



# COMPUTERWORLD



GUIDE TO

# working with *non-geeks*

# GUIDE TO working with *non-geeks*

## Computerworld's Guide to Working with non-Geeks

**A**RE YOU A GEEK? If you are, this won't come as a news flash for you: You and the users you work with on the business side are pretty much nothing alike — not in the way you communicate, not in the way you value various traits, and not in the way you relate to work.

If you're not a geek but you work with them, you probably are just as aware that geeks and users can seem sometimes to exist in parallel universes. You might be working on the same project, but there's a good chance that the geeks and the non-geeks are going to have very different ideas about what is important. They will say things to each other that they don't have the tools to interpret. Their differences can make it nearly impossible for them to see that they both actually want the project to succeed.

Paul Glen has been observing the interactions of geek and non-geeks for years, gaining insights along the way about the ways that their differences can cause them to work at cross purposes or simply misunderstand each other's motives. We have gathered here several of his columns in which he tries to help the two sides bridge the gap that separates them and come to understand why they say the things they do. Included here is advice for geeks to speak in terms that business users will understand, for business users to better understand why geeks say things that seem counterproductive to the users, and for managers who need to motivate geeks and get them to work well with others. No matter which side of the geek/non-geek divide you fall on, you're sure to find it instructive.

— Jamie Eckle is opinions editor at *Computerworld*

## contents

- ➔ Being right vs. not being wrong **3**
- ➔ Talking to the Business: Our Problems, Their Visions **4**
- ➔ When Techies Speak, the Devil's in the Details **5**
- ➔ We techies can put users on edge **6**
- ➔ 'Nothing but the facts' approach just won't work with business people **7**
- ➔ Techies and users are in a vicious circle of mistrust **8**
- ➔ The hazards of literal listening **9**



**Paul Glen** is the CEO of Leading Geeks, an education and consulting firm devoted to unlocking the value of technical people. You can contact him at [info@leadinggeeks.com](mailto:info@leadinggeeks.com).



by Paul Glen

# Being right vs. not being wrong

**W**E GEEKS have a reputation that we neither want nor entirely deserve. To a lot of people, it seems as if we always have to be right -- to prove that, no matter the circumstance, we know best.

I believe that's a false impression, but it's easy to see how it came to be. Some of the most common complaints about technical people are that they interrupt with condescending corrections, become impatient when they have to explain things, qualify every statement so that it is precisely correct and dismiss unsupported opinions as invalid. If you aren't steeped in the psychology of geeks, those behaviors can sure support the idea that geeks just always have to be right. Even discussions among geeks look like knock-down, drag-out fights. From the outside, we can seem like a bunch of egotistical blowhards trying to one-up one another at a meeting of the Always-Need-to-Be-Right Club.

The thing is, though, that most geeks would read that description and say, "I hate those kinds of people. There's no way geeks are like that." And my experience has been that the vast majority of geeks absolutely are not. But we sure look that way.

Why, then, do we give such a false impression? My thinking is that geeks are horrified by the thought of being wrong. That might seem like an excessive attachment to being right, but

those are actually two very different mindsets.

Geeks revere truth and loathe lies, mistakes and partial truths. Our conversations are usually collaborative attempts to find, reveal and articulate objectively verifiable reality. We can't allow mistakes or partial truths to be left unchallenged. We passionately pursue dispassionate objectivity. Unfortunately, our commitment to truth is hard to distinguish from an egotistical need to be right. But there is one important behavioral clue that demonstrates our much more noble intentions: It is rare for a geek to continue to argue for an idea that has crumbled in the face of hard truth. We will explore our own ideas until they are proved wrong or confronted with superior logic or elegance. In the end, we are committed to being on the side of right, not to being right.

If you want to minimize the chances that you acquire the reputation for having to be right, here are a few phrases that might help. (You probably rarely hear them around the office.)

• 'You're right, and...'

Most geeks aren't in the habit of explicitly acknowledging that they've heard things they agree with. Instead, they latch onto the points of agreement and refine, clarify or qualify them, leaving the impression that they need to be right.

