The Freedom of "Yes" — A Personal View of Service

When it comes to IT assistance, a staff member's attitude makes a big difference in how customers perceive the service provided

By **Mark Cain**

Recently I attended an orientation session for a new staff member in my college's Instructional Technology unit. At one point, the department head asked if I had anything to say. Never shy of words, I said something like the following:

> We are, first and foremost, a service organization. The quality of assistance we provide is very important to me. If someone says we haven't given good service, I take that personally. I praise good service and don't tolerate bad. In short, I have high expectations of the service that each and every one of you provides.

All well and good — I felt I had declared my expectations up front. But how does a staff member know when he or she has performed well? While we all strive to deliver the best service we can, our notions of what constitutes "good" vary. If I have such a high standard for service, I need to clarify in my own mind exactly what I mean and find a way to communicate that to others.

Here, then, are some personal opinions about service, what it is, and why we in the IT field provide it, complete with some examples of what I consider best practice.

Why Do This, Anyway?

If I had walked the halls of NASA's manned space division in Houston during the 1960s and asked the janitor sweeping the floor what he was doing, he would have said, "Putting a man on the moon." He would have known very clearly that he was part of an effort that would culminate in Neil Armstrong stepping out on lunar soil. Now that's clarity of purpose.

It's easy to get wrapped up in the minutiae of a job and lose sight of what's truly important. We're not here to fix computers or install software or run wires or stomp on viruses. We work in higher education. We are here to educate.

We're in the people business. We don't deliver technology; we meet people's needs by providing them information and systems that are transparent, trouble-free, and easy to use. We enable people and remove barriers, and in doing so we contribute to the institutional mission.

We should make things easier for people, so that students can get educated. If we get in the way of that, either with procedures or *systems design or attitude, we're not delivering the best service.*

As a corollary to this, I offer the following:

We should make things easier on ourselves only after we've made them easier on the customer.

Does this mean that we have to abuse ourselves to do the job? No, of course not. I'm just saying that we should approach service with the customer's needs first and ours second.

"Yes" Is Liberating

When I first started as CIO of my institution, people in my division joked that "You can't say no." I used to feel guilty about this because I thought that somehow saying yes would make people think I was a doormat. I feel differently now.

This yes/no thing can be a really big deal if you aren't adequately resourced. I've seen IT organizations so resource-starved you can see their suffering, but most of us aren't (I hope) that badly off. If we've got enough resources to get the job done, then we should allow ourselves the luxury of saying yes.

Say yes or provide an alternative. A flat-out no is never appropriate.

Some of this is just a matter of style. For example: A faculty member calls in to the help desk and says, "My home computer is broken, and I want you folks to fix it." Clearly, this is out of scope of what we do, and it would be accurate to say, "We don't fix personally-owned computers." A better response, though, is "We just don't have the staff to work on personally-owned computers. However, there are a number of computer services in town that will fix yours. Some, such as [fill-in-the-blank], will even make house calls. Would you like the phone number?" The first response was a dead end. The second gave the customer a direction to go for solving the problem.

Sometimes there's no easy way to avoid the "no" word. In those situations, I encourage staff members to kick things upstairs. Realistically, if you say no to disgruntled users, with no alternative, they are likely to get angry and go to the boss anyway. The IT staff member might as well send it on its way with a positive instead of a negative spin.

Own the Problem

In the 1970s, there was a sitcom on TV called "Chico and the Man." I didn't watch it much, but one line from the show stuck in my brain: "It's not my job."

When a customer needs something, a good service provider should immediately feel a certain amount of pressure. (Don't worry; it's healthy.) The problem is now yours, and you need to solve it. It isn't that "A secretary's machine is broken," it's "One of our machines, one of MY machines, is broken." The IT service worker should feel the pressure of that problem until it is resolved.

The solution may be that the work gets delegated to someone else, but the sense of owning the problem should remain. I tell my staff that it's okay for them to check with the individual who did or will be doing the work. This isn't done out of distrust but to maintain a stake in the problem and to stay informed.

Dig Deeper

Users often ask for what they think they need, not what they actually need. They frequently propose solutions for their problems rather than describing the symptoms and letting the professional work the service need from beginning to end.

