Introduction to the Special Section on Software Maintenance

I am grateful to Professor C. V. Ramamoorthy, Editor-in-Chief, for suggesting this special section on a subject which has received relatively little attention from the research community. It is encouraging to see substantive work in maintenance going on, such as the work reported in this section. Our aim in this section is to redress in a modest way the neglect of the past by addressing some important issues in maintenance.

I am also very appreciative of the work done by the referees (listed later) who did such a comprehensive and tough review job that only four papers (all requiring substantial revision) out of twenty submitted papers were accepted, so few that it was necessary to publish this special section, rather than a special issue, as originally intended.

Finally, I want to thank the authors for their fine papers and for persevering through a lengthy review process.

The papers comprising this section are the following:

In the survey article, "The State of Software Maintenance," Schneidewind takes an irreverent look at the field of software maintenance and attempts to figure out where the field is and where it is heading.

In "Understanding Software Maintenance Work," Bendifallah and Scacchi address an area that has received little attention in research and practice: the ways local circumstances in the workplace affect how and why people perform software maintenance tasks and, conversely, how maintenance work affects workplace arrangements.

In "Maintaining Configurations of Evolving Software Systems," Narayanaswamy and Scacchi look at the challenging task of keeping track of the moving target of changing software systems so that one knows the state and can preserve the integrity of the system as it evolves.

In "The Use of Software Complexity Metrics in Software Maintenance," Kafura and Reddy explore the relationships between complexity metrics and the effects of maintenance activities on one system. Metrics computed from the source code or design representation of planned enhancements could be used to assess the effect of maintenance on structural complexity or help judge whether enhancements have been inserted in the proper place in the system’s structure.

In "A Controlled Experiment on the Impact of Software Structure on Maintainability," Rombach analyzes the impact of structure on the maintainability of a distributed operating system. He seeks answers to the questions: what are appropriate concepts for structuring distributed system software and what metrics obtained from the software can be used to explain or predict its behavior?

In their invited paper, "Knowledge Representation of Software Component Interconnection Information for Large-Scale Software Modifications," Yau and Tsai describe how automated reasoning techniques can perform validity and integrity checking of software component interconnection in a design specification when a user modifies a large-scale software system during development or maintenance.

Norman F. Schneidewind
Guest Editor

Norman F. Schneidewind (A’54–M’59–M’72–SM’77) received the B.S. degree in electrical engineering from the University of California, Berkeley, the M.S. degree in computer science from San Jose State University, San Jose, CA, the M.S. degree in operations research (engineering) and the Doctorate degree in quantitative methods from the University of Southern California, Los Angeles.

He has been working in software engineering since 1971 when he conducted research on software reliability models for the Fleet Combat Direction System Support Activity and the Naval Electronics Laboratory. He is a Professor of Computer Science and teaches and does research in software engineering and computer networks at the Naval Postgraduate School. He is the Chairman of the IEEE Standard for Software Quality Metrics Working Group, which is developing the first standard in this area.

Professor Schneidewind’s publications in software engineering have appeared in the IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, IEEE TRANSACTIONS ON RELIABILITY, and IEEE Computer. He is an Editor of IEEE TRANSACTIONS ON SOFTWARE ENGINEERING and Editor-in-Chief of the ACM/IEEE Fundamental Concepts in Software Engineering Series. He is a member of Eta Kappa Nu, Tau Beta Pi, and Sigma Xi, and holds the Certificate in Data Processing.