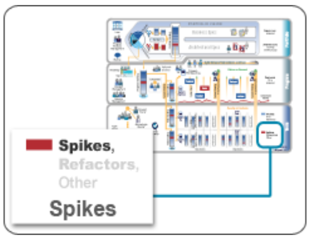

“If we knew what we were doing it wouldn’t be research.

- Albert Einstein

Spikes Abstract

Spikes are a type of story that are used for activities such as research, design, exploration and even prototyping. The purpose of a spike is to gain the knowledge necessary to reduce the risk of a technical approach, better understand a requirement, or increase the reliability of a story estimate. Originally defined within XP, spikes primarily come in two forms, technical and functional. Functional spikes are used to analyze aggregate functional behavior and to determine how to break it down, how it might be organized and where risk and complexity exists, in turn influencing implementation decisions. Technical spikes are used to determine feasibility and impact of design strategies. Like other stories, spikes are estimated, owned by a team member, and are demonstrated at the end of the iteration.

Details

Spikes, another invention of XP, are a special type of Story used to drive out risk and uncertainty in a user story or other project facet. Spikes may be used for a number of reasons:

- Spikes may be used for basic research to familiarize the team with a new technology or domain.
- The story may be too big to be estimated appropriately, and the team may use a spike to analyze the implied behavior so they can split the story into estimable pieces.
- The story may contain significant technical risk, and the team may have to do some research or prototyping to gain confidence in a technological approach that will allow them to commit the user story to some future timebox.
- The story may contain significant functional risk, in that although the intent of the story may be understood, it’s not clear how the system needs to interact with the user to achieve the benefit implied.

Technical Spikes and Functional Spikes

When a story is too complex to estimate, or simply needs additional research and underdstanding before it enters a sprint, teams use Technical Spikes and Functional Spikes to further understand the solution (Figure 1).

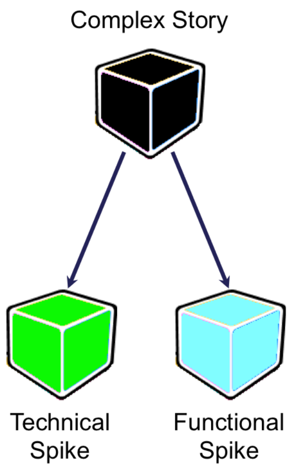

Figure 1. Use Technical and Functional Spikes to better understand complex stories

Technical spikes are used to research various technical approaches in the solution domain. For example, a technical spike may be used to determine a build-versus-buy decision, to evaluate potential performance or load impact of a new user story, to evaluate specific implementation technologies that can be applied to a solution, or for any reason when the team needs to develop a more confident understanding of a desired approach before committing new functionality to a timebox.

Functional spikes are used whenever there is significant uncertainty as to how a user might interact with the system. Functional spikes are often best evaluated through some level of prototyping, whether it be user interface mock-ups, wireframes, page flows, or whatever techniques are best suited to get feedback from the customer or stakeholders. Some user stories may require both types of spikes. Here's an example:

> *As a consumer, I want to see my daily energy use in a histogram so that I can quickly understand my past, current, and projected energy consumption.*

In this case, a team might create two spikes:

- **Technical spike:** Research how long it takes to update a customer display to current usage, determining communication requirements, bandwidth, and whether to push or pull the data.
- **Functional spike:** Prototype a histogram in the web portal and get some user feedback on presentation size, style, and charting attributes.

Guidelines for Spikes

Since spikes do not directly deliver user value, they should be used sparingly and with caution. However, these story spikes represent risks in your backlog because they are hiding other implementation stories that may increase the scope of the plan. Consider prioritizing them ahead of other less risky items to drive out risk early in the release plan. The following are some guidelines for applying user spikes.

Estimable, Demonstrable, and Acceptable

Like other stories, spikes are put in the backlog, estimated, and sized to fit in an iteration. Spike results are different from a story, because they generally produce information, rather than working code. A spike may result in a decision, prototype, storyboard, proof of concept, or some other partial solution to help drive the final results. In any case, the spike should develop just the information sufficient to resolve the uncertainty in being able to identify and size the stories hidden beneath the spike.

The output of a spike is demonstrable, both to the team and to any other stakeholders. This brings visibility to the research and architectural efforts and also helps build Collective Ownership and shared responsibility for the key decisions that are being taken. And, like any other story, spikes are accepted by the product owner when the acceptance criteria for the spike have been fulfilled.

The Exception, Not the Rule

Every user story has uncertainty and risk—this is the nature of agile development. The team discovers the right solution through discussion, collaboration, experimentation, and negotiation. Thus, in one sense, every user story contains spike-level activities to flush out the technical and functional risk. The goal of an agile team is to learn how to embrace and effectively address this uncertainty in each iteration. A spike story, on the other hand, should be reserved for the more critical and larger unknowns. When considering a spike for future work, first consider ways to split the story through the strategies discussed earlier. Use a spike as a last option.

Implement the Spike in a Separate Iteration from the Resulting Stories

Since a spike represents uncertainty in one or more potential stories, planning for both the spike and the resultant stories in the same iteration is risky and should generally be avoided. However, if the spike is small and straightforward and a quick solution is likely to be found, there is nothing wrong with completing the stories in the same iteration. Just be careful.

User Experience Story Spikes

User experience story spikes can be used to develop alternative user interfaces and test them through actual or representative users. These stories are put in the backlog, sized, developed, and tested as any other story. The only difference is that they may not end with working software. The act of scheduling them in the Iteration has the effect of deferring the code development until after the tests are complete. This can improve efficiency immensely.

Learn More

[1] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs and the Enterprise*. Addison Wesley. 2011.

Last update 24 March, 2013

Discussion

0 Comments 2 Trackbacks

LATEST UPDATES TO THE FRAMEWORK

- Extending an Epic Treatment in SAFe (just a little)
- Al Shalloway on Making Scrum SAFe(r)
- A Nice Blog Post on Building Corporate Culture with SAFe
- Lean|Agile Financial Planning with SAFe

BOOK
Leffingw ell, Dean:
Agile Sofw are Requirements: *Lean Requirements Practices for Teams, Programs, and the Enterprise,* (Pub. 2011)

BLOG
scalingsoftw areagilityblog.com

Home Updates Contact top