want to receive a file showing all checks to be cleared so that I can debit and credit the right accounts.

- As a bank, I want to receive a correctly formatted 5300 file so that I can adjust balances as appropriate.
- As a bank, I want to receive a file showing the status of all resubmitted transactions so that I can update the database with new information.

(Note that I am liberally making up specific business rules or details. Presumably someone writing the real user stories could add real and relevant detail instead.) First, notice it is OK to humanize the bank with “As a bank, I want...” Programmers do this all the time in conversation. “OK, suppose I'm the bank and you send me a 5300 file with a bad record. In that case, I'll...” As I said at the start of this post, one of the challenges in an application like this is that some of these user stories we just made up could be large. Suppose our mythical file format 5300 includes a header, three different record types, and a footer. We could break up the 5300 file user story above into things like:

- As a bank, I want to receive a 5300 file with the correct header and footer so that I can process the file correctly.
- As a bank, I want to receive a 5300 file with correctly formatted prearranged payment records so that I can process those correctly.
- As a bank, I want to receive a 5300 file with correctly formatted deposit entry records so that I can process them.
- As a bank, I want to receive a 5300 file with correctly formatted addendum records so that I can process them.

If those were still too big, I would find ways to split them smaller based on the specific formatting of the records. Suppose our deposit entry record type sometimes fits on one line but can also span multiple lines. In that case I could write these user stories:

- As a bank, I want to receive a 5300 file with correctly formatted single-line deposit entry records so that I can process them.
- As a bank, I want to receive a 5300 file with correctly formatted deposit entry records that include continuation lines so that I can process them.

Is the system “potentially shippable” after each of these user stories is delivered? Yes, it kind of is. While I'll admit that we're unlikely to find a bank that will take a 5300 file with only a header and a footer, it's possible. Especially if we think of the situation in which the bank is simultaneously writing their own software to process the 5300 files we send them. Imagine

meeting with the bank before the start of the next iteration and perhaps agreeing, “OK, in this iteration we'll create some blank 5300 files with just headers and footers. But that will let us make sure that the communication problems are worked out, that you can read the files we write, and so on...” These stories also work well because they will likely fit with how the programmers will want to build the system.

Posted: February 24, 2008

Tagged: user stories, agile requirements, iterations
