Empirical Software Engineering, 4, 195–198 (1999)

Introduction to the Special Issue: Papers from the Fifth International Symposium on Software Metrics

WILLIAM AGRESTI
Mitretek Systems

MARVIN ZELKOWITZ
University of Maryland and Fraunhofer Center—Maryland

The Fifth International Symposium on Software Metric (Metrics 1998) was held November 20–21, 1998 in Bethesda, Maryland. Over 120 participants from around the world met to discuss the latest theoretical and practical achievements in software metrics. The foundation of the symposium was a technical program featuring a keynote presentation, 29 technical papers, 5 state-of-the-art reports, and a panel on strategies for academic and industrial metrics professionals to work together. The technical papers spanned all aspects of software metrics featured in the Metrics 1998 logo: science, engineering, measurement, experimentation, and applications.

The Metrics 1998 Program Committee selected four outstanding papers, and invited the authors to extend and revise their papers for this special issue of Empirical Software Engineering. The selected papers illustrate the application of empirical methods and indicate the breadth of topics being addressed in software metrics. In the set of four papers, the reader will find the following:

- Metrics for assessing the quality of systems architectures
- Comparison of the results of computed risk metrics to opinions of human experts
- Design and use of wrappers to collect metrics on objects in distributed systems applications
- Application of G/Q/M to develop performance metrics for CORBA-based systems
- Development and analysis of change and churn metrics
- Validation of a metric for estimating the fault injection rate in a system
- Application of metrics to a 300,000 line real-time system
- Definition of novel object-oriented function points
- Method for estimating the size of object-oriented software projects

The first paper, “Metrics to assess the likelihood of project success based on architecture reviews,” by Alberto Avritzer and Elaine Weyuker, addresses the need for early indicators of the quality of systems by analyzing their architectures. Architecture reviews offer an excellent opportunity to provide constructive contributions to the quality of developed systems. Unfortunately, if the review is entirely qualitative, it is highly dependent on the talents, expertise, and insights of the reviewers. Ideally, reviewers would like to have quantitative help, in the form of metrics, to raise potential problem areas and focus attention, based on actual experiences with implemented systems. The authors present the results of analyzing 50 architectural audits to identify common problems. This study of problems and related issues led to their definition of a risk metric. The metric is applied to seven projects, and the results are compared to expert judgement of the architectures. This metric can be extremely useful if it continues to be refined through experience. It can provide much-needed quantitative assistance to an activity with high leverage to affect the quality of systems.

The Common Object Request Broker Architecture (CORBA) is very influential standard for the development of distributed systems. There is a significant need for metrics support to designers and developers of CORBA-based system. In “Collecting Metrics for CORBA-based Distributed Systems,” John D. McGregor, Il-Hyung Cho, Brian A. Malloy, E. Lowry Curry, and Chanika Hobatr investigate CORBA-based designs to identify measurable features. They look at the development of a transportation system, with the goal of wanting satisfactory performance and facilitating re-configuration. They adapt the mechanism of wrappers to provide the desired metrics. Using the Goal/Question/Metric paradigm, they define metrics that will answer critical questions associated with these goals. They also apply the methodology to collect metrics for a sample application. The authors have addressed a key concern with request broker technology in looking at performance. The results, including the novel use of wrappers, have the potential to contribute to industrial systems development projects.

Another timely and critical problem is addressed in the third paper, “Software evolution and the code fault introduction process” by Sebastian Elbaum and John C. Munson. The problem in question is the introduction of faults during a lifetime of system releases. Organizations want indicators when faults are being injected so they can take action while performing maintenance, rather than waiting until testing to repair the code because a failure has occurred. The authors define a metric for the incremental change in relative complexity and a second measure, churn, which track the magnitude of the change being made. They apply the churn metric to successive versions of a large, real-time system. Using regression analysis, they demonstrate churn offers significant explanatory power to account for the variation in faults. With its focus on the costly maintenance process and the prediction of faults, this paper addressed key issues for the software engineering community.

The fourth paper is “A function point-like measure for Object-oriented Systems,” by G. Antoniol, C. Lokan, G. Caldiera, and R. Fiutem. The authors define an adaptation of function points for object-oriented systems. With the popular use of both FP and OO, this paper should be of keen interest. In the basic set of FP counts, logical files and transactions need re-interpretation for use with OO systems. The authors explain the mappings of these entities to classes and methods, respectively. Then they define OO Function Points, implement an OOFP counter, and try their methodology in an industrial environment. The authors have contributed a flexible methodology that is shown in the reported pilot study to have promise for estimating the size of OO systems.

In summary, the papers represent a variety of empirical methods and a wide range of topics. The papers are very good representatives of the theme of Metrics 1998 to make substantive contributions to the solution of real problems faced by industrial software engineering, while applying novel analytical approaches.

Acknowledgments

As guest editors, we express our deep appreciation to the members of the Metrics 1998 Program Committee for sharing their expertise in reviewing the technical papers for the symposium. We especially thank the additional reviewers who provided detailed review comments and recommendations on the revised papers that appear in this special issue.

William W. Agresti is Chief Scientist in the Systems Technology Center at Mitretek Systems, Inc., in McLean, Virginia, providing software engineering technology support across the corporation. He recently returned from the National Science Foundation where he was Program Director for Experimental Software Systems. His research interests are in empirical software engineering and IT measurement. He previously held positions with the MITRE corporation, and with Computer Sciences Corporation, where he led software development and applied research projects for the Software Enginering Laboratory at NASA's Goddard Space Flight Center. He was technical manager of the NASA/CSC team that pioneered the use of object-oriented design and Ada for flight dynamics mission support. Earlier he was a Professor and Associate Dean of Engineering. He received a Ph.D. from New York University.

Marvin V. Zelkowitz is a professor of Computer Science at the University of Maryland holding a joint appointment with the University's Institute for Advanced Computer Studies since 1988. Since November, 1997 he has been Co-Director of the Fraunhofer Center for Experimental Software Engineering, Maryland, a non-profit corporation that has links with the Fraunhofer Gesellschaft institutes in Germany. His research institutes include environment design and empirical studies of software engineering practices. He is one of the directors of the NASA Goddard Space Flight Center Software Engineering Laboratory, which has been studying the effects of software engineering technologies in practice.

Prof. Zelkowitz was a previous chairman of the IEEE Computer Society Washington Chapter and a previous chairman of ACM SIGSOFT and of the Computer Society's Technical Committee on Software Engineering. He is a fellow of the IEEE, a Golden Core member of the Computer Society, and a member of ACM. He received the Ph.D. degree from Cornell University.