

Editorial Board Thoughts:

Doesn't Work

Mark Cyzyk

The proof of the pudding's in the eating.

Miguel de Cervantes Saavedra. *The ingenious hidalgo Don Quixote de la Mancha.*

Part I, Chapter XXXVII, John Rutherford, trans.

About fifteen years ago I had two students from Germany working for me, Jens and Andreas. Those guys were great. They were smart and funny and interesting and always did their best. I would send them out to fix things around the library, and they would dutifully report back with success or failure. I told them that, particularly if there was a problem with a staff workstation, “If it breaks in the morning, it must be fixed by lunchtime; if it breaks in the afternoon, it must be fixed by 5:00.” They understood that if a staff workstation was down, then that probably also meant a staff member was just sitting there, waiting for it to be fixed. If we had to we could slap a sign on a broken public workstation and get back to it later—there were plenty of other working public stations after all—but staff workstations must be working at all times. Insofar as we had an aged fleet of PCs whose CMOS batteries were rapidly giving out, I kept Jens and Andreas running around the building quite a bit. On occasion, though, they would report back with the dreaded, “Hey boss, doesn't work.” This was the one thing that would raise my ire. “Of course it doesn't work, that's why I sent you down there!” I would think. The phrase “doesn't work” became for me a Pavlovian signal that I was about to drop everything and go take a look myself.

It now occurs to me, though, that this notion of “work” is precisely the point of technology, and that sometimes this gets lost for those of us employed fulltime as technologists in libraries. Let me explain:

In my opinion and *for the most part*, the proper role of the technologist in a library is that of a consultant on loan to the departments to work on projects there, embedded.¹ Two of the best bosses I ever had said essentially the same thing to me in our introductory first-day-on-the-job chit-chat: “You report to me, but you work for them.” Such is the proper attitude in any service-oriented profession. Does this not frequently get inverted, subverted, lost?

What happens is that technology starts to take on an importance undeserved. It becomes self-referential and insular; a technology-for-technology's-sake attitude arises.

Mark Cyzyk (mcyzyk@jhu.edu) is Scholarly Communication Architect in The Sheridan Libraries, John Hopkins University.

But technology-for-technology's-sake is just wrong. Technology is merely a means to an end, not an end in itself. The word itself derives from the ancient Greek *technê*, most frequently translated into English as “craft” and frequently distinguished in the Greek philosophical literature from *epistêmê* or (certain) knowledge.² So it is here that the crucial distinction in the Western world between practical and theoretical activities is made, and technology is clearly a practical, not theoretical activity. As such, it has by its very nature practical outcomes in the world: technology *works* in the world. Technology is *instrumental* in achieving certain practical outcomes; its value is as a tool, instrumentally valuable, not inherently valuable. It is not for its own sake that we implement technology; we implement technology to get some sort of work accomplished in the world.

Our programming languages, application servers, Web application frameworks, AJAX libraries, integrated development environments, source-code repositories, build tools, testing harnesses, switches, routers, single-signon utilities, proxy servers, link resolvers, repositories, bibliographic management utilities, help-desk ticketing applications, and elaborate project-management protocols are all for naught if the final product of our labor, at the end of the day, doesn't *work*. Our product is not only literally useless, it is worse than useless because the library in which we labor has devoted precious resources to it only to result in a service or product that does not properly function, and those are precious resources that could have been spent elsewhere.

Hey there fellow technologists, why am I being so dismal? I would prefer the term “grave” to “dismal.” Significant portions of the library budget are put toward technology each year, and as those whose duty it is to carry our local technology strategies into the future, we need to always be mindful of the fact that each and every dollar spent on technology is a dollar not available for building our collections—surely the direct center of the mission of anyone who calls himself a librarian, A.K.A., a cultural conservationist. (Shouldn't we be wearing badges that read, “To Collect and Preserve”?)

Making it work is Job One for the technologist in the library.

...

A colleague and friend of mine once told me, a decade ago, that our fellow colleague made a snippy comment about an important and major Web application I had written, “Just because it works doesn't mean it's right.” Now, admittedly, I was a very sloppy Code Formatter, and yet I certainly would never say that the applications I wrote were steaming plates of spaghetti. On the contrary, I think the code I wrote consisted of good, solid procedural programming. What my disgruntled colleague meant, I think, was that I failed to follow a framework, and by “framework” he naturally meant the same framework to which he'd recently hitched his own coding wagon. My response to his snippiness was, “Ah, pretty-it-up all you want, organize it any-which-way, but functional code--code that *works*--is actually the Number One criterion for being Good Code.”

Just ask your clients.

That app I wrote has been in production, happily working away as a key piece of the enterprise network infrastructure at a prominent, multi-campus, East Coast university since 2002.³

REFERENCES

- ¹ And here I heartily agree with my fellow Editorial Board member, Michael Witt, when he notes that “[p]art of this process is attempting to feel our users’ pain...”, and I even extend this to the point of us technologists actively *working with* our users toward a common goal, literally sitting with them, among them, not merely being present to offer occasional support, not merely *feeling* their pain but being so invested in our common project that their pain *is* our pain. [Did I really just suggest we take on more pain?! Yep.] See: Michael Witt. “Eating Our Own Dogfood.” *Information Technology and Libraries* 30, no. 3 (September 2011) 90.
<http://www.ala.org/lita/ital/sites/ala.org.lita.ital/files/content/30/3/pdf/witt.pdf>
- ² I’m no classics scholar, but this is my recollection from taking a graduate seminar many years ago on this very topic. So while I’m not pulling this entirely out of thin air, I am pulling it from the musty mists of middle-aged memory – that, and a quick scan of Professor Richard Parry’s fine article on this topic in *The Stanford Encyclopedia of Philosophy*, particularly the section on Aristotle’s views. Regarding my comments below on technology being instrumentally valuable, I cite Parry’s words: “Presumably, then, the craftsman does not choose his activity for itself but for the end; thus the value of the activity is in what is made”. See: Richard Parry. “*Episteme and Techne*,” *The Stanford Encyclopedia of Philosophy, Fall 2008 Edition*, Edward N. Zalta, editor.
<http://plato.stanford.edu/archives/fall2008/entries/episteme-techne/>
- ³ Mark Cyzyk, “The Johns Hopkins Address Registration System (JHARS): Anatomy of an Application,” *Educause Quarterly* 26, no. 3 (2003). <https://jscholarship.library.jhu.edu/handle/1774.2/32800>