• 'Let me make sure I understand.'

Before launching into a well-reasoned refutation of someone's statement, make sure that you really know what is meant. There's nothing more annoying than listening to someone launch a long-winded attack on something you never said.

• 'I really like your idea.'

This may be the most important phrase of all. It lets people know that you personally value their insights and contributions.

Given how much we geeks hate it when people egotistically promote bad ideas, we really should invest a bit in separating ourselves from such behavior -- even if it's just the impression of that behavior. We might even improve our relationships and make our jobs easier in the process. ■



by Paul Glen

# Talking to the Business: Our Problems, Their Visions

**T**HE FIRST meeting for a project is a tense affair. There can be a lot of new things coming at you all at once. New co-workers. New technology. New processes. And, perhaps most problematic, new business partners.

These meetings tend to follow predictable patterns. You, the technical person, want to stick to a process, gathering basic requirements that can be put into a document. So you ask some questions about what's going on with the business and what problems need to be solved.

The business person talks rather vaguely about what she wants to accomplish. She seems excited about something, but you don't know exactly what. So you ask more questions to try to understand.

Before long, frustration creeps in. She can't comprehend why you don't "get it." And you still have no idea what she's talking about.

This disconnect occurs because you both think about work differently. We in IT see work as the act of solving problems. Business people see it as the act of achieving a vision.

A nontechnical colleague and I have been exploring this difference.

Problems have a very specific structure that we bring from the world of math into our work lives. They include problem statements, assumptions,

rules, constraints and solutions. These conceptual tools are powerful because they let us focus on what we're trying to accomplish and formulate detailed approaches to implementing solutions.

Visions don't have as rigid a structure. They are imagined futures -- they're visceral, not analytical. Business people imagine the experience of using the product, its feelings and its effects.

These opposing conceptualizations of work interfere with our ability to collaborate in two key ways.

First, we have trouble planning together because we orient ourselves toward the same work from completely different perspectives.

Problems are rooted in the present. They start with the current reality. To solve the problem, we work forward, plotting a course from the problem statement to the solution, and navigating the assumptions, rules and constraints.

Visions are rooted in the future. Achieving a vision requires working backward from the imagined future and figuring out what is needed to make that future real. As a vision transforms from vague to vivid, a detailed path to creating it emerges.

When this disconnect occurs, we have serious problems planning together because we approach the exercise from opposite ends of the project.

Second, we have emotional reactions

to each other's approach. For us in IT, problems are wonderful. Solving them is our life's work. They are invitations to think, build and create. And to us, visions are annoyingly vague ideas requiring clarification to become actionable.

For business people, problems should be avoided at all costs. When we turn their beautiful visions into problems, they feel like we are dragging their ideas through the mud. And when we ask questions, we interrupt their flow, preventing them from clarifying their visions.

To make sure early project meetings go well, don't try to force those visions into a problem structure too soon. Give the business people time to envision the work completely, and support the process by talking through dependencies. At the end of the session, synthesize what you've heard as a problem statement. It is possible that both parties just might get what they want. ■



by Paul Glen

# When Techies Speak, the Devil's in the Details

**E**VERY IT professional has been here: A business person asks you a question, and your thorough answer just isn't good enough. You try to give more specific information in an attempt to break through the communication barrier. But the more you try, the worse things seem to get. In the end, the business person is seething with impatience, so you start to get confused and angry.

Both parties walk away from such encounters convinced that it's hopeless to communicate with "those types" of people. They both say of the other, "They don't get it." And as a result, the business people stop asking questions, and we, the IT people, are relieved.

I'd grown so accustomed to this type of disconnect that I had come to see it as a natural part of working in organizations. But a recent interaction made me realize that it is one facet of a deep divide between business and IT, and that understanding the root of the disconnect is crucial to resolving it.

I was talking to a smart and articulate business person. Our conversation was following the seemingly inevitable course toward disconnect until we took a step back and examined what was at the heart of our conflict. The conclusion we came to was fascinating: It wasn't the language that was dividing us; it was the fact that technical and nontechnical people have completely

different perceptions of what constitutes a good answer.

