When Users Don't Know What They Want

Identifying software requirements without scaring off business users.

By Vin D'Amico, Principal
DAMICON, LLC

This article originally appeared in *ComputerWorld* on February 10, 2004

The infamous "user community" for your IT project doesn't know what it wants, so how can you? Like it or not, you must pull ideas out of users' heads and move everyone toward a common goal. It's not easy, but it sounds much tougher than it is.

We all know that most IT projects fail and that insufficient requirements analysis is the most frequently cited cause. But think about this: Analysis requires communication. IT people tend to speak in terse, acronym-filled sentences. Business people tend to talk about their problems and responsibilities. Neither side truly understands what the other is saying, leading to false expectations and broken promises. How can you possibly analyze requirements in this environment?

IT professionals will argue that the business people just don't know what they want. That's partially true. How can they be expected to know what technology can do for them? Advancements in the field occur so quickly that what was almost impossible yesterday is commonplace today -- such as the ability to e-mail photos instantly using a mobile phone. The business folks, in turn, will argue that the IT crowd isn't focused on their needs. True again! Too often, IT is overly focused on a technology set such as .Net or J2EE. Business people don't care. Just give them a solution that works reliably!

So how do you cross this chasm and get to a common ground?

One long-standing approach is to create a requirements document several hundred pages long and get all the major players to sign it. I'll bet many of you have been there and done that. It doesn't work. Such documents are impossible to fully comprehend and are subject to widespread misinterpretation even if you can get key players to take the time to read them. They're also a nightmare to update as requirements evolve. Maintaining such a document becomes a full-time (and costly) job.

How about creating a monstrous spreadsheet with all the features listed, along with priorities and stakeholders? Have you done that, too? It doesn't work either! Technical professionals tend to like this approach because the spreadsheet serves as a checklist for implementation. Unfortunately, most people have trouble piecing together discrete functions into a coherent application. Thus, their expectations of the final result will be wildly different.

Instead, **you should get into a room together and talk about the problems to be solved**. Don't concentrate on getting every detail in writing. Focus on extracting ideas, issues and needs. Communicate!

For example, if business users say they want to store information in a database rather than on paper forms, you must uncover the reasons why. Are they running out of physical space to store forms? If so, perhaps all they really need to do is scan the forms to create image files and store those images on a file server.

Is the goal to continue filling out paper forms while simplifying searches for information in the future? This is a more complex problem requiring optical character recognition or online data entry with the results stored in a relational database.

Or do they want to enter and circulate information electronically and eliminate paper forms entirely? This is the most complex problem, requiring extensive design sessions. Business processes will have to change, and some level of retraining will be needed.

Although the answers to these questions are clearly related, the emphasis makes an enormous difference in the nature of the solution. A high-speed, real-time data entry system is vastly different from an information archival system requiring occasional document retrieval.

Of course, you can hear it now: The IT group complains that the users want everything. Sure they do. But it's the responsibility of the IT group to educate and inform. Talk to them about the trade-offs. Show them the risks and complexity inherent in trying to do too much, too fast. And don't forget about cost and time. There's probably not enough of either to do it all.

Now that you're getting down to the real needs and problems, start drawing. Move to the whiteboard and rough out a solution set. Crude? Yes. Effective? You bet! **As you begin to capture major ideas and process flows, start to prototype the user interface with whatever tools you have available.**

If you're good with HTML, a browser-based user interface can be created quickly and inexpensively. You might choose a visual development tool, such as Visual Basic, or even a PowerPoint presentation (yes, it's been done). Regardless of the tool, the goal is to mock up the major functions of the software in such a way that everyone can "see and touch" the result. Now you have multidimensional requirements. These have much more value than thousands of one-dimensional words organized onto reams of paper.

Whether your goal is to build a custom software application or purchase a commercial package, this effort will be rewarded. In the case of your software development team needing to build the application, the team will have a model for starting the build process. This model can readily be turned into "use cases," if you're using a methodology like the Unified Process, or stories, if you're a fan of extreme programming.

If your goal is to purchase a package, you've got a structured demonstration that you can provide to software vendors. Now ask the vendors to give you a demo of their software, showing, specifically, how they'll provide the desired functionality. They won't be able to provide an exact fit, but that's OK. Select the vendor that comes closest -- while meeting your cost and business goals.

Business workers don't need the latest technology. They don't care about standards wars or programming languages. They need software they can really use. Give it to them!

Vin D'Amico is Founder and President of DAMICON, LLC, your ADJUNCT CIO™. He is an expert in using open source software to increase worker productivity and reduce IT costs. He has experience at industry leading companies such as Keane, 3M Touch Systems, Kronos, NetManage and Wang. DAMICON provides Help Desk Design, IT Operations and Change Management services. Vin can be reached at vin@damicon.com or by visiting www.damicon.com.