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

STICKYMINDS

Agile Testing DevOps Security AI

What Does "Being Done" Really Mean in Software Development?

ARTICLE

By Steve Berczuk | October 12, 2011

SUMMARY

Agile New England (which used to be called the New England Agile Bazaar, and which was started by Ken Schwaber) , has this wonderful activity before the main event each month: they host Agile 101 sessions, where people who know something about agile lead a short (30 minutes) small (about 10 people) class on agile basics for those who want to learn more about some aspect of agile.

Agile New England (which used to be called the New England Agile Bazaar, and which was started by Ken Schwaber) , has this wonderful activity before the main event each month: they host Agile 101 sessions, where people who know something about agile lead a short (30 minutes) small (about 10 people) class on agile basics for those who want to learn more about some aspect of agile.

From time to time I lead a session on Agile Execution, where the goal is to help people understand how to address the following questions:

- How can software and other project elements be designed and delivered incrementally? What set of management and technical practices would enable this?
- How do you know whether your Agile project will complete on schedule?

When I lead the sessions, I tend to focus on tracking, defining stories in terms of vertical slices and the importance of continuous integration and testing to making your estimates trackable. Since the classes are so small and since the attendees have diverse experiences, the classes are sometimes more of a conversation than a lecture, and I find that I learn a lot, and sometimes find myself rethinking what I know (or at least exploring things that I thought I understood well enough).

During the October 2011 meeting I found myself reconsidering the value of defining

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

Accept

Read More

Assembla suggested that: "The only useful definition of done is that you approved it to release, in whatever form."

While the goal of agile methods is releasing software, I find that this definition, while appealing in its simplicity, misses some things:

- Agile methods have a goal of continuously shippable code. Of course, "shippable" might not mean "ready to release" and cane mean something closer to "runnable," but you can get there by doing no work since the end of the last release. That isn't the goal of agile.
- With that large scale definition of "done" you have no way of tracking progress within a sprint.
- Without an agreement on what you're going to, it's hard to know when you are changing direction. And acknowledging change is an important part of being agile.

The last point about acknowledging change isn't just about "blame" for things not working out. It's about evaluating how well you understand both the business and technical aspects of your project, and it forms the basic for improvement.

True, having incremental definitions of done that you can use to track progress does help manage budgets. But that really is the least important aspect of having a good definition of done. Even if I were on a project with an infinite time and money budget, I'd want to have a sense of what our goals are. Having an agreement among all of the stakeholders on what being "done" means lets me:

- Improve communication among team members and between team members and business stakeholders.
- Evaluate my understanding of the problem and help me identify what I can improve.
- Set expectations so that it's easier to develop trust between stakeholders and the engineering team that the team can, and will, deliver.

"Ready for Release" is a key component of "done" and and essential part of being agile. But it's not enough.

Topics: agile, project planning,

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.

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

UPCOMING WEBINAR

LETS HANG!

FEATURED RESOURCES

AGILE
Implementing Agile eGuide
White Paper | StickyMinds

AGILE
Leadership for Today's Agile Team eGuide
White Paper | StickyMinds

AGILE
Scrum Master Essentials eGuide
White Paper | StickyMinds

Log In or Register to post comments

USER COMMENTS

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

| October 13, 2011 - 9:13am EDT

I blogged some reasons why I believe that "The only useful definition of done is that you approved it to release, in whatever form." In my opinion, most agile teams focus on assembling releases. It's up to the developer to get a feature into a release, which makes it "done", and also makes it useful. The rest is brownie points.

http://blog.assembla.com/assemblablog/tabid/12618/bid/69440/On-the-Impo...

Log In or Register to post comments

YOU MAY ALSO LIKE

AGILE

From Silence to Self-Organization: What Social Interaction Games Achieve That AI Cannot

Dinesh Sharma | June 17, 2026

AGILE

Why Your Agile Team Is Failing Even When Everyone Knows Their Stuff

Ella Mitkin | March 2, 2026

AGILE

Solving the Problem vs Treating the Symptoms

Payson Hall | January 16, 2026

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.