Software Maintenance

Guest Editors’ Introduction

Robert S. Arnold, MITRE Corporation
Roger J. Martin, National Bureau of Standards, ICST

Maintenance literature in the mid-1970’s to early 1980’s tried to define software maintenance and determine whether the term itself was appropriate. “Everybody complains about software maintenance but nothing is being done about it” was a common lament. Virtually any advance in software maintenance was thought to be meaningful because of high industry-wide maintenance costs.

Nowadays people realize that
(1) arguing about the definition of software maintenance seldom leads to maintenance advances,
(2) giving maintenance a new name does not decrease turnover among maintainers,
(3) much has, in fact, been done and written about software maintenance, and
(4) the industry-wide money-saving potential of a new maintenance technique is not by itself sufficient to justify marketing or procuring the technique.

Software maintenance needs have crystallized. Software maintenance shops are seeking suitable combinations of maintenance, management, and motivation technology for changing code within deadlines and within budget; achieving code quality standards (e.g., keeping code understandable, flexible, reliable); keeping documentation up to date; training new maintainers; and keeping programmer turnover low. At different levels, corporations rely on maintenance to keep their daily operations running smoothly, to keep software in a state that allows easy modification and reuse, and to facilitate strategic corporate transition to new software systems.

The call for better maintenance technology to meet these needs is being answered. Concern for software maintenance has motivated
• new ways to manage software change,
• maintenance tools,
• maintenance cost estimation techniques,
• changes in the maintenance work environment,
• changes in programming environments,
• design of a major new programming language (Ada),
• federal guidelines on software maintenance,
• an IEEE-sponsored effort to achieve a software maintenance standard,
• an IEEE-sponsored software maintenance workshop,
• an IEEE-sponsored software maintenance conference,
• numerous seminars and tutorial conferences,
• several books devoted to maintenance,
• a software maintenance periodical,
• a special issue of IEEE Transactions on Software Engineering devoted to software maintenance, and
• this special issue of IEEE Software.

In addition, computer science departments with a software engineering specialty are increasingly including software maintenance in their research.

The articles in this special issue, though primarily documenting recent maintenance research or pioneering new maintenance practice, deal with practical maintenance problems.

The articles
Understanding and modifying programs are fundamental to software maintenance and are influenced by
• the information about programs that
is available to the maintenance programmer,
- the format in which this information is presented,
- the media used to present the information, and
- the operations available for viewing and analyzing the information.

The first article, “Display Strategies for Program Browsing: Concepts and Experiment” by Ben Shneiderman, Philip Shafer, Roland Simon, and Linda Weldon of the University of Maryland, explores four strategies for improving the presentation of information to the programmer. The authors note that the... strategies do not change the functionality of systems, but allow the programmer to use them more efficiently. ... They match the programmer’s needs, provide contextual, general, and detailed information, integrate the tools, and free the programmer from tedious manipulation and context switching.

Software systems, particularly very large ones, tend to have an unexpectedly long lifetime. Thus maintainers are faced with deciphering the legacy of previous generations of maintainers. When this task becomes onerous, the software may need to be restructured. Such a restructuring task for a system with parts nearly 20 years old is described in the second article, “Using Modern Design Practices to Upgrade Aging Software Systems” by Robert N. Britcher and James J. Craig of IBM Federal Systems Division. A key to the authors’ restructuring approach is having a strong discipline for performing software redesign.

All too often maintainers are faced with hard-to-understand software for which documentation is missing or out of date. Re-creating the documentation is typically out of the question in deadline-driven maintenance shops because of the time required and the difficulty in understanding software previously maintained without programming style standards. The next article, “TMM: Software Maintenance by Transformation” by Guillermo Arango, Ira Baxter, Peter Freeman, and Christopher Pidgeon of the University of California, Irvine, gives a partially automated approach for re-creating missing documentation—in this case missing design information. By postulating and automating a software design methodology, the authors show how programmers can work backward from the software to re-create its specification and then, if they desire, redesign the software.

If we better understand the comprehension errors programmers make during maintenance, we can design better documentation and tools to reduce these errors. The final article, “Delocalized Plans and Program Comprehension” by Stanley Letovsky and Elliot Soloway of Yale University, examines maintainer comprehension errors when documentation about code interactions is spread throughout the code. The authors suggest that this problem involving software and documentation, called delocalized planning, can be mitigated by documenting the plans underlying software construction as well as the software itself. Perhaps someday a programmer may “ask” a variable to explain its presence in a piece of code and receive an enlightening reply!

**Origin of this special issue**

In November 1984, Editor-in-Chief Bruce Shriver suggested a special issue of *IEEE Software* on software maintenance based on papers to be published at the Conference on Software Maintenance—1985 (CSM-85), held last November in Silver Spring, Maryland. We served as program cochairs of CSM-85 and, as guest editors, selected the articles for this issue of *IEEE Software*. In August 1985, we ranked all papers to be published at CSM-85 according to CSM-85 referees’ reviews. We then considered every paper in terms of style and topic interest to readers of this magazine. Of the 28 published papers, 10 were selected for in-depth review by referees according to *IEEE Software* standards. Four of these papers were chosen for further development and publication here. We requested new titles from the authors so that future citations would not confuse the CSM-85 and *IEEE Software* versions.

**Robert S. Arnold** is a member of the technical staff at MITRE Corporation in McLean, Virginia. His interests are software maintenance, software restructuring, system transition planning, and software quality measurement. He has worked at IBM, the University of Maryland, and Sperry Corporation. Arnold is program cochair of the Conference on Software Maintenance—1987. He edited the *IEEE Tutorial on Software Restructuring* and was program cochair of the Conference on Software Maintenance—1985.

Arnold received a BA “with highest distinction” in mathematics and computer studies in 1975 from Northwestern University, Evanston, Illinois. He received an MS in computer science from Carnegie-Mellon University in 1977 and a PhD in computer science from the University of Maryland in 1983. He is a member of Phi Beta Kappa, Phi Kappa Phi, ACM, SIGSoft, and the IEEE Computer Society.

He can be reached at the MITRE Corporation, 1820 Dolley Madison Blvd., Mail Stop W414, McLean, VA 22102.

**Roger J. Martin** is manager of the Software Engineering Group of the National Bureau of Standards’ Institute for Computer Sciences and Technology. His interests are software management, software maintenance, and software reliability. He has worked for the Office of Management and Budget in the Executive Office of the President and for the David W. Taylor Naval Ship Research and Development Center.


Martin received a BS (1969) and an MS (1971) in computer science from Iowa State University. He is a member of ACM, SIGSoft, and the IEEE Computer Society. He received the 1984 Department of Commerce Bronze Medal Award “for valuable contribution to more effective management of maintenance in the federal government.”

He can be reached at the National Bureau of Standards, Bldg. 225, Room B266, Gaithersburg, MD 20899.