The service provider, whether a programmer, a bench technician, a telecommunications staffer, or a reference librarian, must look beyond what's being asked to see the true need.

My best example comes from the library, which is part of our IT division. An adult student comes up to a reference librarian and asks, "Where's the *Reader's Guide*?"

One response would be to point toward the wall where the dust-covered volumes still sit.

A better response would be, "We don't use the *Reader's Guide* any more. For periodical articles, use *Infotrac OneFile*. Would you like me to show you how to use it?"

The best response would be, "Well, the *Reader's Guide* is over there, but we don't really use it much anymore. What are you working on? Perhaps I can recommend some better resources." Then you listen.

Our customers don't know as much about our areas of specialization as we do. If they did, they wouldn't be talking to us. But they often know a little bit, and, as you know, a little bit of knowledge can be a dangerous thing. In this example, the student was working on a term paper, and she vaguely remembered the *Reader's Guide* from 20 years ago. It's her proposed solution, not her real information need.

The moral of the story? We shouldn't assume users know the correct solutions to their problems.

We need to use our diagnostic skills and ask open-ended (who, what, when, where, how) rather than close-ended (yes/no) questions to determine the client's real need.

Become the User

If a computer doesn't work, neither does the employee who uses that computer. If a teaching lab doesn't have a needed piece of software, the lack could have a negative impact on teaching. It doesn't matter if the employee was the one who broke the machine by doing something stupid or if the faculty member should have noticed the absence of that software before three weeks into the semester — the problem remains.

We need to put ourselves in place of the user: How would I feel if my computer was broken, and I needed to get a report out in two hours? What if I were a faculty member who needed that missing software to satisfy a key course objective? In both cases, I'd feel pretty desperate and frustrated.

Empathize with the user. That and a genuine desire to help motivate the good service provider.

Exceed Expectations

This example is a true story: An instructor called the help desk, complaining that a projecting Mac unit (Macintosh and LCD projector, both on a cart) was not working. A staff member checked it out and found that the monitor plug was loose. She tightened it, then tightened all the other cables, just in case. As she was working, she noticed that one of the plugs was applying pressure to the on/off switch on the power strip, so she rearranged the cabling to prevent a future problem. Then she noticed that the unit had a power strip instead of a surge protector, so she arranged to have it swapped out. Then she met with the student whose presentation had been disrupted because the projector didn't project. It turns out the student couldn't have displayed his work anyway because of some PC-to-Mac issues I won't go into here. She helped the student solve this problem so that he could present at the next class session. Finally, she wrote the faculty member and told him all that she had done and why.

The customer was having problems with the monitor. Fix the monitor — that's all he expected. Yet the IT staff member took ownership of the problem, solved the immediate need, and addressed several other things at the same time. The faculty member was delighted. I know, because he wrote and told me. In addition, a number of potential future problems were avoided.

Break Rules (Sometimes)

Rules and policies are very useful things. They provide structure for getting a job done and getting it done efficiently. The process of creating rules is also useful, since it forces you to think through your procedures and optimize them. But if rules are inviolate, they become barriers to service. In addition, they put needless pressure on the service provider.

I know there's a fear that if you break a rule, you've created a precedent, but I think it's okay to make exceptions. In fact, when you do, you'll likely gain a friend for life, both for yourself and for campus computing services. You can even use the act of making an exception as an opportunity to deliver an important message to the user.

When a customer needs something, a good service provider should immediately feel a certain amount of pressure.

Here's an example from personal experience: A faculty member with whom technical support had had problems over the years wanted the staff to install Internet service provider software on his personally owned laptop. Now the rule is, "We don't work on personally-owned computers." He knew this, but wouldn't take no for an answer and was applying a great deal of pressure on the help desk staff. When I got wind of it, I called him and ask what his difficulty was. There was no mystery; he was simply afraid to do it on his own, so I made arrangements for him to stop by my office with his machine. I configured it myself, using the session to explain why we as a general rule can't service personally owned computers and then to talk a little bit about his treatment of my staff. He listened to me carefully because, well, he was grateful. After all, the CIO himself had satisfied his service need; with all false modesty aside, that would have gotten anybody's attention. I don't think he's repeated the behavior — or fought our rules — since.

