
Guest Editors’ Introduction

Setting the Standard

Carl K. Chang, University of Illinois, Chicago
Maarten Boasson, Hollandse Signaalapparaten BV
Tomoo Matsubara, Matsubara Consulting

his issue is dedicated to Alan Davis, our dear colleague, thinker, doer, leader, sometimes dreamer, and a congenial friend always.

Al surpassed the legacy that began in 1991 when the enthusiastic founding members of an Industry Advisory Board met with IEEE Software’s Editorial Board to hammer out a shared vision: to transform Software into a top-rated practice-oriented publication for software professionals worldwide. It has been a long march, exhausting two editors-in-chief. IEEE Software now enjoys a widespread reputation as a magazine for the readers, by the readers, and of the readers.

During Carl Chang’s term as editor-in-chief (1991–1994), the Editorial Board began an extensive dialogue with the software industry via the Industry Advisory Board. The IAB has been instrumental in strategic planning, enrichment, and development of Software’s editorial calendar. Through their extensive networking, IAB members have contributed not only real-world expertise but also numerous contacts that the Editorial Board alone would not be able to reach. Many IAB members actually took the lead to edit or co-edit special issues that focused on critical and timely themes identified at the annual joint EB–IAB meeting. Such a coherent model of editorial development proved to be extremely powerful and highly regarded by the publications community within the Computer Society.
AL’S CONTRIBUTIONS

As the magazine’s fourth editor-in-chief, Al deserves credit for many fine contributions. Here are just a few.

- He further strengthened the Editorial Board and expanded the IAB by recruiting many great minds known to the software industry.
- He enforced a policy on shorter articles. Our readers are busy people; they prefer to read short and relevant articles.
- He shortened the review cycle and employed a fast-track scheme to speed up the review of time-critical articles.
- He converted the conventional “theme issue” approach to “focus” sections, providing multi-perspective views on a particular topic.
- He originated the idea of using specific guidelines for various article genres—thus helping potential authors write the kinds of articles we want to publish. The guidelines also help reviewers approach their assigned article from a specific perspective, and using those categories in the magazine helps readers find the articles of most interest to them. These guidelines are available online at http://computer.org/software/genres.htm.

OUR STANDARD BEARERS

This focus section includes seven “standard” articles—standard in the sense that they conform well to the rules and guidelines defined by our genre specifications. (A Practice Tutorial article is lacking as a result of our rigorous review process.) We invited reputable authors from three continents—America, Asia, and Europe—to contribute to setting these standards.

Essay

In “The Java Virtual Machine—A Passing Fad?” Michael Franz explains why he believes that the Java Virtual Machine is probably on its way out. Microsoft may strive eagerly to make this happen, since the only hurdle is legal, according to the author. These predictions are based on Franz’s experience in developing Juice, an alternative scheme to support software mobility and transportability. It is definitely a thought-provoking piece, and so serves well as our standard essay.

Case Study

In “Tailoring Cleanroom for Industrial Use,” Robert S. Oshana reports Raytheon’s experience in a pilot process improvement program to move from CMM level 3 to level 5 by focusing on Cleanroom software engineering technology. The article shares the group’s encouraging lessons learned from tailoring Cleanroom to their specific needs. Oshana provides evidence to support their claims of success, but cautions readers who want to employ the same approach to their own organizations because concerns, issues, and barriers do exist.

Applied Research Results

Kathryn A. Bassin, Theresa Kratschmer, and P. Santhanam present “Objectively Evaluating Software Development” as our standard Applied Research Results article. By employing the Orthogonal Defect Classification scheme, the authors are able to support management with a firm handle on technical decision making. Through the extensive capture and analysis of defect semantics, information on project management, test effectiveness, reliability, quality, and customer usage can be obtained. The article describes three real-life case studies, and demonstrates the applicability of their techniques.

Experience Report

Daniel Hoffman and David Weiss’s article “Commonality and Variability in Software Engineering” serves as our standard Experience Report. The article describes how to perform domain engineering by identifying the commonalities and variabilities within a family of products. Through interesting examples dealing with reuse libraries, design patterns, and programming language design, the authors suggest a systematic Scope, Commonalities, and Variabilities approach to formal analysis. Their SCV analysis has been an integral part of the FAST (Family-oriented Abstraction, Specification, and Translation) technology applied to over 25 domains at Lucent Technologies.

