Meeting the ever-increasing demands for software quality, reliability, cost-effectiveness, and — above all — more complex functionality is getting more difficult. Software reliability is especially important to customers today.

The need for reliability in critical defense, aerospace, and real-time military systems has received much attention, but surveys show that civilian customers also rate reliability as one of software's most important attributes.

Developers who must ensure a level of quality and reliability before product release obviously need techniques to evaluate reliability. Yet most have little knowledge of reliability models or their applicability and are unaware of their validation results or limitations.

Reliability modeling is still in a very early stage compared with other engineering disciplines, but interest is growing. One book on the subject by John Musa, Anthony Iannino, and Kazuhira Okumoto (see the box on p. 12), is so popular it has been translated into Japanese. And the IEEE Computer Society's Technical Committee on Software Engineering has formed a reliability subcommittee to organize an annual international symposium.

The number of application success stories is also growing. Organizations like AT&T, Hewlett-Packard, IBM, and Cray have realized quantifiable gains using reliability-growth models. Developers are obtaining more accurate results with these dynamic models than with static measurements like complexity. Reliability-growth models are used to quantitatively evaluate both the software and the tools used to develop it as testing progresses. These models are also used to monitor operational performance and control the effect of maintenance changes. Static measurements are used to estimate reliability before test.

There are many static and dynamic models and computational approaches, most using residual fault density and fault-encounter rate (or its inverse, mean time to failure) as reliability measures. Test coverage is also used sometimes, although its correlation with other measures has not been established. This and other aspects of reliability modeling are being investigated intensely.

IN THIS ISSUE

The articles in this issue examine the viability of reliability models in real-world development, analyze application results, and suggest techniques to improve available models.

The first article is by six practitioners, with much experience in reliability measurement, who have attempted to apply reliability theory. Their article addresses the gap between theory and practice and explains reliability modeling's place in the life cycle. It also in-
Most developers either aren’t familiar with reliability models or don’t know how to select and apply them. But the need for accurate predictions is acute, focusing attention on this comparatively young field.
SIGNPOSTS AND LANDMARKS: A SOFTWARE RELIABILITY READING LIST

Interest in applying software-reliability models is burgeoning. Articles, papers, and talks have been appearing in both software- and reliability-engineering literature.

In the recent past, software reliability was viewed as a subdiscipline of software engineering. Thus, *IEEE Transactions on Software Engineering* has published many articles on the subject. In particular, the special issue on software reliability (December 1985) contains many classic articles. Also, *Software Reliability Models: Theoretical Developments, Evaluation and Application* by Yashwant K. Malaiya and Pradip Srimani (IEEE CS Press, 1991) reprints some of the more important papers.

There is now more interest in reliability engineering, and more articles are appearing in *IEEE Transactions on Reliability*. In addition, more talks and papers are beginning to appear at the annual International Reliability, Availability, and Maintainability Symposium, the premier reliability conference. The proceedings for the last couple years are worth looking for (IEEE Press). The Rocky Mountain Section of the IEEE Reliability Society also sponsors an annual conference. Contact Rick Karchic at (303) 673-6223 or rick.karchic@stormtec.com.

Software reliability may be evolving as an engineering discipline of its own, so your key-list should include software-reliability engineering. A Subcommittee of the IEEE Computer Society’s Technical Committee on Software Engineering is dedicated to software-reliability engineering. It sponsors an International Symposium on Software Reliability Engineering, and the 1990 and 1991 proceedings (IEEE CS Press) are good sources. The third symposium is in October; contact Mahajan Vouk, (919) 515-7886. This Subcommittee also publishes a newsletter about practical applications of software reliability. Contact Hwe-Chu Tu, (908) 753-8806, to get on the mailing list.

For the serious student, a good textbook is *Software Reliability: Measurement, Prediction, Application* by John Musa, Anthony Iannino, Kazuharu Okamoto (McGraw-Hill, 1987). The book has two parts, the first oriented to engineering and practical application, the second to the field’s theoretical foundations.


This list is by no means exhaustive, but I hope it helps you get started learning about software reliability.

—William W. Favorri, Associate Editor-in-Chief

includes a dialogue from a panel session on whether the technology is ready for real-world use and where it is headed.

The next article, by Dick Hamlet, investigates the relationship between testing and reliability. He discusses the applicability of reliability models, how testing can and cannot improve reliability, and how trustworthy the results are from such an analysis.

Next, Norman Schneidewind and Ted Keller show how to properly validate a model’s predictions and interpret analysis results using a very interesting and important real-world environment, the space shuttle.

The next two articles offer techniques to improve the performance of existing models. In the first, Sarah Brookheur and Bev Littlewood solve the twin problem of not knowing which model to use and not knowing how much to trust its predictions. Their real-time algorithm analyzes prediction accuracy and enhances predictive power.

Then Michael Lyu and Allen Nikora present a technique that combines the results of multiple models for a better reliability estimate. They propose a family of linear combination models, which they claim have experimented with using historical data sets as well as real project data from the Jet Propulsion Laboratory at the California Institute of Technology.

Finally, Nachimuthu Karunanithi, Darrell Whitley, and Yashwant Malaiya introduce a new technique that uses neural networks. Neural networks are a comparatively new computation paradigm inspired by studies of the brain and nervous systems in biological organisms. The authors show that, given a failure history, neural-network models can automatically develop their own internal model of the failure process and predict future failures with comparatively more accuracy. They use data from a real project to illustrate their approach.

Researchers worldwide are pursuing better, more accurate models and methods to adapt them to real-world needs. We must analyze the experience reports to determine how we can achieve simple, accurate, technically sound, and experimentally verifiable techniques. Many interesting questions remain. We hope these articles provide some insight and encourage people to go further.

ACKNOWLEDGMENTS

We thank IEEE Software Editor-in-Chief Carl Chang for his encouragement and for guiding us through the process of editing this special issue. We thank the many authors who submitted their work and we are immensely indebted to the reviewers, who so generously provided timely feedback. Malaiya is supported by an NSF/DOE project monitored by Office of Naval Research.

REFERENCES


Pradip K. Srimani is a professor of computer science at Colorado State University. His research interests are reliability, parallel algorithms, fault-tolerant computing, networks, and graph-theory applications.

Srimani received a BSc in physics, a BTech in nuclear physics and electronics, and a PhD in computer science from the University of Calcutta, India. He is a member of the editorial boards of IEEE CS Press and the International Journal of Simulation. He is a senior member of IEEE and a member of ACM.

Yashwant K. Malaiya is a professor of computer science and electrical engineering at Colorado State University. His research interests are reliability, testing, fault tolerance, and testable microarchitectures.

Malaiya received a BSc from the Government Degree College, Damoh, India, an MSc in physics from the University of Saugar, India, an MSc Tech in electronics from the Birla Institute of Science and Technology, Pilani, India, and a PhD in electrical engineering from Utah State University. He is a senior member of IEEE, vice chair of the IEEE Computer Society’s Technical Committee on Software Engineering’s Software Reliability Subcommittee, chair of the IEEE Computer Society’s Technical Committee on Microprogramming and Microarchitecture, and editor of Microchips.

Address questions about this issue to Srimani, Colorado State University, Computer Science Dept., Fort Collins, CO 80523; Internet srimani@cs.colostate.edu.