Goodbye!

Robert L. Glass

It’s time for me to say goodbye to the readers of this column.

I’ve been pleased and proud to have written the Loyal Opposition column in *IEEE Software* for the last 11 years, but the time has come for me to park my word processor and make way for younger blood!

My Legacy

There are some things I’d like you to remember me by, things that are about my essence and the lingering essence of this column.

I’m both an academic and a practitioner. I believe that’s an important, balanced viewpoint that’s all too rare in our field. I like to tell people, “My head is in the academic world of computing and software, but my heart is in its practice.” I spent a lot of happy years building software (applications and tools) for several American aerospace companies, and I have a deep-seated belief in practitioners and their ability to build good software. I also spent some years in academe—as a professor in the pioneering software engineering graduate program at Seattle University, in the Software Engineering Institute’s education program, and in my current position as an honorary visiting professor at Griffith University in Brisbane, Australia. I think I understand both academe and practice.

I’m a contrarian. I feel strongly that our field hasn’t been as successful as it should in pursuing promising directions. That goes for both academic research (the failure of researchers to evaluate their findings in a practical setting before advocating them is scandalous) and practitioners (there’s no valid excuse for not holding retrospectives in which project successes and failures are identified and posted as lessons learned for the future). And I haven’t hesitated to speak out with my contrarian, sometimes-outrageous opinions.

I don’t believe in the software crisis. There are project failures, of course, but they are far fewer than those who write about them tend to imply. This is the Computing Era, an era that wouldn’t be possible were it not for the successful software that enables it. Most software projects, I deeply believe, are mostly successful. Data from such companies as the Standish Group that suggests otherwise should be seriously examined (and has been, as many academic researchers are beginning to question its validity).

I do believe in a software estimation crisis. I think it’s vital to separate software projects that fail to meet estimation targets from those that fail to meet functionality targets. Estimation is a deeply flawed activity in the software field:

- Estimates are typically provided by the wrong people. Several research studies have shown that most estimates are provided by upper management or marketing, the last people who have insight into what it will take to build a software product.
- Estimates are typically provided at the wrong time. The software life cycle dictates that estimates should be provided at the beginning of the life cycle, before the requirements analysis process—that is, before the problem is clearly understood! Furthermore, there’s usually no provision later in the life cycle for correcting such obviously flawed estimates.
- There’s little agreement on how estimates

Continued on p. 95
Continued from p. 96

should be provided. Approaches range from algorithmic (based on data from past projects) to intuitive (based on educated, experienced analysis). Most estimation tutorials suggest using both approaches.

I’m grateful to the researchers and practitioners who continue to pursue practical solutions to this software estimation crisis.

Choosing Sides
We engage in too many battles involving what I call “local loyalties.” We choose up sides between Microsoft and open source, between IBM and the old “Seven Dwarfs” (you won’t understand that one unless you’re nearly as old as I am!), between agile and disciplined approaches, between object and functional (and perhaps even structured) approaches, between favorite operating systems and text editors and other tools of the trade, between contemporary wisdom and older wisdom. I believe there’s no need for choosing sides. There’s wisdom to be gained from agile, open, contemporary believers as well as from disciplined, closed, and traditional believers. And it would be a shame to try to rely on only one side of these loyalties without understanding the other (I like to think of this as my “make love, not war” belief set).

I think we understand too little about computing research, even after we’ve been doing it for a half-dozen decades now. I mentioned that researchers, especially those in computer science and software engineering, do too little evaluation of the concepts they advocate. There’s been virtually no effort to match methodologies to applications, with the assumption resurfacing regularly that each new methodology addresses all the problems (I like to think of this as the extremely naïve “one size fits all” belief set).

What will I do after my Loyal Opposition ends? Continue to publish and edit my newsletter, the Software Practitioner, which is built on this same essence. I’ll enjoy being a visiting professor Down Under. And I hope to keep hearing from you as a result of these columns and my other writings, at rlglass@acm.org.

Robert L. Glass is the publisher/editor of the Software Practitioner newsletter and a visiting professor at Griffith University. He’s been in the software field since its inception in the 1950s, spending most of his time in the American aerospace industry but also as an academic. He has grey hair, but hopes you’ve found his viewpoints interesting in spite of that! Contact him at rlglass@acm.org.