Over-Engineered vs. Good Enough.

Don MacVittie, 2008-13-05

The Boss and I were discussing the quality vs. timeliness debate yesterday, and then last night I was drinking a bottle of New Glarus (Cherry - the only brewed alcohol I consume, but that doesn't matter) - my first consumption of alcohol in 15 months - and looking at a model I was painting (An M4 GMC with mortar, but that doesn't matter either). The vehicle was nowhere near finished, but I was holding it and going "that looks good enough! I should be done!" The New Glarus was clouding my vision, allowing me to ignore the glaring unfinished and sub-par bits.

In case you're wondering, the M4 is not mine any more than the Belgian Red logo, it's a finished one from Military Modeling.

Sadly, it doesn't take New Glarus to get in this position. Most organizations are there every once in a while, ignoring glaring faults with their process or product (in the case of IT, 'product' is the service you provide the rest of the organization) because they've adapted one-or-the-other mentality.

The never-ending debate that in most organizations runs in circles goes like this - "we need to stop analysis paralysis" versus "we need to improve quality".

Want the truth? There is no panacea.

So if you're standing on one side of the fence prepared to defend it, you're on the wrong side.

But that's okay, because experience says you'll hop over that fence several times during the course of your career.

There is a place in the world for both, but very few projects warrant all of one or the other. I *want* my car over-engineered, thank you. I want mission-critical applications - in this case mission critical is "the company cannot run without them" to be over-engineered also. That app the sales managers use to calculate their bonuses? Not so much. For most organizations you want your public face to be over-engineered also. If your website is down, it is the 20th century equivalent of having no sales people on the floor to help prospective customers. But if you over-engineer every system in your data center, either you have a big staff or you're not getting anything done. That's just simple fact. Look around your data center, and take a second to marvel at how many applications actually run there every day. It's astounding.

Over-engineered implies too much time is spent on it - not possible on some systems like the US 911 emergency system. Good Enough implies that it was rushed out the door just complete (and stable) enough to be acceptable - and some companies have built multi-billion dollar empires on this very concept. For most of us, and for most of our applications, reality is somewhere in between. Something is indeed better than nothing - unless that something causes anger and disillusionment, or keeps something productive from being done - "because we have a solution we should work within".

So what are you to do? Make certain you're investing time and effort on the systems that truly are required should the worst occur because those are the critical systems. Remove politics from your analysis, and think about it in terms of "what would we need to function should things go horribly wrong?" the answer is a tiny subset of the systems you manage every day. Get those to be supported by products like ours (LTM for server-sized emergencies, GTM for data-center sized emergencies, etc), give them more slices of your timeshared pie, and only after you're comfortable that these apps are darn-near over-engineered do you worry about apps like the Janitors' cleaning scheduler. You're already doing this, but politics does tend to cloud the discussion, and what's important is not always obvious (I worked at a good sized enterprise once where the apps were well taken care of, but the databases... well, it was ugly because they weren't considered in light of the apps that required them until *the incident*) .

Focus on the important, remember that just because a particular business manager screams a lot doesn't make his app more important, it does say something about him/her though.

At least one of you is already thinking "not possible here..." I disagree. When you start talking about "protecting the business", people listen. They have to, you're talking about what's important to them too. And when you start talking about modifying processes (in general, not just DR processes) to better suit the needs of the business, people listen to that too. Because I've never been at the place where IT was perceived in a positive light, so any attempt at improvement will be well received.

There you have it, my advice of the week - (1) Don't drink New Glarus while modeling unless the model isn't important to you... It *will* be "Good Enough". (2) Over-engineer what's important, and make certain you've got a handle on what isn't.

Don.

/reading: Can't say, I signed a non-disclosure. But it's good.

/imbibing: Coffee and Mt. Dew, clearing my head ;-)

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com