How To

Tsuneo Yamaura presents our How To article, “How To Design Practical Test Cases.” Published ar-
articles seldom discuss real-world software testing in detail. Yamaura works at Hitachi Software Engineering, which is respected as a developer of high-quality software products—a feat rarely achieved at most software organizations. He reveals strategic and tactical secrets that his company has institutionalized, offers some data for actual testing activities, and describes how to achieve similar high-quality standards. He also points out the importance of designing and documenting test cases.

Tool Report
Mikio Aoyama’s “Web-Based Agile Software Development” is our chosen standard Tool Report. Although many of us realize the importance of tools in software development and have access to a variety of conventional ones, we have not tried to use integrated tools very much because of their seeming rigidity and complexity. Along with widespread use of the Internet and intranets, new ideas for building flexible integrated tools have been proposed. The author’s Web-based Agile Software Engineering Environment is grounded in a time-based rather than volume-based life cycle, and introduces the just-in-time concept.

Research Tutorial
“Exploring the Software Development Trilogy” by Daniel Le Métayer, Valérie-Anne Nicolas, and Olivier Ridoux serves as our standard Research Tutorial. Software development is concerned with more than just generation of code; the program has to have the desired properties, and these have to be demonstrated via suitable tests and correctness arguments. One interesting way of viewing these aspects is to group them into programs, properties, and data. When represented as vertices in a triangle, the edges represent processes to produce one element from another.

Most formal methods take into account these three aspects only partially; they also fail to restrict the expressiveness of predicate logic or programming languages. This results in problems that cannot be solved algorithmically. The research reported in this article aims at practical methods for automatic test generation by restricting the use of both predicate logic and programming constructs. The tradeoff between fully automated test generation and reduced expressive power is innovative and very interesting. It promises to eventually result in practical domain-specific programming languages, with a significant boost in both quality and productivity.

In the future, these published articles will serve as our genre examples for authors. The issue also includes a roundtable discussion: five contributors from around the world take up the issue of who should define standards and whether standards should even be defined at all.

We hereby present seven “standard” articles for your reference in the future. We feel this is a proper way to acknowledge Al’s efforts in defining the magazine’s article standards, and his constant effort to set high standards of quality in everything we published under his helm. This entire issue is meant to honor Al’s faithful service to our profession.

What we learned from Al’s unique leadership style is that one can lead successfully with sincerity.
He says what he thinks and he does what he says. He himself has set quite a standard of work ethics; we can only imagine how much energy he devoted to the job of editor-in-chief. We wish him well in his new endeavor as a practitioner (again).

Carl K. Chang directs the International Center for Software Engineering at the University of Illinois at Chicago. He is the 1998 secretary of the IEEE Computer Society’s Board of Governors and chairs its Magazine Operations Committee. He served as editor-in-chief of IEEE Software from 1991–1994 and established the society’s first Industry Advisory Board. He organized the 1997 President’s Roundtable commissioned to answer the needs of practitioners, and recently chaired the task force for “Year 2000 Model Curricula for Computing.” Chang earned a BS in mathematics from National Central University in Taiwan, an MS in computer science from Northern Illinois University, and a PhD in computer science from Northwestern University. He is a member of ACM, Sigma Xi, and Upsilon Pi Epsilon and a senior member of the IEEE. He can be reached at c.chang@computer.org.

Maarten Boasson is head of the Applied Systems Research Department of Hollandse Signaalapparaten, a division of Thomson CSF. In addition, he is professor of computer science at the University of Amsterdam, where he occupies a chair in distributed reactive systems. He developed a distributed system architecture, based on the shared data model, that is the basis of Signaal’s command and control products. Boasson received an MA from Groningen University. He is a member of the IEEE Computer Society, ACM, AAAI, and NGI. He can be reached at boasson@signaal.nl.

Tomoo Matsubara is an independent consultant in software technology, management, and business, and a Cutter Consortium faculty member. Prior to this he worked at Hitachi and Hitachi Software Engineering for 36 years. He is associate editor-in-chief of IEEE Software (first joining the IAB when Carl Chang organized it), editor of the Soapbox column, and an editorial board member of Cutter IT Journal and Information and Software Technology. His current interests are in applying theories, technologies, and best practices at software organizations, and in the social and industrial impacts of new technologies.

Matsubara received a BS in mechanical engineering from Waseda University. He is a member of the IEEE Computer Society. He can be reached at t.matsubara@computer.org.


**Figure A.** The frameworks quagmire. Red indicates CMM; green indicates US government or military documents; purple indicates international standards; blue indicates commercial, professional, or industrial associations, mostly US ones; and black indicates all others. The arrows indicate “derived from,” via information or people. Ovals distinguish the more popular and important frameworks.