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

STICKYMINDS

Agile Testing DevOps Security AI

More on Being Done

ARTICLE

By Steve Berczuk | October 16, 2011

SUMMARY

Continuing the conversation from last week, Andy Singleton followed up on my post on being done with this post. Which is good as this is one of those questions that sounds simple in theory, but in practice contains some subtlety.

Continuing the conversation from last week, Andy Singleton followed up on my post on being done with this post. Which is good as this is one of those questions that sounds simple in theory, but in practice contains some subtlety.

While I was advocating a good definition of "Done" to enable you to measure progress along a path, Andy's point seems to be that many teams don't establish enough of a path. He says:

In my opinion, most agile teams aren't doing "Test Driven Development", and they aren't doing scrum iterations where they plan everything in advance. Instead, they are doing "Release Driven Development." They focus on assembling releases, and they do a lot of planning inside the release cycle.

This is probably true in more cases than not, though one could argue whether if you are not doing iterations or TDD whether you are, in fact, doing agile. But even if can concede (which I'm not sure if I am) that you can do agile without planning and acceptance criteria of some sort, what Andy describes above still sounds like it has an element of plan, execute, adjust, though perhaps in a more chaotic way than a "textbook" scrum process, and and perhaps at a smaller scale. So knowing having a clean definition of what you want to do is still important. In the Indivisible Task I discussed how teams get stuck with planning because it's difficult to think of tasks that are small, discrete, and which produce useful work. It is possible to do so, but it is hard.

Perhaps the disagreement here is more of a misunderstanding. I consider being able to

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

essential. If you do this at a fine-grained enough level, and have a good testing infrastructure, you can release software more frequently. So, at the limit, perhaps Andy and I are talking about the same thing.

Since I only summarized, I encourage you to read **what Andy has to say** for yourself. And I look forward to hearing comments both from Andy and other readers.

Topics: agile,

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

LETS HANG!

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

Shifting Security Left: DevSecOps eGuide
White Paper | StickyMinds

Log In or Register to post comments

USER COMMENTS

0 comments

YOU MAY ALSO LIKE

SECURITY
Agentic AI Opens Up New Security Flaws
Nahla Davies | August 27, 2025

SECURITY
Countering the Threat of Ransomware-as-a-Service (RaaS)
Nahla Davies | August 22, 2025

SECURITY
Building Trust in Blockchain: The Role of Smart Contract Testing
Ekaterina Egorova | November 18, 2024

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

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.