Features to User Stories

Big Feature

The word processor should be able to support formatting text styles.

Smaller Story

The user should be able to select a word and specify that it be italics.

Mike Cohn's canonical form

As a <type of user>, I want <some goal> so that <some reason>.

"As a *user of the word processor*, I want *to select a word and specify that it be italics*, so that *I can add emphasis to my writing*."

Just a suggestion

You don't have to write the story using Mike Cohn's template (It gets very monotonous), but it's good to consider these three things:

1. Who will use this feature?
2. What do they want the system to do?
3. What larger goal of the user does this support?

One more important question

4. What will tell me that this story is done?

Ron Jeffries' "Card, Conversation, Confirmation"

Card—The story card is just a token. It's the title; it doesn't carry all the details. You can make notes of any details you might otherwise forget, but *don't expect the written story to transfer understanding from the business to the developer*. That's the purpose of the conversation.

Conversation—Communication is a two-way process, even when we're trying to transmit information in one direction. Developers need to ask clarifying questions. Business representative needs to ask questions confirming understanding. Sometimes, something will come up that throws the whole story into question.

Confirmation—How will we know that we've accomplished this story? Can you come up with some concrete examples? Turn these examples into automated acceptance tests.

William Wake's INVEST

Good stories are: Independent, Negotiable, Valuable, Estimable, Small, Testable

Don't get carried away

Not every task that needs to be done has to be mashed into the mold of a User Story. But when you're contemplating something that isn't a User Story, ask yourself if it *really* needs to be done. Does it add value?

And don't forget about other users: sysadmin, customer service, businessman looking at ROI...

Splitting Stories

Stories in the backlog should be coarse-grained. They are often called features or epics, and are too big and vague for any sort of accurate estimation of effort. They will generally include both essential and nice-to-have components. They will generally not have any specific, testable acceptance criteria. Over time, they will be split into smaller, more concrete stories for development.

Stories selected for development should generally be small enough to accomplish in a matter of hours to a couple of days at the most. This allows progress to be tracked effectively during the iteration. When splitting these from the feature or epic, it is usual to find some stories that can or should be deferred until later, after more important work has been accomplished.

Developers may want to split stories further (minutes to a couple hours at most) when they're implementing them. This allows them to check their progress and avoid dead ends or over-engineering. At this point, one test scenario is roughly equivalent to one story.

Suggestions for Splitting

Handle a simpler case before proceeding to more detailed or complicated ones. Think in terms of progressive refinement of the functionality.

- Zero, one, many. Handle the empty scenario first.

- Happy path, then each alternate flow and exceptional condition.

- Core function, then prerequisites for public consumption.

- Single option, then add each additional option.

- Simple (or no) UI, then add bells and whistles.

- Transient case (no memory between sessions) before persistence.

- Static elements, then dynamic based on context.

- User specified, then more automation.

Resources

- William Wake, *"Twenty Ways to Split Stories"* http://xp123.com/xplor/split-summary/

- J.B. Rainsberger, *"Splitting stories: an example"* http://jbrains.ca/permalink/5

- Lasse Koskela, *"Ways to split user stories"*
  http://radio.javaranch.com/lasse/2008/06/13/1213375107328.html

- James Grenning *"Story Weight Reduction Toolkit"*
  http://www.renaissancesoftware.net/blog/archives/48

Rev: 09/07/2009 15:49:50