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

STICKYMINDS

Agile Testing DevOps Security AI

Definitions ARTICLE

By Steve Berczuk | November 22, 2011

SUMMARY

Words and definitions help with communication. And it's hard to come up with an unambiguous word. But try hard to use words as they are defined in your context. That will help you to communicate, set expectations, and maybe even be more agile.

The other day a colleague use the phrase *open interval* to refer to time periods that included their endpoints. This didn't sound like the definition that I'd learned in analysis class, which I will admit was not my favorite subject in grad school, so I searched for a definition and found:

An open interval is an interval that *does not* include its end points.

This led to a brief, but interesting, discussion about why my mathematically-minded colleague flipped the meanings. It came down to the feeling that the definition didn't make sense in terms of common english usage (when you open a door, or a group, for example, you're being inclusive).

Definitions add precision, but when you use words that have meanings in multiple contexts, you can find yourself in confusing conversations. Consider a term like normal, which can be defined as:

a lack of significant deviation from the average

This sounds pretty benign, but people often carry around more that just definitions when they hear words. Using "normal," for example, in conversation can get tricky if the group you are in has not established a baseline context; even though there is no value judgement inherent in the word, people might assign values to being more than a certain distance from the average. And then rather than than discussing the problem at hand you get distracted by a meta conversation.

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

Read More

practices that you need to follow to be able to say that you are using a method. Can you do Extreme Programming as a solo programmer when **Pair Programming** is a practice, for example?

The discussion is very interesting and useful, as it leads to an understanding about what is useful about XP practices. But from a definitional sense, I would err on the side of saying "No, XP is defined as the practices, but you can be doing something XP-inspired." The question you want to ask is "why does it matter if I'm doing XP (or Scrum, or another process)?" Maybe it doesn't. Unless you want to learn the value (and limits) of a practice.

The discussions get more interesting when it comes to concepts like "agile" which are very overloaded in different contexts. You can be more agile from a business perspective, yet not follow a defined agile software development method, for example. But if you want to say that your team is agile, be sure that you understand what your definition is.

Doing an "inspired by" process can be good if you are getting the results you want. When you are learning a process, it's **important not to customize the practices** until you understand why they are in place. And when communicating with others, being a bit fussy on the definitions (or qualifying your description) can be helpful.

Definitions help us to communicate. It's a useful default to stick with the definition in the domain. Then apply value judgements as needed. It seems better to say that you aren't agile, yet getting better results than you were, than to morph the definition of agile so that it fits your practices. Making "Agile" a checklist item itself is counter to the principles of agile. Define what you do in terms of your practices and evaluate effectiveness in terms of your results.

Words and definitions help with communication. And it's hard to come up with an unambiguous word. But try hard to use words as they are defined in your context. That will help you to communicate, set expectations, and maybe even be more agile.

Topics: personal 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

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

LETS HANG!

FEATURED RESOURCES

SECURITY

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

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

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.

See All Member Benefits

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