Robert L. Glass

Y2K and Believing in Software Practice

Naysayers made an event-turned-nonevent more unpleasant than it needed to be.

"Countdown to doomsday"—One book's pre-Y2K description/prediction of what was happening in the computing world.

The smoke (and the mirrors?) of the Y2K computer problem has pretty well blown away now, and for the most part that IT bug-a-boo has been a nonevent. Of course, there were some problems:

- Some ATMs with a four-day, look-ahead feature refused service when the four-day period spanned the turn of the century.
- A videostore that charged for a video tape assumed to have been checked out in 1900 (the late fee: $91,250!).
- The stores that resorted to pen and paper when their computer systems failed (or was it quill pen and parchment?).
- The central heating system that malfunctioned in a Korean apartment complex.
- A Korean newborn whose birth date was recorded as 1900.
- An X-ray machine that failed in Norway.

LAURA CORNELL

But considering the breadth and scope of computer usage worldwide, these early return failures are the proverbial pimple on a pumpkin. Things could conceivably change as the millennium marches on, but for now it looks like IT is relatively Y2K home free.

I'd like to use this semi-joyous occasion to address two categories of naysayers who made the Y2K event more unpleasant than it needed to be:

1. **Those pundits who spoke out but *didn't* know what they were talking about**. An awful lot of highly visible professional people spoke out on the subject of the Y2K computer problem, although their technical knowledge of the field was basically subzero. Take the economist Edward Yardeni as perhaps the most visible example. (No, Yardeni is not the same person as IT's very own Ed Yourdon!) Yardeni, who has no apparent background in developing or maintaining software, was one of the most frequently quoted doomsday scenario forecasters. Why should anyone have believed

A resounding raspberry to those who were so quick to foresee their skewed visions of a human disaster in the making.

him? After all, the nature of the Y2K computer problem was that (a) it was very simple, but (b) it was ubiquitous, and (especially) (c) it was highly technical (it was mired in the deep innards of *N*-thousands of programs in a place where only practicing software maintainers were going to be able to find it and fix it). Yardeni—and the many other technology-ignorant pundits—had a thoroughly insufficient basis for making any prediction, good or bad.

The know-nothing naysayers who really bug me, though, are those who say, now that the nonevent has occurred, that the money spent on remediation was wasted. Wasted, indeed! Those people have boxed IT into a Catch-22, damned-if-you-do and damned-if-you-don't, predicament. Based on my understanding of the problem itself, and the enterprise approaches to the problem, I would suspect that 99% of the money spent on fixing Y2K was well spent. In fact, this would be an area where it would be really difficult to waste money—certainly money spent on fixes was not money wasted (unless—for some odd reason—the fixed system will never be used again). I cannot imagine that many enterprises spent Y2K money on things other than fixing the problem (perhaps contingency plans, given the nature of the nonevent, could be called money wasted, but only the most callous optimist could in good conscience take this position).

2. **Those pundits who spoke out but *did* know what they were talking about.** Some of our field's best friends were right in there with the know-nothings predicting disasters. Sure, they couched their positions in sophisticated statistics (such as "there's only a 10% chance that a catastrophe will occur"), but nevertheless they rushed to print with their most negative thoughts.

Now I think there's an odd dynamic at work here, one that those "best friends" may not be aware of. It's a commonly acknowledged fact that some fields are burnout-prone. For example, people in the crime/corrections field tend to see everyone around them as a potential criminal; people in social work tend to see everyone as, at best, troubled. I know someone whose father was a forest ranger, and because most of his work was spent on the problems caused by a malicious few, he tended to see most people as basic troublemakers.

How do these examples relate to the IT gurus who saw Y2K as a disaster-in-the-making? The problem with being a guru is that you are called in to help only when your client has a big problem. Because of this, you tend to see the IT field as populated with problematic projects. After a few years of that, you begin to lose your belief in IT practitioners. And who would be most likely to see the Y2K computer problem as an immense disaster of awesome proportions? Those who believe IT professionals are so inept as to not be able to fix the problem.

Well, think again, know-something naysayers! The name of this column—and the name, I would assert, of the computing game circa 2000—is "Practical Programmer." This name was chosen because, at least in my view, there are many positive things to say about the state of software practice, and there was a need to provide a platform from which those things could be said. And I believe, given the Y2K nonevent, it's time to shout from the rooftops about the practical programmers who made the computer problems of Y2K fade into obscurity.

Congratulations to all of you practical programmers. And congratulations to everyone, know-nothing or know-something, who saw a serious need to step in and fix this very real problem. But a resounding raspberry to those who were so quick to foresee—and shout "fire" in a crowded theater—their skewed visions of a human disaster-in-the-making. C

ROBERT GLASS (rglass@acm.org) is the publisher of the *Software Practitioner* newsletter and editor of Elsevier's *Journal of Systems and Software*.