The frustration that these conversations normally produce arises from our completely different understandings of what is true in the world -- as well as our incompatible ways of thinking about truth, identifying it, defining it and feeling it.

The business person had asked me a simple question about a project we were working on, and I had responded with an itemized list of the six key elements that related to the question. As I introduced each item, she became increasingly agitated. I had seen this reaction many times before, but it had never made any sense to me. After all, I was doing a great job of explaining things; my list was quite complete.

Because, for me, a technical person, details reveal truth. Broad, sweeping statements are vague and untrustworthy, nothing more than assertions. They need to be deconstructed, clarified, qualified or proved before I believe that they embody any form of truth. Analysis is my chosen method for discovering truth. Big things are broken down into manageable components that are then examined individually. If the parts cohere, then the whole makes sense.

But for her, a nontechnical person, details cloud truth. Broad statements give her a handle on what's at stake, so she can test the truth of an idea against

her internal sense of what's what. Essentially, she wants to "feel" the truth. Too many details interfere with her ability to process the explanation, thus preventing her from "getting it."

In my answer, I focused on completeness. That's my form of truth. To her, it felt like I was being evasive, and that I was overly concerned about trivial matters and unaware of what was important. If I had offered a high-level answer that focused on one or two key ideas, she would have been much more receptive. She explained that she would have eventually been interested in the details, but only after she had gotten the gist.

So, think twice before you assume that a more detailed explanation is a more credible one. You might be talking to someone with a very different experience of what it means to "get it." ■



by Paul Glen

# We techies can put users on edge

**WE TECHIES** need to take the edge off once in a while.

As I've sought to improve the way we communicate with nontechnies, I've recognized that we often resort to a surefire way to confound, if not irritate, them: We talk about what I call edge cases.

I do it all the time. Say a colleague makes a suggestion. If the idea sounds good, I start mentally wandering its edges, testing its validity. When I find an edge case where the idea wouldn't work, I blurt it out, wanting to show that I am giving the suggestion my full attention. To me, the edge case could indicate that there will be things that we'll need to address that aren't immediately apparent, or it could help prevent us from pursuing a path that would ultimately prove fruitless.

To my surprise, though, my colleagues get upset. That can be confusing to the techie in the conversation. Isn't it helpful to pursue new ideas with logic and discover the areas where they might fail? We think it is. But as is so often the case, we just aren't able to see things the way the nontechnical folks do. As far as they're concerned, we might as well have just greeted their idea by saying, "Well, it won't work when there's a full moon on a Tuesday." They can't help but feel that we are being deliberately negative and unhelpful. We seem to be disruptive to the flow of idea generation, dismissive

of the potential advantages of the idea and oblivious to the big picture. We come off as know-it-alls who can't resist a chance to show that we know better.

Of course, you're probably screaming that technology has to account for edge cases. It's true. But once I started noticing a pattern in my edge-case statements, I realized that it occurs in all my conversations. I saw that what feels to me like a commitment to completeness and truth leads me to bring in edge cases at inappropriate times -- for example, during brainstorming and strategic discussions. When people are in the middle of thinking up new approaches and ideas, edge cases tend to disrupt their flow of thought. And big-picture discussions are about, well, the big picture. Small exceptions are not only unimportant; they also distract from what is important.

The best times to bring up edge cases are when they add genuine value: when it's time to vet the ideas that came out of the brainstorming session, and when we get to the detailed planning. Edge cases are an essential element in identifying the complexity, costs, obstacles and benefits of ideas. And no idea, no matter how good, has been adequately addressed if we haven't accounted for edge cases in our plans.

Beyond that, we can take a better approach to the way we talk about edge cases. I've noticed that when I'm in edge-case mode, I don't preface my observations with an acknowledg-

ment that the idea itself has merit. That means the others have no way of knowing that I'm not rejecting their ideas outright or missing their points entirely.

