JOIN OVER 100,000 PROS! Join Log In | Member Benefits

STICKYMINDS

Agile Testing DevOps Security AI

Questioning before Answering

ARTICLE

By Steve Berczuk | December 8, 2011

SUMMARY

The other day I came across a short video in which a parent is faced with answering a unexpected question posed by her young child. I found this video amusing because, being the parent of a kindergartener, I expect to be faced with many awkward moments like this in the future. I also found it an interesting metaphor for software requirements gathering.

The other day I came across a **short video** in which a parent is faced with answering a unexpected question posed by her young child. I found this video amusing because, being the parent of a kindergartener, I expect to be faced with many awkward moments like this in the future. I also found it an interesting metaphor for software requirements gathering.

In the video the parent could have answered the question a whole lot more simply has she just asked taken a moment to try to understand what the real question was, and not taken the question at face value. (I'm being purposely vague at this point in case anyone wants to watch the video.)

Most of us have been on projects where much effort has been spent in an attempt to solve a customer's stated problem, only to discover that that either:

- The program didn't actually solve any of the customer's real problems.
- The team didn't solve the problem that the customer wanted solved.
- There was a much simpler solution.

I'm not suggesting that exhaustive requirements capture is the answer. Nor am I suggesting that you should not start building software until you have complete understanding of a problem. The reason that agile methods work well is that they acknowledge uncertainty, and provide a simple path to refining understanding: evaluating working software.

We use cookies to ensure you get the best experience on our website.

Accept

Read More

system. In many cases a customer will know the reason they want to do something, but may not be able to step back enough to express it. In these cases, but not trying to ask "why?" your team is ignoring information that can help them do their work. In more cases than not, a customer can fill in the pieces of **the user story template**:

As a (ROLE) I want to (DO SOMETHING) so that (REASON)

and provide insight into the problem that she wants to solve. This will save time, effort, and help the team deliver a better system.

If your team takes customer requirements statements at face value without exploring the reasons why a customer wants a piece of functionality, they aren't holding up their end of the collaboration part of agile software development.

Had the parent in the video though to ask why the child asked the question that she did, the process of answering might have been simpler.

Topics: agile, process improvement,

ABOUT THE AUTHOR

Steve Berczuk

Steve Berczuk is a Principal Software Engineer with experience as a manager, Scrum Master and technical lead in Boston, MA. The author of *Software Configuration Management Patterns: Effective Teamwork, Practical Integration*, he is a recognized expert in software configuration management and agile software development. Steve is passionate about helping teams work effectively to produce quality software. He has an M.S. in operations research from Stanford University and an S.B. in Electrical Engineering from MIT, and is a Certified ScrumMaster. Contact Steve at steve@berczuk.com or visit berczuk.com and follow his blog at blog.berczuk.com.

UPCOMING WEBINAR

We use cookies to ensure you get the best experience on our website.

LETS HANG!

Log In or Register to post comments

USER COMMENTS

0 comments

Join the crew for FREE! Access exclusive content, contribute, and more.

See All Member Benefits

We use cookies to ensure you get the best experience on our website.

We use cookies to ensure you get the best experience on our website.