Grace Murray Hopper: We Still Need You!

During my 15-plus years of writing BACKTALK columns, I have often found myself working a column while traveling. I am usually in one of three conditions: 1) bored, stuck in my hotel, not much to watch on TV; 2) bored, on yet another flight with no movie and a bad book that doesn't hold my attention; or 3) frustrated and bored, sitting in an airport, waiting for a late flight.

This week, I had the pleasure to attend the SIGCSE—Special Interest Group on Computer Science Education—in Milwaukee. Great conference, but, unfortunately, on my way home, the third condition applied. You see, I planned this conference around my spring break. Once I landed in Houston, I was driving to Orlando with my family to visit my parents. We were leaving as soon as I touched down in Houston, so I was in a hurry.

RADM Grace Murray Hopper

The weather in Milwaukee has been mixed all week, so I watched the weather closely—and on this morning we had fog, rain, and 20 mph winds. However, the airline assured me that the incoming and outgoing flights were on time. When I got to the airport, my departing aircraft was 15 minutes early—it was quickly deplaned, cleaned, the old-for-new luggage exchanged, and the boarding door was opened. All was right with the world.

One problem: There was no flight crew. I found out that while the plane itself had come in from decent Midwest weather, our flight crew was coming in from Newark where, apparently, Noah was floating down the main runway, picking up small animals left and right.

So unfortunately, while all the "hardware" was ready, something critical was missing—the human components.

At one of the SIGCSE workshops, we discussed Massive Parallelism: The analogy was that if one person can complete a jigsaw puzzle in two hours, two people could possibly do it in one. Maybe three could do it in 40 minutes. However, if you put the entire class of 20 students to work on it, it would probably take three or four hours. Human beings do not parallelize well. One of the instructors made a comment that Moore's Law applies to hardware only—and while one could argue that it takes 1.8 years to double software capacity, it takes 18 years to double software engineering capabilities. It's hard to significantly increase human capabilities. My feeling is that as newer and newer skills and languages and technologies come, older knowledge and associated skills decrease. And, given that most of us support the DoD in some way, we are sometimes working with technology that was developed and fielded *many* years ago. Face it: We *need* "legacy" skills to keep older systems up and running.

I have been involved in education and training for more than 30 years. I have taught all the latest trends and methodologies and processes as they develop and evolve. However, I find myself having to make a deep, dark confession: I am *still* teaching COBOL. In fact, I am teaching it in college this very semester. Why? Because there is still a real need for it in the "real world." In fact, there is a growing need! Mature and experienced developers (that's a nice way of saying "the old folks") are retiring, and large companies that have heavy investments in older languages over the years need new developers that can maintain and even expand this older code.

What happens when the current (and aging) pool of developers of legacy code dwindles, and fewer and fewer newly graduated developers know how to maintain older systems? Well, those who know (or are willing to learn) the older and more "traditional" languages will be in demand. Like that old Broadway song says, "everything old is new again".

While working on this column, I was buying an online ticket. The Web site kept rejecting my data for "unspecified reasons." Eventually, I was connected to one of their "application specialists" (short for developers) who said that the problem was that my house address contains the fraction "1/2", and their application could not handle the "/". I jokingly told the developer "Oh, storing the number in a PIC 9, not a PIC X, huh?" He laughed—and asked if I was available for some contract work. Seems they have a hard time hiring knowledgeable COBOL developers. If this anecdote—or, for that matter, the name Grace Murray Hopper[1]—doesn't register with you, maybe it's time to look again at those "legacy" languages.

Remember, there are no old languages—just job opportunities.

—David A. Cook, Ph.D.
Stephen F. Austin State University
cookda@sfasu.edu

Note

[1] To learn more about RADM Hopper (1906-1992)—including her role in COBOL—visit <www.chips.navy.mil/links/grace_hopper/womn.htm>. As stated in a December 1999 CROSSTALK feature called *Influential Men and Women of Software*, she was:

> ... creator of Common Business Oriented Language (COBOL). She was an officer in the Navy who became an Admiral. COBOL came about in the 1950s when the need for higher order languages was seen as a way to increase the productivity of programming computer applications.

Can You BACKTALK?

Here is your chance to make your point without your boss censoring your writing. In addition to accepting articles that relate to software engineering for publication in CROSSTALK, we also accept articles for the BACKTALK column. These articles should provide a concise, clever, humorous, and insightful perspective on the software engineering profession or industry or a portion of it. Your BACKTALK article should be entertaining and clever or original in concept, design, or delivery, and should not exceed 750 words.

For more information on how to submit your BACKTALK article, go to <www.stsc.hill.af.mil>.