May, 1997

Editor's View

Slow Down!

The rate of change in the software industry has reached the point where developers can't keep up.

By Tamar E. Granor, Editor

A reader recently asked me for advice on what to learn next. He wanted to learn something that would still be around 10 years from now. It got me thinking about the pace of change in our industry.

Ten years ago, I was a full-time mother, taking some time off between graduate school and work. Most of my work up to that point had been done on mainframes, most of it in Pascal. I owned a personal computer—a DEC Rainbow. It had no hard drive and used 5 1/4" floppy disks. I think it had 64KB of RAM.

Out in the world, PCs were beginning to make a dent. The University of Pennsylvania, from which I'd graduated and where I'd taught programming to hundreds of undergraduates, was starting to set up PC labs to replace the time-shared terminals students had been using. Microsoft Windows 2.0 was released in 1987, as was Excel for Windows (though Lotus 1-2-3 was the hot spreadsheet product). Word 4 for DOS and Word 3 for the Mac were released then, too. A little product called FoxBase+ had been around for less than a year.

What a difference a decade makes. I'm on my third desktop PC (in addition to a pair of notebooks). My latest machine has a 3.2 GB hard drive and 64 MB of RAM. The most important external drive is the CD-ROM, from which I do most of my software installation.

As for software, things have changed there, too. There hasn't been a new version of Word for DOS for quite a while and the current version for Windows is 8. Excel, also at version 8, is the leading spreadsheet and there are two current versions of Windows, 95 and NT 4.0. Finally, FoxBase+ has gradually transformed into Visual FoxPro 5.0.

This kind of change is not bad in and of itself. Each new version of the various products has added power and flexibility. Over the years, I've welcomed various updates to the products I use as the new features make my life easier. I can remember waiting impatiently for the next version of a product to ship.

But, in the last couple of years, the pace of change seems to have accelerated dramatically. There was almost two years between FoxPro 1 and FoxPro 2 and another year and a half between FoxPro 2 and FoxPro 2.5. It took almost two and a half years to go from the initial release of FoxPro 2.5 to Visual FoxPro 3 (with some incremental improvements along the way). But the time between Visual FoxPro 3 and Visual FoxPro 5 was just a little more than a year.

Other products have been affected by this acceleration, too. Internet-related products seem to have shortest revision cycles.

So what? Why isn't this good for us?

The big problem is time. We need time to learn the new products while still being productive with the old ones. When there's a quantum leap in the technology such as the addition of OOP in VFP or all the new Internet material, it takes a long time to get up to speed. As a developer, I expect to spend a fair amount of time on professional development, but we're reaching the point where we have to choose between earning a living and keeping up to date.

Short revision cycles also mean that beta tests are much shorter, as is the time between beta cycles. (One of my friends recently referred to one product as being in "perpetual beta.") Beta testing is another thing that eats into productive work. Most of us can only allocate a small amount of time for it. Back in the old days, when there might be months between beta cycles, we could organize our work and our lives to give us more time for testing when it came around. With one beta on top of another, we simply can't exercise the product to the same depth as before. So, the quality of beta testing suffers.

At the same time, the developers writing the product have less time to debug it and the internal testers have less time to put it through its paces. The result is software that reaches the market with some pretty nasty bugs.

The brevity of the revision cycle has cut down on the number of interim (bug fix) releases of products, too. The final, fairly stable version of Word 6 was 6.0c. But Word 7 (Word 95) never made it as far as 7.0c - the last version was 7.0a.

Even when manufacturers produce interim releases, sometimes they cause more problems than they solve. Service Pack 2 for Windows NT 4 was rushed to market because it fixed some bugs, but it contained other bugs so serious that Microsoft has made "hot fixes" available on their web site rather than waiting to issue a Service Pack 3.

In addition, the pace of change cuts down on the number of quality third party books available. Writers don't have enough time to really learn about a product before writing about it. Plus, the length of time in which the book is current is so short that it's hard for publishers to make money. Expect to see more third party books by professional writers and fewer by the people who use the products day in and day out.

What's the solution to all this? Easy, it's the same thing your mother told you when you made mistakes because you were in a rush: *Slow down!* For most of us, software is not a life or death venture. We'll survive if the next version of our favorite application doesn't ship for a little longer. Unfortunately, we're not the ones who need to take this advice - it's the software companies who must do so. We need to tell them loud and clear that we mean it.

Finally, to the reader who wants to know what to learn that will still be around in 10 years, the answer is good programming skills. No product that you learn today is going to be the same in 10 years, but the practices you learn will serve you no matter what language you're working in.