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

STICKYMINDS

Agile Testing DevOps Security AI

Software Engineering or Software Craftsmanship: Do We Care? ARTICLE

By Steve Berczuk | January 23, 2012

SUMMARY

Much like a tenet of agile software development is that "planning is more important than the plan," there are some questions about software development that are useful to explore, even if you can't suggest a good answer. One of these these is whether what we, as software developers do, is (or can be) engineering. Or should we call what we do software craftsmanship?

Much like a tenet of agile software development is that "planning is more important than the plan," there are some questions about software development that are useful to explore, even if you can't suggest a good answer. One of these these is whether what we, as software developers do, is (or can be) engineering.

I was talking with a colleague about a comment someone made about a post about lessons that software engineers can learn from artists. In this comment the poster raised an issue that comes up a lot when someone uses the phrase "software engineer." The question was, in essence:

Is what we do when we build software "engineering" or "craft?"

My colleague, who has a background in Naval Engineering and who is engaged to an Electrical Engineer (who designs and builds hardware) suggested that the difference is that one rule of thumb might be:

It's engineering if you need to use calculation (about physical constraints).

An example was that to build a bridge you need to do calculations to determine if the bridge, as designed, will support the load you want to support. Many of the "classical" engineering disciplines (Mechanical, structural, electrical) fit that criterion. But by that definition, some things which most would agree that are crafts, such as carpentry, involve calculations. So there is probably more to it that that.

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

Accept

Read More

builds a sensor to measure the performance of his home heating system is he doing engineering, or is that activity more like a hobby or craft?

As I think of the question, and the frequency at which it comes up, I wonder:

Why does it matter whether we are software engineers or software craftsmen?

Engineering, to me, implies a certain level of discipline are repeatable process, and all of the classic engineering disciplines are grounded in physical laws that we are fairly confident in for the scales at which they apply.

If the "software engineering" question is important to you, take a minute to understand why you are asking the question, as the reason for asking may affect the answer. You can bring discipline to building something you regardless of whether or not you are engineering. I don't think that the question of how (or whether) to make software into an engineering discipline is an unimportant one. But I do think that it is often asked in a vacuum, much like the an attempt at a user story that is missing the "so that..." clause.

To me, as a professional, it's more important that I bring discipline to what I do. I want to build useful, quality, software where the definition of "quality" that is often most appropriate is Jerry Weinberg's from **Quality Software Management: Systems Thinking**:

Quality is Value to Some Person.

It might be fair to say that we're not always "crafting" or "engineering" software when we sit down at an IDE and program. Sometimes we're just coding (or hacking, or programming). But if you (and your team) are committed to building software that meets your customers needs, and you are always working to learn how to work better, you're doing a good thing. And you don't need to call it engineering to acknowledge that.

Topics: software engineering, teams,

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

1 comments

| January 26, 2012 - 11:19am EST

I was educated as an engineer, and I define engineering as the application of scientific principles to providing solutions for problems. Software is based on scientific principles such as mathematics, logic and psychology, and provides solutions to problems. Therefore software is engineering.

Log In or Register to post comments

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

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.

See All Member Benefits

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