

[Print](#)

Five Easy Tough Questions You Should Ask

By: Ken Hanley (04 Apr 2003)

IS Guerrilla

Start any major systems initiative, and like it or not - and I know a whole bunch of technical people who don't - you're going to be dealing with people and people issues. I'd go so far as to say that people issues are far more likely to deep-six any systems initiative you're involved in than troubles with the technology, but you already knew that. So what to do about it?

I suggest a five-question checklist. Yes, that's right, another checklist, one more piece of paper, one more small cog in the bureaucracy. But this one is important.

On top of all the other checklists you're sure to be putting in place (you are using checklists aren't you? As part of your basic project documentation? To make sure that something important isn't left out e.g. Has the server been ordered? Is the application upgrade complete? Has the test database been loaded?), I suggest you add the My Favourite Five Questions that Help Save my Backside' checklist, as follows:

Question 1: Can you help us with this? If you want/need co-operation on any project, and who doesn't, this is the single most powerful question you can ask. Fact is, people like to be asked to help; it makes them feel involved, it makes them feel important and it demonstrates that you respect the person you're asking. Most people won't turn down a direct request for assistance, and because you're consulting them, they'll feel more "bought in" to the whole process, especially if they can see that your project is better for the help they're providing.

One thing to watch for: if you ask for help, and it's offered, make sure you don't ignore it - seeking assistance and then not using it is almost as bad as asking someone's view and then completely ignoring what they say.

Question 2: What can I do to help you? This is the other side of the coin, and also a powerful question. No matter what level you're at, or how close to or far from the systems you are yourself, you'll find that your help will probably be, well, helpful, and maybe even absolutely critical to the success of the initiative. Even if by simply agreeing to be one of the first testers of a new application, even if you just volunteer to come out to a presentation and support the project team, your offer of help will be appreciated and remembered.

Question 3: Who else should we talk to about this? Here's a question you want to ask if you're in the thick of things. In my experience, stakeholder management, or rather, lack of stakeholder management, is one of the key failings of most systems projects - more people are affected by what you're doing than you generally realize, and more people can positively or negatively affect the outcome than you'd expect. It's your job to make sure that all these people are in the loop, that they're listened to, that they get a response that will make them a supporter, or at least not a strong opponent, of the initiative you're pursuing. Besides, these "other people" we should talk to probably know a few things about your project and the people you need to make it a success that you don't, and they may only share what they know if you go ask 'em.

Question 4: Is there anyone who's done something like this before? It's amazing to me how many systems organizations, no matter how many projects they've done, act like every new

one is the first one. Little in the way of repeatable process, no going back to lessons learned and very little seeking out the wisdom of those who've gone this way before. Because the application or the technology is different, we assume that what happened before isn't applicable to what we're doing now. That's just plain wrong. Almost every project that came before has something to teach us, and so do the people who were involved - nothing we're doing is truly new, and the people who have been there before probably have something to say, and would be glad to be asked to share what they know.

Question 5: What happens if we don't do anything at all? An often-overlooked, out-of-the-box question - too often we get way down the track on an initiative before asking ourselves about the "do nothing" option. We know that doing something will cost people and money, and that the outcome is probably somewhat uncertain. What if we just leave things the way they are? What if we don't replace the old application? How much will it cost/what are the risks in the status quo? Sometimes a hard look indicates that standing pat is truly the best solution.

Five simple questions with answers that often have a huge impact. Easy to ask 'em, tough to deal with the consequences if you don't.

Hanley is an IS professional in Calgary. He can be reached at atisguerrilla@hotmail.com

Copyright © 2007
ITworldcanada.com