Editor in Chief: Warren Harrison ■ Portland State University ■ warren.harrison@computer.org

Our 20th Anniversary

Warren Harrison

This is *IEEE Software*'s 20th anniversary and its 120th issue. During the 240 months since the first issue rolled off the presses, things have changed in both the world and our profession. In 1983, the Internet as we know it didn't exist, it was a toss-up whether Apple or PCs would become the dominant computing platform, and Java hadn't even been conceived.

I thought that this issue would be a good opportunity to gather some reflections from *IEEE Software*'s past editors in chief on both the magazine and the state of software engineering. (Steve McConnell, 1999–2002 EIC, declined because, as he pointed out, he gave his thoughts only a year ago in his Nov./Dec. farewell column.) This is truly a rare opportunity.

Bruce Shriver, 1983–1986:

When Oscar Garcia, the IEEE Computer Society's president, phoned me in 1983 asking if I would take on the job of launching a new periodical, *IEEE Software*, as its editor in chief, I accepted. It was a formidable task: the projected launch was less than a year away. Since then, much has changed in our industry; our ability to design, implement, test, and maintain complex systems has steadily improved. Nevertheless, some of my first editorial message, published in January 1984, still rings true today:

> *We still, by and large, lack the necessary methods to increase our ability to design and implement high-quality systems. We do not have, nor are we able to teach to those entering the field, methods to significantly increase programming productivity. We have not yet developed a specification and verification methodology and its supporting technology that can be used by the bulk of software practitioners.*

I went on to say,

> *We need to see rapid progress in tools and environments for specifying, designing, implementing, and testing systems; automatic programming; document preparation environments; operating systems, security, reliability, and services; and programming languages and the associated compiler technology for distributed and highly parallel systems.*

We are still wrestling with many of these issues, and I suspect that we will for many decades into the future, particularly since computer-based systems are infiltrating almost all aspects of our lives.

Each editorial board of *IEEE Software* since the inaugural issue along with the publications staff have had the ambitious goal of making it the preeminent magazine for software practitioners—that is, a magazine they can rely on to help them understand software and software-related issues. To this day, it continues to be informative, timely, technically stimulating, and useful. Congratulations to all those who have been involved with this 20-year effort.

Ted Lewis, 1987–1990:

In the late 1980s, *IEEE Software* was all about CASE, reusability, and object-oriented C++. Today, these technologies are as passé as MS-DOS and 256K RAM. 'Tis the sign of progress. Software development today is radically different from 20 years ago. In addition to the controversy over Extreme Programming and agile methods, the whole idea of large, monolithic desktop software is dead—replaced by Web-based technologies that are 10 times faster in development speed but requiring 10 times as much know-how.

The Microsoft monopoly may be the main reason proprietary desktop programming no longer holds much interest among developers. But in addition, the very idea of developing for a desktop platform is dead—the Web has become the platform. So, it's no longer good

0740-7459/03/$17.00 © 2003 IEEE

IEEE Software

Editorial: All submissions are subject to editing for clarity, style, and space. Unless otherwise stated, bylined articles and departments, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in *IEEE Software* does not necessarily constitute endorsement by the IEEE or the IEEE Computer Society.

To Submit: Access the IEEE Computer Society's Web-based system, Manuscript Central, at http://cs-ieee.manuscriptcentral.com/index.html. Be sure to select the right manuscript type when submitting. Articles must be original and not exceed 5,400 words including figures and tables, which count for 200 words each.

enough to know UML and a programming language. Rather, today's developer has to know numerous Web technologies, including Java, Java APIs (and there are lots of these), XML, LDAP, PKI, a half-dozen scripting languages, GUI programming, database programming, application frameworks, and more.

As for reuse, the number of reusable components to be found on the Web has exploded to the point where no single programmer knows them all. Tools have become so powerful that we can develop far faster, test and deploy, then go back to the design stage and redevelop again, more quickly than we can capture requirements even one time. And the knowledge needed to lay down one line of code has grown exponentially—perhaps outstripping a whole generation of developers' ability to grasp it all. I suppose this is why *IEEE Software* magazine is needed even more today than in 1988.

