Culture Clash

How a move to consumer products turned my definition of quality upside down

by Kathy Iberle

"THESE PEOPLE JUST DON'T CARE ABOUT quality!" I can't tell you how many times I said that in the year I moved from medical products to consumer printers.

Working on cardiographs and defibrillators was like living in a software-engineering textbook. We used most of the classic practices: detailed specifications, frequent inspections and reviews, exhaustive requirements-based testing. We set release criteria, and the product simply did not ship until those criteria were met. Schedule slips weren't trivial, but correctness and reliability won out over schedule every time.

After some years, I transferred into the inkjet printer business. The practices in my new workplace were very different—the specifications were less detailed, release criteria were much less formal, and hitting the release date was all-important. There was much less test documentation—in fact, sometimes I would see testers testing with no written test procedures at all. How could people be so sloppy? Why didn't they do things "by the book"? Didn't they know any better? Or did they just not care?

As I got to know my new coworkers, I realized that they were neither incompetent nor uncaring. In fact they cared a lot about what they were doing, and they were delivering pretty good software. I had to look elsewhere for explanations.

What *Do* They Care About?

As my first project in the new organization neared release, the team's worries and fears became more apparent. I realized that problems I considered disastrous didn't necessarily worry this team, and issues that I was accustomed to treating lightly were considered deadly serious. (It's a good thing that I wasn't managing the project at this point.)

INFO TO GO

- Best practices may rightfully differ among industry segments.
- A quick test of what quality means in your business is to ask people, "What keeps you up at night?"

I had been accustomed to concentrating on safety, reliability, and correctness. In medical products, misdiagnosis was considered as serious as an actual injury, and failure to operate on demand was literally a life-or-death matter. We hadn't invested heavily in usability testing because our users were a fairly predictable bunch. (They actually read the manuals.) Since our software products were sold preloaded with a predetermined operating system, compatibility with different operating systems wasn't even considered.

In contrast, the printer team ranked compatibility problems very high on their fix list. Usability issues also got a lot of attention—consumer equipment has to stand on its own with a wide variety of users. Imagine the chaos if thousands of people called the help lines at once on the day after Christmas, unable to install their shiny new printers.

The printer team seemed obsessed with hitting release dates. In medical products, deadlines often slipped a few weeks while we fixed the last few problems. These slips could be made up by putting on extra shifts in the manufacturing plant or by putting the product on back-order for a bit. I didn't really grasp high-volume manufacturing until my new team put together a contingency plan for missing the CD release date. The plan involved using a courier to hand-carry thousands of CDs from the CD plant to the assembly plant, then putting in massive amounts of overtime to hand-pack the CDs into the printer boxes. Delaying the final printer shipment was not an option, since it would have meant missing a crucial holiday sales window. Suddenly, the obsession with release dates made perfect sense.

What Is Quality?

It eventually occurred to me that the two organizations might be defining quality differently. In his book *Quality Software Management: Systems Thinking,* Jerry Weinberg defines quality as "value to some person." The printers provided value to the customers by printing nice documents easily from any application the customer used, on any computer the customer owned—in other words, by being correct, user-friendly, and highly compatible. The defibrillators provided value by being immediately available and delivering the correct treatment—in other words, by being reliable and correct.

But doesn't everyone want a reliable product? Sure, but what the user considers "reliable" is partially dependent on the price of a failure. An emergency medical technician with a dying patient wants that defibrillator immediately available, whereas reprinting a document is rarely a life-and-death matter.

The business also derives value from its products. Profit comes from satisfying the customer but is also dependent on other factors. Meeting deadlines had more effect on profit in the printer world than it did in the medical market. Extensive documentation provided value in the medical business by satisfying regulatory audits.

Ideally, businesses would spell out which aspects of quality are most valued by their customers. Unfortunately, you often get an emphatic, but vague, "quality is important" instead. The distinctions come out when hard decisions have to be made—justifications for major changes in plan often explicitly state the need for a particular aspect of quality. If you'd rather not wait for trouble to arrive, asking people "What keeps you up at night?" can be equally enlightening and less traumatic.

What About Best Practices?

Once I realized that printers and medical products had different quality criteria, I started to recognize that some of the printer development practices that appeared sloppy or strange to me were actually sophisticated methods of addressing problems that didn't even occur in medical products. For example, those

testers who were working without written test procedures were doing exploratory testing—in skilled hands, this is an effective method of finding compatibility problems without testing every possible combination.

Conversely, some medical products' practices addressed problems that were less prevalent or less serious in the printer business, so in the printer business those practices could be scaled back. Medical product developers delivered exactly what the doctor ordered by writing detailed specifications to be reviewed by medical experts. The printer users didn't have a specific user interface in mind—they wanted a printer that was easy to use, and it was up to us to define it.

In short, the "best practices" for medical products and for printers were different. Why didn't the software-engineering textbooks explain this?

Why Didn't I Learn This in School?

As I searched for software-engineering advice for consumer products, I found a few good books and heard some helpful presentations at conferences, but often it seemed like open warfare between the "shrinkwrap software" people and everybody else. Then, in conversation, Cem Kaner pointed out that the software industry falls into several large categories, each of which has characteristic practices. Around the same time, James Bach wrote in his article "Good Practice Hunting" (*Cutter IT Journal,* February 1999) that no practice is best in every context. He, too, broke up the industry into groups of practitioners who use similar practices.

The concept of categories or *practice cultures* within the software industry explained a lot of things, including why the textbooks never mentioned that I would need different practices in different businesses.

The people who wrote the majority of these textbooks originated in a single practice culture—the one that writes custom software systems on contract, such as large financial systems or military applications. This is the oldest practice culture, and its members publish prolifically. When working on a fixed-bid contract, people worry about running over budget or failing to deliver what the customer ordered, in addition to the usual issues with reliability and correctness. Naturally, the textbooks recommend practices that prevent or reduce these sorts of problems. The Capability Maturity Model in particular is laden with excellent practices to reduce the likelihood of running over budget or delivering unexpected variations to the agreed-upon features.

The medical products team found most of the "classic" practices to be appropriate because we had to deliver correct results in the formats dictated by cardiologists. There were some classic practices we did not strictly adhere to. For instance, we could make up small cost overruns over time, so fewer stringent budget control practices were needed.

The printers operated in a very different business environment, and belonged to a different practice culture—the mass-market software practice culture. When I moved between cultures, I had to learn some new practices and drop some old ones. This happened again when I moved into Internet applications.

In that move, I made fewer assumptions and asked more questions. I listened for the values inside worries and the reasons behind practices. In the end, I learned that most people really do care about quality—the quality that matters to their customers, not the quality that fits my preconceptions. STQE

Kathy Iberle (www.kiberle.com) is a senior software engineer at Hewlett-Packard. She's been in the software business for eighteen years in a wide variety of fields.

***STQE* magazine is produced by Software Quality Engineering.**