Mainframe Hall of Fame

A Conversation with Dave Cole of Cole Software

By Robert Shimizu

Q: When and how did you get into computers?

Dave: I started in college as a freshman at the University of Pennsylvania in 1965. I took a course on a "small" early computer. In physical size, it filled a room, but in "cyber-size" it was microscopic, with something like 1000 memory locations.

I liked the class so much that I went looking for another course. But they didn't have anything. So I spent the spring semester writing a Fortran program just to learn the language. I didn't even have a computer to run it on.

In the fall of '66 I picked up a *Time* magazine that described computers as the wave of the future, so I went to the Student Employment Office and asked if they had anything at the Computer Center. They did, but the job had nothing to do with computers. It didn't matter. After that, one way or another I wormed my way in.

Q: Do you have a university degree, or did you work your way up?

Dave: I've got a worthless degree—a BA in math, which I got out of unintentional intimidation rather than merit. My math skills are well below algebra these days. I can do arithmetic, but that's about it.

Q: What is your outstanding achievement in computing?

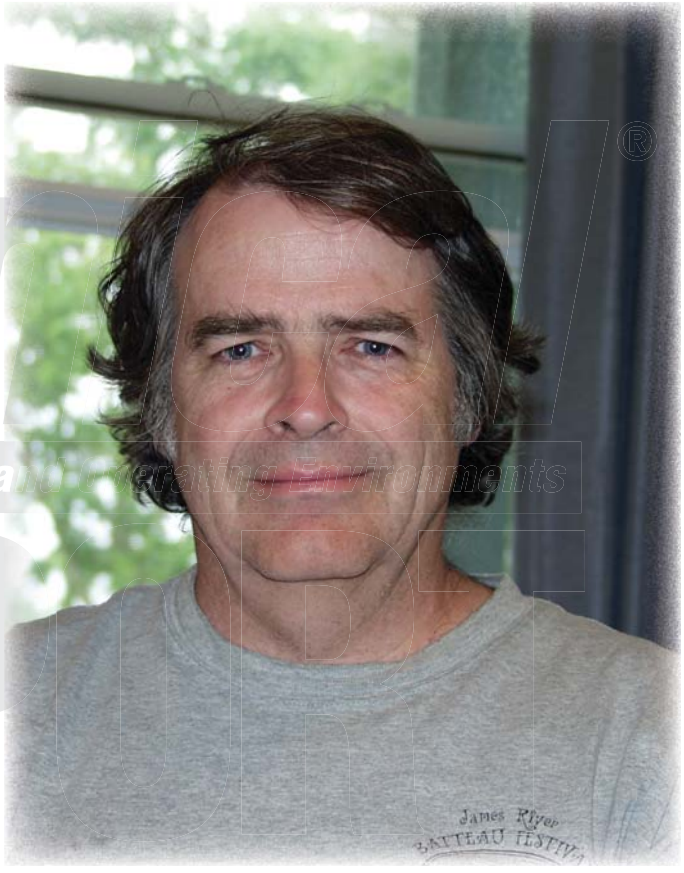
Dave: Well, the obvious answer there is z/XDC (a tool for debugging mainframe Assembler programs). I've written a lot of programs that I've been pleased with and proud of, but not in a long time. For the last 15-20 years I've devoted my efforts entirely to z/XDC.

Q: Where do you see the industry going in the next five or ten years?

Dave: Mainframes are becoming a smaller percentage of the overall market. Nevertheless, I believe they are still growing in absolute numbers. And as servers, they can't be beat! That's why Cole Software's market continues to grow.

Computers, however, are going to take over the world. With the convergence of packet networking with GPS technology and satellite communication, governments have the ability to know and control just about everything. As that capability grows, you get things like ONSTAR, which people put in their cars as a convenience or safety measure. Then it becomes required.

Nowadays, the onboard computers in cars are required to be Internet-aware, are required to have GPS, and have the potential to have two-way communication whether you have an ONSTAR installed or not.



Where are computers going in the future? Computers are going to be used to control the population. You get these things because they're helpful, they're convenient and useful. The other side is that they can be oppressive.

Q: Who do you consider your mentor(s) in computing?