We should also calibrate just how edgy our edge case is. It's natural for co-workers to assume that attention equates to importance. If you spend 20% of a meeting talking about a use case that represents 0.05% of system usage, they'll think you're obsessed with unimportant things. Referencing the likelihood of an edge case happening reassures them that you get the context and importance.

Although completeness and perfection are important in code, conversations are not code. They're part of human relationships. Learning to use this powerful analytical tool appropriately is essential to working effectively with nongeeks. ■



by Paul Glen

# 'Nothing but the facts' approach just won't work with business people

**I**'VE BEEN talking lately about how IT and business people have trouble communicating. It goes beyond speaking different languages. The two groups really think differently.

If you believe I'm overstating my case, then try this experiment.

The next time you give a presentation to business people, do a follow-up a day or two later. You will likely find that nearly everyone in your audience completely missed your point.

The reason we often bomb when it comes to presenting to business people is that we misunderstand how they tend to process presentations and information. We make the mistake of believing that they think like we do. They don't.

Anytime you give a presentation, you need to share four things with your audience. And you have to think about what each of those four things means to nontechnical people.

**Facts.** Most presentations by technical people are built around facts. We believe that our obligation to our organizations -- and to the concept of truth itself -- is to present the cold, hard facts as best we know them.

Unfortunately, facts don't penetrate most people in the same way that they do techies. Because facts are objective and verifiable, we find them compelling, even exciting. They stand on their

own and provide a sense of order and structure that we like.

In our minds, if you have the facts, you have all you need to make a decision. But for business people, facts are neutral at best, and not motivating in many cases. They need more than facts if they are going to arrive at your meaning.

**Insights.** Insights depend on facts, but they only come when you have illuminated the implications of the facts. An audience of business types won't arrive at these "aha!" moments if you don't point the way to the larger meaning to which the facts give rise.

And if you don't do that, you won't get through to them, because, for business people, insights are more influential than facts. You might feel uncomfortable telling your audience what they should conclude from your facts, but if you don't guide them to the insight, they may not understand what you're trying to tell them, or they may at least miss its significance.

**Stories.** As essential as insights are, they can be impotent without a story to illustrate them. Humans seem to be wired to think in narrative terms, and for nongeeks, stories are the dominant structure for understanding facts and insights, making them viscerally accessible.

Techies often complain that anecdotes don't prove anything. That's true,

but this fact doesn't change the reality that stories are compelling to most people. Don't think of narrative as a means of providing proof; think of it as a device to help people remember your important points.

**Emotions.** Most importantly, people remember what they felt during your presentation. Maya Angelou wrote, "People will forget what you said. People will forget what you did. But people will never forget how you made them feel."

Your challenge is to have an impact on your audience. To do that, you need to plan out not just what you want them to think, but also what you want them to feel -- especially in cases where you'd like them to make a decision, change course or up your funding. It's the emotional impact (which includes facts, insights and stories) that persuades business people to take action. ■



by Paul Glen

# Techies and users are in a vicious circle of mistrust

**W**E IN IT like to complain that we don't get the respect, engagement and trust that we deserve. There's no shortage of outrage in IT departments about the low regard we are accorded by our business partners. They hire us and pay us good salaries, presumably because they consider us the experts on technology and its use in business. But they frequently exclude us from strategic conversations, make decisions without consulting us and ignore our advice. In response, we feel disrespected and untrusted.

When I ask nontechnical business people how they feel about us, they use words like "condescending," "confusing," "defensive," "evasive," "legalistic" and "excessively detailed." They recall bad experiences that have led them to be skeptical of anything that IT people say. They have gone through project failures, and at times they have been treated poorly and felt bad about it. Fair or not, you are emblematic of those experiences even if you had nothing to do with them.

But though they rarely mention it, they have one more reason to mistrust us: We don't trust them much either -- and they know it. We're not known for our acting skills.

Our lack of trust arises from the many negative perceptions we have of business people. When I ask technical

people how they feel about working with "the business," they use words like "ignorant," "unrealistic," "aggressive" and "unappreciative." They say business people don't know what they want and constantly change their minds. We can recall our own bad experiences that have led us to be skeptical of anything that business people say. We have seen project sponsors shirk responsibility and shift blame, and just like the business people, we have at times been treated poorly and felt bad about it. And we make generalizations about business people based on those experiences.

