

What Is Quality?

by Mike Cohn •

57 Comments

Agile teams build high-quality products. Agile team members write high-quality code. Agile teams produce functionality quickly by not sacrificing quality

image not readable or empty
Quality Checked Stamped Seal

Each of these is something I've said before. And if you haven't said these exact things, you've likely said something similar.

Quality gets mentioned a lot in discussions about agile. And so, perhaps it's worth clarifying my definition of quality. Of course, others have thought about quality more deeply than I'm capable of. And so, I won't be providing a new definition of quality here. But I will explain how I think of quality.

One of the leading advocates for quality was Philip Crosby. In the 1970s he proclaimed that "quality is free" because doing something right the first time at a high level of quality was cheaper than fixing it later. Crosby defined quality as "conformance to requirements."

I never really bought into Crosby "conformance with requirements" approach (even before agile came around) because there was never a way to be confident requirements were accurate. Saying something like old [Microsoft Bob](#) was high quality because it complied with some ill-conceived requirements document never felt right to me.

Similarly, quality isn't just being bug-free though, as that's the same problem.

Another approach to defining quality comes from Joseph Juran. He was one of a number of management theorists who worked in Japan in the 1950s. Juran defined quality as "fitness for use":

"An essential requirement of these products is that they meet the needs of those members of society who will actually use them. This concept of fitness for use is universal. It applies to all goods and services, without exception. The popular term for fitness for use is Quality, and our basic definition becomes: quality means fitness for use."

This definition of quality really resonates with me. Quality is "fitness for use." A high-quality product does what its customers want in such a way that they actually use the product. Something that conforms to ill-conceived requirements (such as Microsoft Bob) is not high quality. Something that is buggy isn't high quality because it isn't fit for use.

What do you think? Is quality best thought of as "conformance to requirements?" Or "fitness for use?" Or perhaps something else entirely? Please share your thoughts in the comments below.

Posted: October 7, 2014

Tagged: agile requirements, quality

About the Author

Mike Cohn specializes in helping companies adopt and improve their use of agile processes and techniques to build extremely high-performance teams. He is the author of User Stories Applied for Agile Software Development, Agile Estimating and Planning, and Succeeding with Agile as well as the [Better User Stories](#) video course. Mike is a founding member of the Agile Alliance and Scrum Alliance and can be reached at hello@mountaingoatsoftware.com. If you want to succeed with agile, you can also have Mike email you a short tip each week.