Carl Chang, 1991–1994:

During my tenure as EIC, I followed the road map described in my vision statement. First, I established an industrial advisory board. I was overwhelmed by the enthusiastic response of my industry colleagues to serve on the IAB. The editorial board's main objective was to transform *IEEE Software* into a true forum, where researchers and practitioners could meet and nourish each other. We planned many special issues during those four years that truly represented collaborative efforts between EB and IAB members. I trust that our readers also felt and enjoyed the energy radiating from our joint board meetings and activities.

Bruce Shriver
EIC 1983–1986

Ted Lewis
EIC 1987–1990

Carl Chang
EIC 1991–1994

We focused on software engineering issues pertaining to real-world complexity and reviewed techniques to assist both managers and developers. Notably during those years, renewed interest in object-oriented methods and better development and process support environments consumed much of our editorial resources.

In a conversation in 1992, Alan Davis and I determined that there were simply not enough results in requirements engineering to warrant a special issue on that topic. So, we decided to launch the new IEEE International Conference on Requirements Engineering and began publishing ICRE's "best papers" in *IEEE Software*. These "technology transfer" articles received complete rereview by the EB to meet *Software*'s own publication criteria.

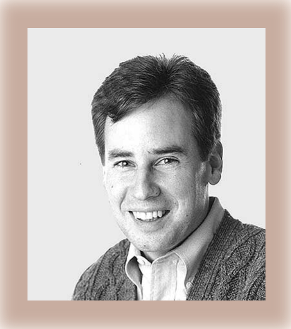
In recent years, more research has moved into the front end of the software engineering life cycle, so we tackled more "soft" software development issues such as requirements elicitation.

We also started a genuine volunteer-staff partnership. Managing editor Angela Burgess (now the Society's publisher) and I worked synergistically to coordinate many joint board issues and maintain a healthy supply of quality articles. It was a really enjoyable four years.

Al Davis, 1995–1998:

At one of the Editorial Board meetings during my tenure, one of the volunteers remarked, "The papers that appear in Volume 1 of the magazine are as applicable today [in the mid-1990s] as they were when first published. This is proof that we have been printing important, timeless, archival materials." As we should be!

Alan Davis
EIC 1995–1998



Steve McConnell
EIC 1999–2002

Warren Harrison
EIC 2003–

I added that although the observation might say good things about our magazine, it says poor things about our industry. What is our industry doing? If the relevant issues 20 years ago are all still relevant today, are we improving at all? Or are we just checking off boxes (for example, toward increasing levels of so-called maturity by some organization's definition) to make us all feel like we are making progress? Our mission as editors is not just to print interesting material but also to use the medium to affect our industry positively. Have the six of us done an acceptable job if we just shepherded the synthesis of significant words? I do not know where the blame rests (on us editors, the review process, the readers, or the industry's practitioners), but something is clearly broken if our problems are not evolving.

Don't get me wrong. I am proud to be a member of this austere family of fellow editors in chief, I still read every issue religiously, and I still feel it is the best magazine available for the "thinking practitioner." I just don't think it is yet time to celebrate a victory. The industry and the magazine have a long, long road ahead before we can even start to crawl, let alone walk upright.

These comments have provided interesting and at times divergent insights into our community, its evolution, and the role of publications such as *IEEE Software*. To some degree, we all said the same thing—the way we build software has indeed changed. On the other hand, as Al Davis points out, we haven't solved all the problems. Many of those we had in 1983 are still around today, albeit in different forms.

We still have trouble specifying products. Software is still not shipped defect free. And maintenance still makes up the bulk of software activities. However, we should consider that even though software development might change, its environment never will.

People specify products. We might give them some useful tools to make it easier, but ultimately, the customer (a person) must tell us what they want. Likewise, people write software. People make mistakes in other domains, so why are we surprised when they make mistakes building software? Again, we might come up with methods and tools to prevent or find mistakes, but until we get people who never make mistakes, expecting software to be defect free is a pretty high bar. And of course, there is maintenance. If the environment or mission evolves, we should expect that the software will, too.

Naturally we should expect things to get better, but I truly believe that there are some fundamental classes of problems that will never go away. We can chip away at them, but as long as we have to work within our environment, I don't think we'll ever totally solve them.

A peaceful and prosperous New Year to everyone. Please write me at warren.harrison@computer.org.