acmqueue

Gardening Tips

A good library is like a garden.

Dear KV,

I've been maintaining a set of libraries for my company for the past year. The libraries are used to interface to some special hardware that we sell, and all of the code we sell to our end users runs on top of the libraries, which talk, pretty much directly, to our hardware. The one problem I keep having is that the application programmers continually reach around the library to talk directly to the hardware, and this causes bugs in our systems because the library code maintains state about the hardware. If I make the library stateless, then every library call will have to talk to the hardware, which will slow down the library and all of the code that uses it. What do you think is the right way to get people to actually ask for the features they need instead of reaching around the library?

Tired of the Reach Around

Dear Tired,

I find that the best way to get people to use my libraries correctly is to embarrass them in group meetings by asking questions such as, "What part of 'clean API' design don't you understand?" I usually ask this question in a loud voice, and try my best to get the veins in my neck and head to start pulsating wildly. I find that having a generally insane demeanor has the desired effect on my coworkers, as well as bank tellers, people in long checkout lines, and those people who feel they need to stand right next to me on a crowded subway. Another option is to watch every single check-in and to change every program to go through your library, which might be tedious, but don't rule it out. You would not be the first programmer to take that route. In all honesty, unless you enjoy being threatening all day, and I'm told that this is not the normal state for most people, you'll probably have to find out what these so-called application programmers want and try to address their needs.

A good library is like a garden: "As long as the roots are not severed, all is well. And all will be well in the garden. ...In the garden, growth has its seasons. First comes spring and summer, but then we have fall and winter. And then we get spring and summer again." Alas, Chance the Gardener (*Being There*, 179) didn't really have much to teach anyone about programming or economics, but that's another film altogether. The fact is that we can learn from one particular type of gardener about how to write an effective library.

Consider the following story. A gardener was tasked with placing paving stones so that people could walk through and enjoy the plants and flowers, as well as get to places that were on opposite sides of the garden. A week later the owner of the estate went out into the garden to see how the work was progressing. To his surprise there was not a single stone yet laid anywhere; the garden was pristine but had no paths. Thinking that his gardener must be busy tending to the plants and trees, he decided to ignore what he saw and go back to his office to work. A second week went by and the owner went out into the garden again to see what progress had been made. He was no longer surprised, but more shocked, that the gardener seemed to be continuing to ignore his request for

workable garden paths. After a third week went by without any progress, the owner became quite angry and stormed off to find the gardener.

The owner demanded of the gardener, "It has been three weeks! Why haven't you laid a single paving stone? The grass will get torn up by people crossing it any way they please!"

The gardener looked at his boss with apparent shock and said, "But, Sir, how can I know where to put the paths until the people have shown me where they would walk?" He then asked the owner to follow him down to the garden. When they arrived he showed the owner how, rather than walking any which way on the grass, paths had already been worn where the largest number of people needed to cross to their destinations. He turned to the owner and said, "Now that I can see the paths, I can lay the stones."

If you want people to use your library, then you need to find out why they're reaching around it and try to address their needs. The library isn't there for you; it is there for them.

KV

Dear KV,

Have you ever noticed that when you submit a ticket to a ticketing system no one ever reads it? They just e-mail, call, or drop by your desk to ask you about the ticket, but it quickly becomes painfully obvious that they haven't read what you've written, except for, maybe, the summary? How do you deal with such people?

Ticked off at Tickets

Dear Ticked,

How do I deal with those people? Well, I have this special chair near my desk, and when they sit in it, it runs about 1,000 volts AC through their bodies. I got the idea from an old James Bond film. The problem is, I can't get our janitorial staff to take out the bodies, so I have to do it myself, and the stench is amazing!

OK, no, I don't really electrocute people at work. I do find myself amazed, however, when people who are clearly capable of reading simply don't take the time. The reason you have a ticketing system is to streamline work, and their abject stupidity and neediness pretty much ruins that streamlining. They also seem to interrupt at exactly the wrong moment, sort of like those waiters who always ask you how your meal is when your mouth is full.

I admit that I take a fairly sarcastic, or perhaps a more sarcastic, tone when dealing with such people. I was recently called by a tech-support line that was supposedly fixing a computer for me, and when it became obvious that the caller had not read any of the things that the previous technician had written down, I began to speak—very slowly and very quietly—and told...them...in...very...short...sentences...what...had...passed...before. I then finished up the call by asking, in what can only be called an icy voice, "Now did you write that down?"

My favorite example of this phenomenon involved a tech-support person I worked with who was dealing with a particularly difficult customer. The product we worked on was an operating system that systems integrators would extend in building their products. Customer accounts were assigned to individual customer support reps, and this rep had a doozy of an account. The customer in question would never read the manual but would call this tech-support engineer, sometimes more than once a day, and ask simple questions that were clearly answered in the manual. One day the

tech-support engineer had his door open and we all heard him say, “Do you have volume X of the manual? Yes? Good. Please turn to page Y. Now read along with me...” and he made the customer read the manual page along with him. I’m not sure if this cured the customer, but it made the rest of us laugh pretty heartily.

I think the best thing you can do when people ignore your tickets is to ask, very politely, “Did you read the ticket?” If it continues to happen, then you can go down the politeness scale to, “Did you read what I wrote?” and from there to, “Can you read?” I would save that last one, though, because from there, there’s no going back.

KV

KODE VICIOUS, known to mere mortals as George V. Neville-Neil, works on networking and operating system code for fun and profit. He also teaches courses on various subjects related to programming. His areas of interest are code spelunking, operating systems, and rewriting your bad code (OK, maybe not that last one). He earned his bachelor’s degree in computer science at Northeastern University in Boston, Massachusetts, and is a member of ACM, the Usenix Association, and IEEE. He is an avid bicyclist and traveler who currently lives in New York City.

© 2010 ACM 1542-7730/10/1000 \$10.00