Opinion

Stop 'Gathering' IT Requirements

By Paul Glen

October 6, 2003 12:00 PM ET
Computerworld

Over the years, I've come to the conclusion that one of the most destructive notions circulating inside technical groups involves "gathering requirements." For decades, virtually everyone in the industry has accepted that the first phase of every IT project should be to gather requirements from business users. At least in theory, it should be the point of departure for all our efforts. (Of course, it's also the phase of the project that's most often skipped.) So now that our success rate for IT projects has risen to the still-dismal level of about 25%, perhaps we should question some of this time-honored wisdom.

As I travel the country for consulting and speaking engagements, I often ask, "What are the main causes of project failure?" And invariably one of the first answers is something like,
"There's a failure to gather good requirements."
And when I ask why projects get bad requirements, the answers are,
"Users won't tell us what they want," or
"We don't ask good questions," or
"What they told us they wanted turned out not to be what they really wanted."
But I think that the problem is more subtle than any of those answers.

The problem with gathering requirements is right there in the word gathering. What images does it conjure? For me, it's an image of a harvest, of a group of people standing among endless rows of vines picking ripened grapes and carefully placing the bunches in a bin. For others, it might be an image of a child collecting seashells on the beach or of a group of people huddled together at a town meeting. What's common in all these images of gathering is that there's something out there to be collected, like crops, shells or people, and that those things are already whole and complete.

So if we're gathering requirements, we assume that they must be out there, ready to be assembled like a roll of coins. Our problem is finding and selecting the right ones. So if the users don't tell us what they really want, we should grab them by the ankles, hold them upside down and shake them until those pesky requirements fall out on the floor. The only logical conclusion is that if we don't get good requirements, we haven't shaken enough.

Of course, this is ridiculous. Requirements don't exist out in the ether just waiting to be discovered. They aren't out there whole and finished. Clients and users aren't playing an expensive game of hide-and-seek with us. Usually, the clients' pockets are empty. Most of the time, they don't exactly know what they require. And even if they do, it's in the form of incomplete and inconsistent ideas that can be only partially articulated. Projects rarely start out with clear objectives or requirements; they begin in confusion and ambiguity.

The other problem with gathering requirements is that it suggests subservience or disinterested passivity on the part of the IT group. It gives the impression that the job of a technical team is to take orders like a waiter who couldn't care less what you eat and then deliver the cooked meal. Technical teams with this attitude rarely deliver high-quality service.

So what's the alternative? Obviously, projects need solid requirements on which to build technology. We should think of a set of requirements as being like a multilateral treaty among a group of nations. Representatives of nations negotiate treaties by seeking out points of agreement, acknowledging constraints, compromising and trading off. The forging of a treaty is an active and difficult process of creation. No one would suggest that you "gather paragraphs" to write a treaty.

So we must negotiate requirements among the many stakeholders whose positions and interests need to be acknowledged. There are the business interests of executives who fund projects, of course. There are the utility needs of the users who will work with the systems every day. There are also the technical needs of those who build, deploy and support those systems. The list can go on and on.

Successful projects begin not with a harvest, but with a difficult set of discussions on what should be done. If you stop trying to gather requirements and start negotiating them, your projects will yield richer crops.

P***aul Glen*** *is the author of <\i>Leading Geeks: How to Manage and Lead the People Who Deliver Technology (Jossey-Bass Pfeiffer, 2002) and principal of C2 Consulting in Los Angeles. He can be reached at info@c2-consulting.com.*