One Bagel With My Requirements, Please!

I'm pretty much a creature of habit. Every morning, on the way to work, I stop at my local grocery store and get a large coffee with lots of room for cream. I add two French Vanilla creamers, and then add one artificial sweetener (the yellow – *never* the blue or the pink). Finally, I top it off with some skim milk (OK – I use half-and-half, but the skim milk sounded healthier).

Except on weekends. On Saturdays, I stop at ... well, let's call it Albert's Bagels. For years I would walk in and inevitably order a whole-wheat bagel with low-fat cream cheese, smoked salmon, and extra capers. And – of course – a large coffee (see above). There were multiple servers, each one handling one customer at a time. When I got to the front of the line, the pleasant person on the other side of the counter would take my order and make it to my specifications. If the customer at the front of the line had never in his/her life been to a bagel store (and, therefore, took forever to comprehend the intricacies of *plain* or *whole wheat*), another server would be available to handle my order.

That was until a few years ago. I assume that, in an effort to better control inventory and automate things, they went electronic. Now, when you walk into Albert's, you stand in line number one to order from a single person using a hand-held terminal. They actually seem to have two ordering machines, but they never have two working at the same time. There are generally long lines stretching to the door. When you get to the front, the person enters your order into the computer and never seems to know where the extra capers button is. And – occasionally – he or she runs off to help somebody else with production, leaving nobody to take orders until they return. When my order is finally taken, I then go to line number two to pay. The people making the sandwiches now face the other way with their backs to the customer. No more can I say, "Wait – add some more capers." To add insult to injury, people who don't need bagels, but are only ordering a drink have cut ahead of me into line number two. And, inevitably, the person who cut in front of me orders two double mocha cappuccino frappes with extra sprinkles of cinnamon and light nutmeg. The cashier will stop ringing up customers to make the drink – further frustrating me. Of course there are two cashiers' stations – but one cashier is always making a drink or handling the people who cut into the second line *just for a quick bagel*. I've complained about having customers cut – and get dirty looks from the person who cut and blank looks from the staff.

I guess this makes me a curmudgeon[1]. It irritates me that a process that used to work well (and suited the customer) has been updated to *not* work. The person who makes my order isn't the one who took the order, so my order is frequently wrong. Standing in two lines is frustrating. Having people cut into the second line is equally frustrating.

But I understand. The price of ingredients are rising, and Albert's had to economize. Less staff, more work. Electronic ordering makes inventory much easier. Selling *foo-foo* fancy drinks keeps them competitive, and certainly these drinks have a higher profit margin. Still though, it seems something is wrong when loyal customers who like things the way they used to be feel slighted. And since I have lots of time to talk to others while waiting in line, it appears that I'm not the only disgruntled customer.

Isn't it interesting: losing loyal customers to improve the process of the producer? It would appear that making sure the customer is first (or, at the least making sure that the customer *thinks* he or she is first) would be a primary goal. There must have been ways to upgrade service in a manner that would not torque me off every time I walk in.

Updating obsolete software is a fact of life: What is not updated or rewritten to meet new requirements and changing environments becomes obsolete very quickly. New software *will not* have the exact same form, fit, and function. It *can't*. The new software has to accomplish more, in different ways, with new functionality. Like Albert's Bagels, change has to happen to allow them to stay economically competitive. If you're a curmudgeon like me, and you expect release 3.0 of your software to have the same look and feel of version 2.X – well, it's not feasible. If it did look and feel the same, it would be equally obsolete as soon as it was released.

New software must meet the needs of the customer *and* still provide service in a customer-acceptable manner. If you remember to keep customers in the loop as you juggle new environments, limited resources, and changing requirements, then you are doing the best you can! Customers need to understand there will be changes, but as long as the overall mission can still be accomplished, they need to be willing to accept some changes. You know, my bagels still taste just as good – and that extra five minutes really is not that big of a deal. It beats having Albert's go out of business.

Are you keeping your customers in mind as you upgrade and improve? Are the customers involved in changing and building the new system? Will it be acceptable to them? If not, your customers will feel the way I do when I get unordered bean sprouts.

And happy 20th birthday, CROSSTALK.

—David A. Cook, Ph.D.
Principal Member of the Technical Staff and Curmudgeon
The AEgis Technologies Group, Inc.
dcook@aegistg.com

P.S. One very nice thing about Albert's Bagels: They let you get free refills all morning long and pleasantly let you sit with your laptop and write your BACKTALK column without rushing you!

Note

[1] Curmudgeon: A bad-tempered, difficult, cantankerous person.

Can You BACKTALK?

Here is your chance to make your point, even if it is a bit tongue-in-cheek, without your boss censoring your writing. In addition to accepting articles that relate to software engineering for publication in CROSSTALK, we also accept articles for the BACKTALK column. BACKTALK articles should provide a concise, clever, humorous, and insightful perspective on the software engineering profession or industry or a portion of it. Your BACKTALK article should be entertaining and clever or original in concept, design, or delivery. The length should not exceed 750 words.

For a complete author's packet detailing how to submit your BACKTALK article, visit our Web site at <www.stsc.hill.af.mil>.