CIO

Print Article ☒ Close Window

From: www.cio.com

Cloud Development Documentation Insanity

– David Taber, CIO

December 13, 2011

Everybody knows that complex technology needs documents and training materials so that developers can effectively use it. In the cloud, this need is magnified by the fact that developers have to work with several languages at once (HTML, JavaScript, XML, CSS, jquery, Ruby, PHP, SQL...the possibilities are endless). So developers need more docs, right?

Time for another rant.

• The vendor people who are intimately familiar with the technology are not interested in writing documentation. If developers won't comment their code, you think they're going to write docs? Besides, their managers have powerful economic incentives to use tech writers who are experts at information packaging, but are pretty far away from being developers.

• One cloud vendor now has over 7,500 pages of documentation overall. A huge investment, consistently written, full of data...and little more than a starting point. The moment you try to really use a feature in a real-world system, it's off to Google and the user forums. Of course, there's enough misinformation on the Web that you'll waste plenty of time. But at least you get the comfort of knowing that other developers are lost in trial-and-error just like you are.

• There's that wonderful acronym, RTFM. Every support organization uses that as their answer to 50 percent of the incoming calls. And that's 'great', particularly when the vendor's docs are wrong, or simply doesn't cover the use case. I recently reported two bugs in a vendor's docs...after several e-mail exchanges, both cases were closed. No resolution, just the message to basically 'go pound sand: the vendor is never going to fix those docs.'

• Docs cover one vendor's technology. But in the cloud, we're all using a range of vendors' technologies plus a bunch of open source goodies. Nobody has the time to document the real environment we're working with -- it's a boil-the-ocean project.

• Docs are like dictionaries (remember, those big paper things&): the information is all there, but you can't effectively communicate just by knowing the syntax and word definitions. The way you really learn a language is by reading real writing (novels, journals, comic books...you know those paper things...)

or listening to newscasts, conversations, or songs.

So, tell me again why we're feeling the need for more docs? No developer wants to read them. No developer wants to write them. In the immortal words of Cool Hand Luke, "what we have here is a failure to communicate."

This isn't a matter of "documentation quality" or even quantity. I'm saying that doing big docs is the wrong answer to the question "how do we get developers effective with a new API?"

It ain't gonna be YouTube.

Training courses? Not bloody likely.

It's simple: Speak the language that developers around the world understand. Code.

A Modest Proposal (Google it...)

Vendors, divert 2/3 of the money you spend on writing developer docs in India to fund coders in India. Create a serious pool of code samples that actually do something (rather than the usual snippets that can't actually execute because they're missing that "obvious" constructor). Instead of writing about design patterns and best practices, show them in code (with enough comments for the novice to follow). Post working applications in demo systems that registered developers can get to. Show your stuff off.

Yes, the vendor's lawyers will whine about the risk of an attack from patent trolls. So put all those samples in open source. Vendors, invite your developer community to improve and extend your code samples -- particularly with cross-language examples using infrastructure that is outside of (but frequently used with) your API. Provide incentives for the best contributions (sometimes, all you need is to offer recognition on your blog).

And yes, some of your system integrators will be all bent out of shape because you've made it easier for the customer to be self-sufficient. But by democratizing the knowledge around your APIs, you'll be greatly growing the pie: there will be more successful customers and more economic benefit for all. In the long run, those same system integrators will have more to do, even though they won't get the easy kills from exploiting hidden knowledge about the basics of your API.

Customers, judge products by their code samples, and let the vendors know you're doing so. Ask the sales rep to get his SE on the phone, and grill them on where to find the repository of code samples for their product before you buy. Don't let them off the phone until you get that URL. If they don't have one, start laughing at them.

Humor is, after all, the best antidote.

David Taber is the author of the new Prentice Hall book, "Salesforce.com Secrets of Success" and is the CEO of SalesLogistix, a certified Salesforce.com consultancy focused on business process improvement through use of CRM systems. SalesLogistix clients are in North America, Europe, Israel, and India, and David has over 25 years experience in high tech, including 10 years at the VP level or above.

Follow everything from CIO.com on Twitter @CIOonline.