In this case, I gave myself the freedom to say yes. I made an exception, he got good service, and I ensured that the need for the exception would not occur again. That's a win-win. I like win-wins.

Be Friendly

A study I once read (don't ask me where) explained that customer satisfaction correlates better with the way a user feels she or he has been treated than it does with the actual effectiveness of the service received. If you are friendly, if you show you care, and if the user is convinced you tried, there's a good chance the user will feel satisfied with the transaction.

Be nice. Smiling is good, too.

Be Speedy

In 1995, the average turnaround time for resolving a problem with an office PC on my campus was two days.

This is the way the system worked: A user called into our help desk and reported a problem. The staff member taking the call recorded the problem in a database. At the end of the day, the college's single technical support specialist printed out the requests and distributed them into the mailboxes of his student technicians. A student starting his or her shift checked the mailbox, pulled out the work orders, and headed out to the office in question. Often the person who called in the request wasn't there, so the student tech returned to the office and placed the unfulfilled work order in the box of the technical support specialist, who reassigned it, with possibly the same results.

Needless to say, we didn't have a very good reputation for service in those days.

In 1996, we changed everything, moving to an immediate service delivery model. Now, we strive to resolve lost functionality within one hour for 50 percent of the problems, within four hours for another 40 percent, and within 24 hours for the last 10 percent. We routinely exceed our targets.

People like us a lot better now. Why? A couple of reasons, I think. One, lost functionality for an hour or a day or two days means lost productivity. They can't do their jobs if we haven't done ours.

But there's another less rational, but perhaps more powerful, reason: We no longer make them wait for service.

We live in an impatient society. We get our money from ATMs, our food at the drive-through. We order garments over the Internet and expect them to be delivered within a couple of days. A three-minute wait at a traffic light seems an eternity. None of us wants to wait, and our customers aren't any different.

A proverb: He gives twice who gives quickly.

Communicate

When things go wrong, there's a natural tendency to hide. "Joe Smith's computer is broken. I don't know what's wrong. Boy, I sure hope I don't run into him before I can figure it out." In a case like this, rather than hide, it's better to call Joe and say, "I've tried this and this. So far, no success, but I've got a call into the manufacturer's help line, and as soon as I hear something, I'll let you know."

Here's another example from personal experience: In the fall of 2000 my staff and I were involved in a major technology rollout. Unfortunately, the IT division had spent months building up faculty expectations about it, then couldn't deliver. Only a few faculty members were actually coming to me complaining, but I knew they represented the tip of a very large iceberg. I didn't have any solutions, but I did have information, so I started communicating with them regularly. I told them we knew about the issue and were concerned, explained the problem in detail, and reported on what we had tried and intended to try. I kept them informed every step of the way, even though it took an incredible amount of effort to do it. Those communications were crucial; they bought my division the time it needed to solve the problems.

In good times and in bad, communicate, communicate, then communicate again.

One final thought about communication: Sometimes nothing will do but a direct, face-to-face encounter. If you find yourself hiding behind voice mail, if you realize you've gotten into a never-ending series of e-mail exchanges with a client, do the unexpected. Go find the person and have a non-mediated encounter. A 15 minute face-to-face talk will often clear the air and solve problems much faster than a series of e-mails. It's good exercise, too.

The thornier the problem, the greater the need for in-person contact. This is especially true if one of the parties is getting angry.

Final Thoughts

In all of this, I don't mean to minimize the enormous challenges that face the IT operation and its staff. Sometimes I think we are like Sisyphus, rolling that boulder up a steep incline, only to have it come crashing back down on top of us. The work is hard, and the resources we have to work with are never enough.

Ultimately, though, good service isn't about adequate resources or technical proficiency or any of that stuff. It's about caring — and about treating others as we would like to be treated ourselves.

Where have I heard that before? *e*

Mark Cain (mark_cain@mail.msj.edu) is Executive Director of Information Services and Support at the College of Mount St. Joseph in Cincinnati, Ohio.