

# Why you should care what your users think about your killer application

Posted At : August 4, 2005 12:57 PM | Posted By : Mark Kruger

Related Categories: Project Management

You and your crack development team have just built the greatest killer Ap since VisaCalc and it's going to save your client millions. You've taken a process that used to consist of printed paper and sticky notes circulating the various floors and cubicles and turned it into a sleek race car of automation and productivity. You are so excited to show the final version to the various stakeholders in a big launch meeting. This is the equivalent of making your mom that plaster of Paris handprint when you were a kid and waiting with bated breath while she sees it for the first time. Will you get the accolades you seek? Will the users be as thrilled as they *should* be? Well...

The answer is *not necessarily*. As anyone who has tried and failed to throw away a pair of worn-out but comfortable sneakers, change is hard - even good change. As developers we believe in technology. We believe in the power of technology to alter and improve the way business is done. Regular users are not focused on technology as a solution. In fact, they often see technology as an inconvenience. Consider this business axiom.

*"There is company policy, and then there is the way things are **actually** done."*

If you are a CEO or a President of a company (or sometimes even the COO or CIO) you should stop reading right now and remain blissfully ignorant. For those of you *this* planet you are probably already aware that folks find their own groove when doing their job - and they get comfortable with it. Sure, they know that the product sample should be finished and tested before contacting the art department, but they also know that Jill in graphics and Jack in marketing can "fake it" and get them proofs so they don't have to sit on their hands. They know that the proper way to get a PC fixed is to fill out the help desk form. But they also know that Bob in IT likes to be accommodating - and likes wandering the building better than sitting at his desk, so they give him a ring. Often, they see themselves as ingenious - the "goto" guy or gal.

Now, here comes your productivity application that is going to "streamline" the process. Do yourself a favor and every time you are talking about your application and you say the word "streamline" or "organize" or "automate", replace it with the word "restrict" in your head. Pretty soon you will see a different picture. You are taking away some of the special knowledge of the inner workings of the company and replacing it with a soulless application. To the user, your application means they have to learn a set of rules that may have no work arounds. They see their sphere of influence shrinking.

Now, I'm not saying unequivocally that users are unwilling to change. I'm being somewhat facetious to emphasize the point that (with a few geeky exceptions) they will not have the energy and enthusiasm for your application that you have. You have to tailor your approach to them - and you have to get the right people involved. The problem is more serious than you think. You may have heard the statistic that 70% of all software projects fail. In my view, many such projects fail because they lose inertia. To bring a project to launch *and have people adopt the new application* requires interest at all levels. Sometimes a new application is foisted upon

unsuspecting users by managers or CIO's who see a need to "bring business processes in line" - but they haven't spent any time selling the idea. Here are a few tips for making sure a project is adopted.

## Early Stakeholder Involvement

If you are building an application with input from a VP and CIO and no one else, you are in for a big let down. Somewhere in the early stages you need to convince the principles to let you talk to actual users about the proposal. Don't waste time explaining what your application purports to do at this point. Instead, ask questions about how certain tasks are accomplished. Let the users walk you through the process as it exists today. Ask questions that delve into how they do their job. If you are a people person, try to sense where users find their greatest sense accomplishment or expertise. Ask dumb questions that allow stakeholders wax eloquent.

People love to talk about their jobs, and they rarely get a chance to do it in a detailed way to someone who's actually interested (cocktail parties notwithstanding). Without a conscious effort you will find that your own view of the project will change. You will be more likely to create tools and processes that truly assist users in doing their job. You will be less likely to miss important aspects of your application that would not have been evident in discussions with the CIO or VP. In the process, you may become an informed novice or even an expert on something you previously knew nothing about - and that will serve you well in future projects.

## How to Pitch Your Application

When you come to that initial launch meeting, come prepared to do more than show off the bells and whistles that you feel great about. Instead, come prepared to be humble and apologetic about how work life will change with your new application. Sympathize with users when you sense that they are uncomfortable with the new approach - or when they show insecurity at learning all the features and flow of your application. You have to learn to see yourself as a salesman at this point. Your job is to convince the users that your application will save them time and make them more productive. You have to make them believe that on balance, your application is a *good* thing. Here's a tip, this is not a "hard sell". Don't go in guns blazing and blow them away with how great everything is. Prepare yourself to soft-sell. Don't be afraid to use phrases like "It is our hope..." and "the idea is..." You want the *users* to see it as *one small step for them* even as the CIO and VP see it as *a giant leap for the company*.

## Campaign Management

My final tip is quite practical. Work with the principles to develop an sales campaign for you new application. This could be posters and flyers, regular emails highlighting it's features, quickie demonstrations at departmental meetings or even a few web pages designed to tout it's benefits to the company and department. Even if your application is the most fabulous and incredible thing since God made Eve it still takes 40 to 60 days for a new application to be fully adopted by a company. People forget the URL. They forget how to login and what to do next. They don't know where the help files are. They find bugs you hadn't thought of (you can bet on it!). During that time they are less likely to "feel good" about your application. They are tempted to go back to the old way of doing things. Some reminders can help the users remember the big picture and keep some positive energy pointed at the application

I know it sounds crazy, but the most forgotten part of the equation is the people part.

Being a developer who considers the "people" part of the equation is often what's sets developers and development companies apart from the rest of the geek squad.