



Schnauzers, Modems, and Sound Techniques

My wife and I are proud owners of two toy schnauzers. For you dog lovers out there, I can see the email already: “There is no such breed as a TOY schnauzer.” It’s true. We have two miniature schnauzers – both runts. Together, they weigh about 18 pounds. Trust me; they are TOYS.

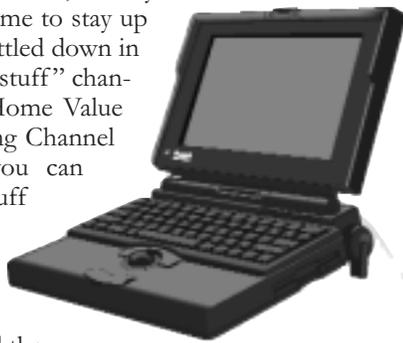
Funny, when I am traveling and away from home, they happily sleep the night away. However, when I am home, they seem to know that Daddy just “loves” to get up with them around 5 a.m. and take a short walk around the backyard. We have a nice walk, and soon we go back into the house.

The other night I had recently returned from a trip to the East Coast, and my body thought it was time to stay up – so the dogs and I settled down in the den to watch the “stuff” channel. You know that Home Value Quality Stuff Shopping Channel (HVQSSC), where you can order all kinds of stuff that you didn’t know you needed. That night, the HVQSSC was trying to convince me that I needed the newest and greatest computer on the market. While I was listening, half-asleep, the announcer was discussing the computer’s features. Understand, now, that while white teeth, a dazzling smile and a smooth voice are necessities for hawking products on the HVQSSC – technical competence does not seem to be a prerequisite.

The announcer, in an excited voice, told me that the computer comes complete with a “Model 56K model manufactured by Baud.” Not sure I heard it right, I immediately hung up the phone (from ordering the all-in-one left-handed apple corer and spaghetti curler previously advertised) and listened intently. Sure enough, the announcer again said that the computer came with a Baud modem. That’s sort of like saying that a new car comes with a supply of miles per hour.

I used to be a modem expert. Long ago I owned several off-brand modems until I shelled out about \$500 for a Hays 300 baud Smartmodem in 1982. Wow – 300 baud! Remember the Hays Smartmodem instruction set? To dial a number, you very carefully typed in “ATDT18005551212.”

The “AT” meant attention, and the “DT” meant tone dialing, and the number followed. “ATDP” meant pulse dialing. “ATS1” returned a string that contained the modems’ current internal settings. I eventually graduated from the Hays 300 Smartmodem to a 1200, then a 2400. Still, the 300 was like a first childhood crush. I have kept that 300-baud modem for more than 20 years now. It’s a nice souvenir of the way things used to be. It’s fun remembering how I used to start up the computer, boot DOS from a floppy, launch Procomm, and then use the modem to connect to the world.



Now, of course, you don’t have to know how to use a modem. The knowledge I once mastered and was so proud of is now pretty well obsolete. Is my knowledge now useless? Far from it – the knowledge allows me to understand what a modem really does, and what the operating system does for me. I don’t have to access the modem directly anymore, but I understand how the modem actually works. I know how the modem integrates into the operating system, and if necessary, can use the control panel to debug and hopefully fix a malfunctioning modem. It’s old knowledge, but still useful even with modern computers and operating systems. The knowledge might be old, but it’s tried and true. It’s sound knowledge (pun intended!).

This issue of CROSSTALK contains the winners of the Top 5 Quality Software Projects. I was privileged to be a reviewer for the projects, and was impressed with the quality of software that the Department of Defense puts out.

Did the projects succeed because they

incorporated some new language, tool, or technique? No, not really. The projects succeeded because of three issues: good people, good management, and sound techniques. Some used modern languages and techniques, but all used pretty standard stuff. Some used languages that have been around 20+ years. Some didn’t even mention what development methodology they used – but you can bet all used some type of life-cycle methodology. All had a risk management plan. All collected metrics, discovered trouble spots, and took preventative actions to get the project out the door to correctly meet user needs.

I remember teaching a computer science course a few years ago that used a book that stated “Good Algorithms + Good Data Structures = Good Programs.” That might work for computer science, but for software engineering, the equation reads “Good People + Good Management + Sound Techniques = Quality Product + Satisfied Customer.”

Every so often, I start to feel a bit out of touch. I talk to recent graduates who understand Java Beans, Extreme Programming, SuperVisual J++++, and other tools that are not in my repertoire. Then I think “big deal.” Not one of the Top 5 projects said, “We couldn’t have completed the project on time and under budget if it had not been for the marvelous GUIs that came with the polymorphic inheritance package of SuperVisual Cobol ++.” Instead, requirements management, risk management, and project management made the project succeed – sound, tried-and-true techniques.

You want to go back to being a hacker? Then go take a course, and learn some new language that probably won’t be around in 10 years. What, you would rather get a project out the door on time, under budget? You prefer a satisfied customer? Then quit worrying about being a hacker and keep on concentrating on sound techniques. You aren’t obsolete – you’re just sticking with sound techniques that you know still work.

–David A. Cook,
Software Technology Support Center
david.cook@hill.af.mil