

The State of the Practice of Software Engineering

Robert L. Glass, *Computing Trends*

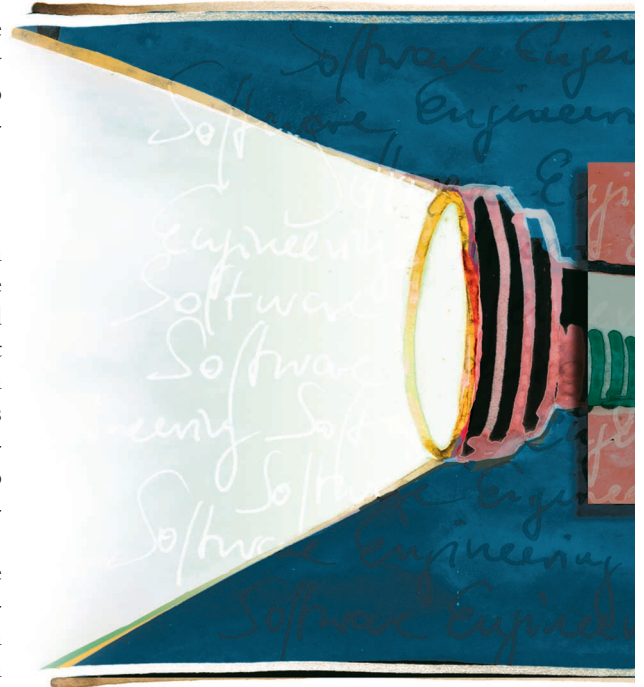
This special issue of *IEEE Software* is about the state of the practice of software engineering. If you're like many people who stumbled over this idea when I first presented it to *Software's* board, you're probably not at all sure what such an issue would be about, and why anyone should care. Please permit me to explain.

This issue's authors will describe the state of software practice circa late 2003. What they will *not* do—what I have studiously striven to prevent—is prescribe what the state of software's practice *ought* to be.

Why description?

There's a reason for sticking to description rather than prescription. For most of software engineering's history, authors have eagerly told practitioners what they ought to be doing. But rarely have those "oughts" been predicated on what practitioners actually are doing. This special issue aims to provide a baseline definition of practice, so that future authors who want to prescribe can at least have a description of what they're prescribing for.

As a career-long student of both software engineering theory and practice, I feel a particularly strong need to provide this description of practice. Defining the *state of the art* (which



I identify as theory + “best” practice) is easy; conferences, journals, and books do that for us all the time. Yet defining the *state of the practice* is difficult. Almost none of those conferences, journals, and books have much to say about what practice is about.

The right articles

Getting good material for this issue was an interesting exercise. Some authors and reviewers simply couldn’t switch from prescribing to describing. One well-known figure in the field warned me that “most academic researchers do not really know what goes on in practice” and that—worse yet—they think they do!

There were other problems along the way. I contacted the Standish Group, whose periodic CHAOS studies are so often cited by those who claim the existence of a “software crisis.” After an astonishing number of inquiring emails on my part, Standish eventually declined, noting a low “level of enthusiasm” for participation. (They charge US\$5,000 for the CHAOS reports, which probably accounts for the lack of enthusiasm for contributing a free version.)

I also hoped to get something from the ACM SIGSOFT’s Impact project, which is studying the effect of software engineering research on software engineering practice over the last 30 or so years. Once again, I failed to obtain an article based on this work (in this case, be-

About the Author



Robert L. Glass is president of Computing Trends, the publisher and editor of the *Software Practitioner* newsletter, editor of the Loyal Opposition column in *IEEE Software*, editor emeritus of Elsevier’s *Journal of Systems and Software*, and a certified “premier curmudgeon of software practice.” He has an honorary PhD from Linköping University, Sweden. Contact him at rglass@indiana.edu; he’d be pleased to hear from you.

cause the Impact project has made only limited progress).

But in the end, many authors and reviewers did “get” what this special issue is all about. I received invited articles (from Capers Jones, Don Reifer, and Mike Cusumano et al.) and contributed articles on such a diversity of topics as agile and open source development, Web and hypermedia projects, product line approaches, reviews and inspections, embedded systems, documentation, and project management. Then came the painful realization that we couldn’t accept all those contributions. The good news is that we could select an excellent collection of articles on an almost astonishing range of topics. The bad news is that we had to reject some equally excellent articles.

In the spirit of maximizing the number of excellent articles *Software* can publish, I will not try to summarize, article by article, what the authors have to say. They speak for themselves quite well.

I do want to say one final thing. My intended audience for this special issue is twofold:

- Practitioners, who often have their noses buried so deeply in today’s projects that they don’t have time to raise their heads and see what others like them are doing
- Researchers, who (as we’ve already seen) all too often simply “don’t know what goes on in practice”

I am personally offended by those who see the state of the practice as largely chaotic and unsuccessful. I hope this collection of articles will allow practitioners and researchers alike to get a more accurate perception of today’s state of the practice of software engineering. ☺