It is very hard to trust someone who doesn't trust you. And since this feeling exists on both sides, a vicious cycle kicks in: They don't trust us, in part because we don't trust them, and we don't trust them, in part because they don't trust us.

It sounds kind of hopeless, but it doesn't have to be. Cycles can be broken with a little self-awareness and honesty. You just need to do two things:

Recognize your negative assumptions. If your sponsors changed requirements midway through the implementation on your last five projects, it's natural to assume that it's likely to happen again. Because you're a techie, and therefore good at solving the problems you recognize, you might incorporate something in your project process to mitigate the risk of requirements changes, such as demanding ap-

proval signatures on the requirements documents and imposing penalties for subsequent changes.

But that is likely to lead to a problem you might not recognize. Your new stakeholders don't know the history and just think you are naturally mistrustful, defensive or overly rigid and legalistic. And they feel mistrusted.

Share your feelings and concerns. Instead of making rules based on assumptions, you'd be better off telling new stakeholders that you feel concerned because of past experiences. Ask them to help alleviate your reasonable apprehensions. When you share your feelings, you demonstrate trust rather than mistrust. And in turn, they may expose their fears and concerns based on past experiences and give you a chance to help them.

What's the worst that could happen? If they don't want to participate in that discussion, then you have good reasons to not trust them and you're right back where you started. So you've got everything to gain and rather little to lose. ■





by Paul Glen

# The hazards of literal listening

**I WAS DESCRIBING** the data sources for each field in my colleague's report, when I saw her expression transform from interest to upset. "Is something the matter?" I asked. "You did ask why the report is showing the wrong information. Right?"

She replied, obviously annoyed, "You know that's not what I meant. Please don't be so literal."

After a moment of confusion, I realized that she really wanted to know what we needed to do to fix her report, not a detailed explanation of the failure mode.

At some point, everyone in IT has this sort of conversation. We geeks find this complaint, that we are annoyingly literal, both confusing and unfair. What's wrong with being literal, especially at work? In our work, precision is a virtue. Isn't it disrespectful to presume that you know what someone else is thinking? Wouldn't answering a question other than the one they asked make them think that we don't listen to them? They'd complain about that too.

The fact that this sort of problem is so common should tell us that there's something we're missing, but instead of learning from these episodes, we write off the nongeeks as illogical and difficult. We stick to our self-righteous perspective and refuse to consider what their annoyance tells us. The message we're missing is that we need to master more than one way of listen-

ing and recognize which mode is best in a particular situation.

In literal listening mode, we assume that our job is to absorb information. We pay close attention to words, ask clarifying questions to ensure that we understand questions and respond to them exactly as they are posed. We assume that the content of communication is complete, unambiguous, well understood and transparent -- that language is like code.

I think it's safe to say that we as geeks default to literal mode.

Others usually prefer to operate in a more flexible, connected listening mode. Connected listeners pay attention differently, listening to both literal and intended meaning. Connected listeners interpret information with assumptions about context as well and the subjective experience of the person they're listening to, about their emotions, ambitions and constraints. Connected listeners seek to understand what the other person intends to communicate, what's important to them rather than precisely what it was they said.

When our colleagues expect the more adaptable connected mode but instead get literal listening from us, they become frustrated. In those moments, they feel that we're like machines, incapable of relating to them as humans. Sometimes they even feel that we are being condescending, belittling them for their lack of specificity.

So how can you tell which mode you

should be in? The simplest cue is to ask yourself, "Are we discussing what technology will accomplish, or how it will work?"

Literal mode is appropriate when discussing how things work, but connected mode is essential for deciding what it will accomplish. When our business colleagues are excitedly imagining a possible future, focusing on mechanics feels like we're crushing their dreams.

The first step to becoming a more flexible listener is to analyze the nature of every conversation and recognize which mode fits best for that moment. When you've mastered this, you'll find that you get invited to be part of more conversations where important decisions are made. ■