Dave: Probably the programmer I admire most is Howard Gilbert, who was a co-worker of mine at Yale University back in the '70s. He's still there.

I was at Yale for 9-1/2 years. There was Howard Gilbert, Carl Williams, Joshua Auerbach, Phil Long, and a few other people, all brought together by a man named Grey Freeman. It was an extraordinarily talented pool of programmers in an environment in which you could

thrive in programming. There was a tremendous amount of knowledge in the room, and what you didn't know you got help with. You could say, "I can't get this to work," and someone might ask, "Did you think about thus-and-such a solution?"

It was a very creative group. I've been running Cole Software since 1988, and I miss that environment. Here, the programming is pretty much a one-man show. I don't get the cross-pollination that you get when you have a room full of talented experts coming together.

I think that my time at Yale was the most influential and constructive period of my development.

Q: What is your normal work week? How many hours do you spend, and how do you spend them?

Dave: Nowhere near as many hours as I used to put in! When I was doing the bulk of z/XDC's development back in the '80s I'd put in 14-18 hours a day doing nothing but coding, and I was frustrated that the days were too short. I would run out of energy before I'd run out of the desire to write.

That's no longer the case. I find that as I've gotten older it's much harder to gain the focus and the "groove" where the programming comes forth. It's a lot easier to have distractions shatter the mental structures that you're creating. It's very frustrating.

Q: If you had it to do all over, what would you have done differently?

Dave: To me, that's a nonsensical question. Definitely, in my professional life, I have no regrets. But I do miss the connection with other people at my skill level that I had at Yale. I don't know if I could have created that and at the same time created Cole Software. When you get down to it, I'm not a Bill Gates. I'm just a pretty good programmer who got lucky.

Q: What would you suggest for newcomers to mainframe computing?

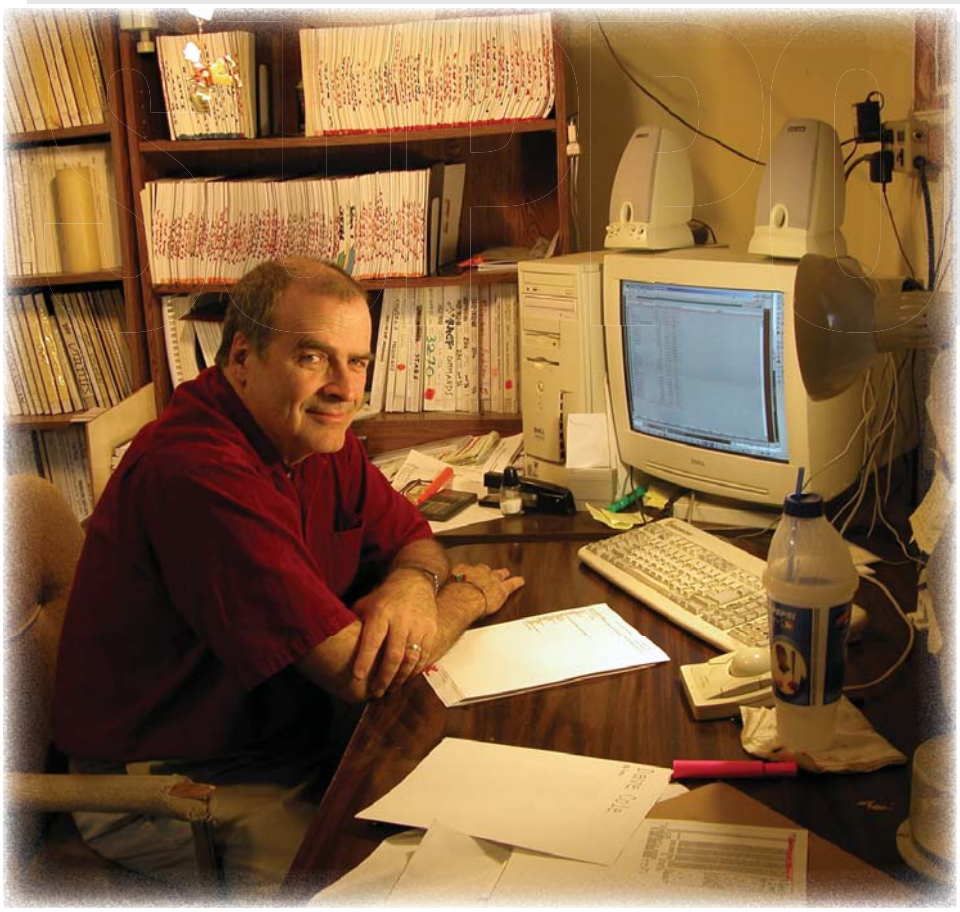
Dave: That's a hard one. The thing that attracts people to computers is a certain amount of "glitz," or "wow factor." Back when I was starting up, the wow factor was the existence of the computer itself. Nowadays the wow factor is what you can do with a PC. PCs are bright, glitzy, flashy devices. Mainframes are seen as stodgy and dull. The wow factor in the mainframe is very abstract. It's hard to get the attention of young people who are more interested in something immediate, like something they can see on the screen.

