Resistance is not Futile: Why Change is so Hard

Posted At : November 21, 2013 12:25 PM | Posted By : Mark Kruger
Related Categories: Business Of Development

CF Webtools does more than 3000 hours of consulting every month. As you might imagine there launches, releases and deployments happening constantly. One thing we run into constantly is *resistance to change*. When users are confronted with a new screen, new functionality, or (especially) a new system there is always resistance. It can range from a mild teeth gritting to kicking and screaming depending on the depth of change. Developers and managers are often nonplussed by this resistance. In virtually every case developers see the changes they have made or the systems they have created as *enhancements* or *improvements* over the "old way of doing things." They usually see resistance as futile and self-defeating - not to mention a little absurd. I think this is one of the reason's that developers often have a negative view of end users who are not technical. They simply don't understand the dynamics at play because they are not thinking through the human dimensions.

Great New Systems Still Face Resistance

Take a deep breath and listen to the muse - resistance to your improvements is not based on the *quality, appeal, innovations or the time saving nature* of your improvements. In other words, the fact that it's the greatest thing since your mother's apple cobbler is not going to make users like it or want it. See if this rings true for you. You are presenting a new system to stakeholders. Say you re-engineer the process for approving a manufacturer's wholesale orders. You create a slick application that interfaces with the companies ERP system. You add approval gates and requirements that must be met to move the process forward. The CEO is ecstatic. It used to take 2 weeks to get an orders done because so many folks had to sign off on the pricing and the deadlines. Now the request won't sit around in someone's inbox or on their desk. The system will move it forward and acquire the needed vetting and approvals. Decreasing the time it takes to get bulk orders approved improves cash flow and the bottom line.

Sales folks are unhappy about it however. Why? Doesn't it mean faster commissions, more time for sales? Well maybe, but what you will hear from them is *"We have always done it this way."* Let's call that the WHADIT Way. Now before you get all huffy and accuse them of intransigence you should look a little deeper. There's a good reason that folks fall back on the WHADIT Way and its Cousin the WNDIT Way (i.e. "We've never done it that way before"). Consider for a moment how regular *users* of a system differ in perspective from you. When you got your new IPhone it was a splendid day right? You spent hours noodling with it and figuring out all the bells and whistles. That's because as a developer or IT pro you are a *technology adapter*. Far from being intimidated by new systems, hardware, phones and devices, you embrace them and revel in learning how to make the most of them. It's not a trial for you to learn. Indeed it's only a *minor* investment for you. Why? Because your day is filled with climbing up to the cutting edge of technology. You are *oriented* toward the new.

Now let's talk about the broad masses that include everyone else. Yesterday my wife (who is *not* technical) was frustrated trying to send a picture from her iPhone. She has a 5s and she was sending via email to her own email inbox. She was doing this to get the picture from her phone to her computer. She would take a picture and when she wanted it on her computer she would forward it to herself, then check her email on her computer to pull it up. Because of some network issue or whatever the email would not leave her outbox. I said to her, "Use the synch cable and copy it directly." The fact that she could do this was news to her. Given one solution to getting a picture out of her phone, it did not occur to her that there might be several.

So here's the question. What is it about *me* (and *you* Muse reader) that is different from my wife Ann? Why did I have a ready solution? Is it just because I'm smarter? I can tell you that this is not the case (my wife is extremely smart and savvy). The answer is that I envision technical solutions to problems and I assume that such solutions exist because "that's what I would have done" if I was building a UI, a site or an interface. I knew that the cable would work of course, but let's suppose I did *not* know. I have no doubt I would simply *assume* that there was a way to connect and copy images off of my phone. Why? Because it "stands to reason" - not Ann's reason or a regular user's reason - a technology worker's reason.

But this is not the case for the majority of folks who have to use your system (unless you are building it for IT, in which case they will pick it apart long before you get to brag about it in a meeting). Most people can't make leaps and confident assumptions about what something should do or can do or ought to do. They are tethered to what they know. They have made a *major investment* in knowledge to get things done surrounding their job. This knowledge might be how to fill out forms or which requests should go first or who to contact to get prices changed or how to navigate a legacy menu. It will almost certainly include some knowledge that they feel makes them important and is a source of *status*.

This idea of status is one we often forget. Consider how your technical knowledge makes you feel about yourself. Doesn't it heighten your sense of worth at the workplace? When people stop by your desk to ask about their hard drive or printer you get exasperated but inside aren't you secretly gratified that they depend on you? Non technical users are not so different, they just have different realms of knowledge. The WHADIT Way is really a ritual that binds users together. The current process might be byzantine and require lots of hoops, but knowing how to get it done is part of the power invested in competent employees. Once that power resides in an automated system it no longer requires special knowledge.

So new systems very often have this downside - *they diminish the sense of value that employees feel when doing their jobs* because they take things out of their hands.

So the Muse always recommends to board room types that they take a different approach when implementing new systems. Here are the Muse tips for stakeholder buy-in.

Campaign

Start with a sort of marketing strategy. Advertise the new system. Gather testimonials from pilot users. Put screen shots in the newsletter. Find a way to project a positive image for your new system prior to roll-out. This will make adoption easier and it will be harder to criticize.

Engage Early

Early on in the process of outlining the new system, engage your stakeholders and get their input. Make sure the system is not just solving problems that are seen by management. Get real input from users and solve their problems as well. If you get early engagement from the end users they will feel invested in the outcome and grease the skids at release.

No Implementation By Fiat

CEO's and CIO's are famous for saying "They'll just have to live with it." This simply never works - at least not in the U.S. Here we value creativity and innovation. We are looking for thinking, energetic employees who solve problems, not automatons who do things by rote. Valuing creativity and innovation comes at a cost for the manager. He or she cannot afford to force feed employees a solution. When it's tried it is a matter of weeks before there are workarounds and alternate paths for tasks that circumvent the new system. Instead, managers must find a way to:

- Insure that all voices at the table are heard.
- Find ways to alleviate concerns by stakeholders and users - and do it in a way that makes it clear you are investing in those concerns because of the input given by those individuals. In other words, they have to feel empowered to make a difference in the solution.
- When changes in responsibility are needed due to automation, find ways to organize responsibilities, titles and job descriptions to take the sting out. For example, if your customer service manager can no longer approve an RMA without a new gate, give them the ability to provide free shipping or incentivize staff in some other fashion. The idea here is to provide for a *lateral move* with regard to status and responsibility.

Conclusion

In reality user Buy-in is probably more important than the slickness or usefulness of the system itself. So for all you techies in my audience, have a care with those users. Remember who writes the checks in our world. Take time to help them out.