Doubt and Software Standards

Robert L. Glass

... in which I oppose the current uncritical view we tend to take on software standards.

Software standards. Now there’s a subject that brooks no “loyal opposition,” right? Standards are material provided by some software god, biblically significant, subject to no doubt? In recent years I’ve come to question all that.

How the Doubt Started

In 2006, my colleague Johann Rost wrote a guest Loyal Opposition column on the standard for requirements documents.1 He explained that at the conceptual level he understood the standard quite well, but at the implementation level he found it impossible to follow. I offered him sympathy, hosted his column here, and sort of forgot about it.

Time passed. Another colleague, Barbara Kitchenham, was beginning to struggle with certain software standards. One standard gave “inappropriate advice for measuring software engineering processes.”2 Another standard was “not suitable for measuring the design quality of a software product.”3 Echoing Johann’s concerns, Barbara commented that “experienced designers will be able to construct a number of different interpretations of the standard, implying that [it] is not a standard at all.”

Barbara pointed me at yet another critic of software standards, Magne Jørgenes. He noted that one standard for software quality “requires quality measurement and at the same time admits that there are no (universally) accepted quality measures.”4

A Heavy Folder

At that point, this topic became something I wanted to pay attention to. Generally, when this happens, I set aside a folder and drop into it any material pertaining to the topic. Eventually, the folder gains enough weight that it cries out to me to do something with it. That’s what happened here.

Perhaps the most powerful article stashed in my folder is “Unsafe Standardization.”5 In it, Martyn Thomas said that standards bodies “consistently produce standards that fail the basic criteria for good engineering.” He added that standards “should be based on established scientific results and best industrial practice” and argued that our contemporary software standards tend not to meet those criteria. To him, the fact that sometimes “excellent professionals have followed … standards and built systems that are … acceptable” was irrelevant. “Such achievements provide no evidence that the standards are either adequate or … cost-effective. Belief in such standards is superstition, not science or software engineering.” He concluded with, “We need change. … We need a fresh approach.”

My antennae were still out when I ran across “Standards: What Are They Good For?”6 In that article, Simone Santini was concerned primarily with software standards’ usefulness for software researchers. He argued, “Academics shouldn’t use existing standards in their research work. … These standards will not render a service to either academia or industry unless researchers actively try to destroy them.” However, no matter how much I puzzle over “try to destroy them,” I

Continued on p. 103
can’t make sense of it. Perhaps he means we need to evaluate standards carefully in order to destroy the unworthy ones?

Santini did provide a clue as to why standards can be so contentious. “In general, standards result from a bargaining process that involves different corporate economic needs, corporate cultures, and technical needs.” In other words, the (political) process of defining standards is hardly one of choosing them on the basis of established scientific results and best industrial practice.

Not everything I ran across was anti-standards, of course. For example, in a particularly laudatory article, Suzanne Garcia supported certain project management standards even while noting circumstances “when standards don’t help.”

What to Do with Standards

So, where do we stand here? Standards should be based on scientific results and best industrial practice. They should be subject to evaluation to ensure they really work in the environment for which they’re intended. All that is difficult because standards tend to be produced in a highly politicized environment in which corporate economic needs and cultures can take precedence over usefulness.

It strikes me that all of this adds up to one thing. As Martyn Thomas said, “We need change.” Both the process and nature of our software standards demand objective review.

I’ll bet this column will stir up a lot of controversy. After all, if you sew doubt in a field that some see as godlike and biblical, you’re asking for trouble. So be it, I guess. This issue demands an airing.

References


Robert L. Glass is editor emeritus of Elsevier’s *Journal of Systems and Software*, publisher and editor of the *Software Practitioner* newsletter, and a visiting professor at Griffith University, where he’s involved with the Australian Research Center for Complex Systems. Contact him at rlglass@acm.org; whether you agree with him or not, he’d be pleased to hear from you.