What would I recommend to someone starting out? The same as I'd recommend to anyone starting out in a career: If you're passionate about it, go for it. If you're passionate about something, then you will be successful at that. If you're just into it because it's something to do, then you may be successful—you may have a comfortable life, but you'll be mediocre.

You can't build passion, but you can let passion happen. If you're not passionate about it, then you're in the wrong business.

Q: What do you consider the basic skills necessary for a good programmer?

Dave: To be a development programmer takes an abnormal person. It takes a person who finds it enjoyable to do nothing but spend 14 or 18 hours a day typing on a keyboard conceptualizing and bringing to fruition abstract structures. That's not normal. People like that are, as they say, "Kind of a loner, kept to himself, didn't bother nobody…never woulda believed he could do something like this." I believe it to be a fundamental truth that every truly great programmer had an absolutely horrible childhood. I'm sure somebody's got a counter-example somewhere, but I think fundamentally it's true: If you want to hire a good programmer, you look for someone with lousy social skills!



Q: What other skills besides typing ability and abstraction are useful for a good programmer?

Dave: That's what it is: abstraction and analysis. Part of it is being able to follow in your mind a detailed rule-set. That's not the creative process. That's the desk-checking process.

The creative process is vague. It's being able to take an amorphously stated problem and break it down into its discrete,

individual programmable steps, going down deeper and deeper as you do your analysis.

Q: Don't you have to be methodical as well?

Dave: For sure! People say, "Cole is so smart. He's such a genius." Well, I'm not. I am not that smart. If someone hands me a Rubik's Cube, I can't solve it. I can't hold that many variables in my head to solve complex things.

I know someone named Mike Lewis who can solve a Rubik's Cube in about 25 seconds, no matter what state you hand it to him in. One time I realized that if you popped a corner off and turned it around, it would be impossible to solve. So I did that to him.

Mike got about fifteen seconds into the puzzle and handed it back to me, saying, "Someone has turned a corner around."

I'm not that smart, but what I am is methodical as hell, and I have an extremely high tolerance for tedium.

z/XDC is about 650,000 lines of code these days, and if you want to make a change that is systemic to the product, then you are faced with one or ten thousand discrete instances of the change. To a certain extent you can automate that, but a lot of it you can't. You have to be willing to sit down and just DO it.

For example, the z/XDC Messages Manual documents about 500-600 separate messages. Early on, I made the mistake of not keeping up with that, and then I was faced with the job of documenting 300-400 messages at one time. That's the kind of thing I can do—a tedious process for hours and hours and hours, or days and days and days on end. In the case of building z/XDC's 64-bit support, the tedium lasted for *years*!

The other thing is that I've been around long enough to have developed a rather large tool kit in my head. I know the kinds of things you have to do to accomplish a particular objective. I have an idea of the places to look and the kinds of things to look for to solve bugs! That's the other aspect of my skill. It's just a mental tool kit.

This is all very, very pedestrian. But if you put it together…In my case it led to a certain amount of success.

Q: What is your legacy—that which you'll leave for people coming along behind you?

Dave: I guess z/XDC. It will last at least for a little while. If the younger people I'm bringing onboard today are able to bring it into a more modern context, a broader context, then it will survive longer.

NaSPA member Robert Shimizu provides first-level technical support for Cole Software.