

Twenty-five Years of the Wrong Choices!

Just a few weeks ago, I was notified that my article submission for the 25th Anniversary Edition of **CrossTalk** was accepted—so somewhere in this issue, there is my scholarly article about how we have progressed over the last 25 years. However, this BackTalk column is a fitting end to the issue. It is more about where we have digressed since 1988. After all, in retrospect, it is easy to see where we made the right choices. It is where we made the wrong choices that few people wish to elaborate on.

In 1988, I was the proud owner of a really high-quality video tape recorder, chocked full of awesome features, complete with stereo recording. It had a digital channel selector, and could program up to 12 (that is right – TWELVE!) future recordings. A high-quality electronics manufacturer, Sony, made it. And it was a Betamax. Arguably, a better product than VHS—it had stereo and higher quality video—but it was a losing battle.

Of course, I had a “backup” format for the movies that I found really important—ones that I paid money to buy, so that I could have a high-quality movie that I could watch over and over, for years and years to come. Yes, I owned a LaserDisc. I had LaserDiscs of “The Wall” and “Rocky Horror Picture Show.”

Back in 1991, I graduated from Texas A&M with my Ph.D. I took not one but two courses on parallel algorithms and parallel sorting. It was not a question of, “if we would be converted to parallel processing by 2010.” It was more of a question of, “what kind of parallel architecture would we all be using?” Choices included the mesh, the cube, and the butterfly, just to name a few. Granted, we now use multi-core, multi-threaded machines, but few programmers really know how to write code to truly take advantage of parallelism. Instead of large-scale parallelism, we now do parallelism “in the small”—nothing at all like what we envisioned back in the early 1990s.

Also in the early 1990s we thought that by 2000, there would really be only one programming language used in the DoD, right? Heck, I was a member of the Ada Government Advisory Group (a.k.a. the Ada GAG—a horrible acronym if there ever was one).

In the mid 1990s, I was convinced that the 3.5” floppy disk was eventually going to disappear – the thin floppy was incapable of holding enough information – so I made sure to back up everything I had on the one medium that we just knew would be around for years and years to come—the Iomega Zip Drive.

By the year 2000 came along we decided that the “single programming language” idea was never going to work, so we decided to agree on a common operating system instead. I was on the working committee for the Defense Information Infrastructure Common Operating Environment (DIICOE). Bet you have not heard of DIICOE in a while either, have you?

Even though I am the epitome of a die-hard Mac user, for about 15 years, I used another OS. What did I switch to? Linux, of course. In the 1990s, we just knew that by the early 2000s, Linux would be the predominant operating system for both home and office.

Speaking of the Apple Macintosh, who would have predicted that both Macs and PCs would share the same chips? Over the years, I learned and then taught 6800/68000 assembly language, and also mastered the Power PC (PPC) architecture. Now, my Mac runs on an Intel, and using a virtual machine interface, it boots either OS X, Windows 7 or Windows 8.

I only represent one lowly software engineer—and the list of projects, technologies and initiatives I have been on that are obsolete and no longer part of the DoD is really long. One could argue spectacularly long. So, this means I have been a failure, right?

Well ... no, to put it bluntly. In fact, almost everything I have listed above actually contrib-

uted to progress in engineering and computing science. Ada is still used, and some of the features it heralded became part of other, newer languages. Parallel processing is still a critical component of supercomputing. In fact, it appears that Moore’s Law might apply to the number of processors in a system. DIICOE helped us standardize some critical components of embedded operating systems, and helped standardize some real-time operating system. The Un*x OS is not extremely popular for home computing, yet it runs a lot of servers, supercomputers, and large-scale systems. It is also the basis for the Mac OSX operating system.

What about the 68000 and PPC architectures? They are used in high-speed embedded systems. The LaserDisc? The DVD simply eclipsed it—higher capacity, smaller size, cheaper technology, and better quality video. Same with the Zip drive. It was great for its brief time, but the non-moving technology (and eventual greater capacity) of the USB drive sounded its death knell. These were not failures, just technologies that were eclipsed by better technology. There is no shame in having worked on a once cutting-edge technology that becomes obsolete.

That is just the way progress is. Two steps forward, one step back. Every great new technology we have today is based on something that preceded it. You cannot judge progress by the number of technologies that have failed and been replaced. You can only say “What we have now is better than what we had yesterday.”

Learn, improve, discard, and move on. I would bet that every decent developer or software engineer could point (usually with pride) to some project they worked on that has been made obsolete by the steamroller of progress. And every one of us has learned from the experience.

**Progress marches on.
Just like CrossTalk.
Happy 25th Anniversary!**

David A. Cook
Stephen F. Austin State University
cookda@sfasu.edu