Empirical Software Engineering, 2, 109–118 (1997)

Methodologies for Performing Empirical Studies: Report from the International Workshop on Empirical Studies of Software Maintenance

CHRIS F. KEMERER ckemerer@katz.business.pitt.edu
Katz Graduate School of Business, University of Pittsburgh, Pittsburgh, Pennsylvania

SANDRA SLAUGHTER sandras@andrew.cmu.edu
Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, Pennsylvania

Abstract. The International Workshop on Empirical Studies of Software Maintenance workshop took place following the International Conference on Software Maintenance in Monterey, California. The focus of the workshop was on experimental quantitative and qualitative studies of software maintenance processes. Of particular interest were the design of empirical studies, their underlying methodologies and techniques, and the lessons learned from them. This is the paper resulting from the charge to the group "Methodologies for Performing Empirical Studies". A description of each paper in the group is presented, along with a summary of the discussion. The sessions were summarized as follows:

- continue to address so-called "stale" research questions
- carefully define all research constructs and models
- offer insights on the perceived level of generality of the research results
- use research methods well-suited to the problem, and use them rigorously
- combine methods where this would add significant additional insight—collaborate where necessary to achieve this goal
- don't ignore factors relating to maintainers (e.g., ability and experience) despite the known difficulties in their measurement
- maintain a strong linkage to practice to ensure the research's continued relevance.

The group concluded with a high degree of agreement and encouragement that the field was moving in appropriate directions.

Keywords: software maintenance, research methods, methodology, software maintainers, complexity, experience

1. Introduction

The International Workshop on Empirical Studies of Software Maintenance workshop took place following the International Conference on Software Maintenance in Monterey, California. It was sponsored in part by the Fraunhofer-Institute for Experimental Software Engineering (IESE), Kaiserslautern, Germany.

The focus of the workshop was on experimental quantitative and qualitative studies of software maintenance processes. Of particular interest were the design of empirical studies, their underlying methodologies and techniques, and the lessons learned from them. Controlled experiments, field studies, pilot projects, measurement programs, surveys or

analyses based on questionnaires, maintenance process models, etc., were all candidates for empirical studies of interest. Examples of applications are:

- maintenance cost models (changes, releases) (Basili, et al., 1996)
- reliability of maintained systems
- assessment of system maintainability based on measurement (Gill and Kemerer, 1991)
- models for impact analysis and their evaluation (Gode, et al., 1990)
- maintenance process assessment, modeling, and improvement (Banker, et al., 1991)

The objectives of the workshop were three-fold:

- promote discussion and exchange of ideas among researchers and practitioners in this area
- better identify practical problems and research issues
- share and identify existing and potential solutions

Participants submitted a brief position paper describing their position on one of the topics mentioned above or on any other topic deemed relevant by the author. The workshop committee selected the most appropriate position papers and clustered them into a limited number of topics.

Members of each group met and performed a structured synthesis of practical problems, research issues and solutions related to the topic of interest. The results of this effort were presented to all workshop participants. Each session had a moderator in charge of coordinating the effort and presentations of his/her group, introducing the speakers, and writing down a final position paper resulting from his/her group discussions.

This is the paper resulting from the charge to the group "Methodologies for performing Empirical Studies". Prior to meeting at the workshop, all members of the group exchanged position papers as a means of introducing themselves to the group. When the group met in person at the workshop, the first exercise was to review a shared data set to discuss potential analyses of the data. This grounded approach generated significant discussion around the issues of appropriate research questions, appropriate methods, and issues around presentation of results.

We begin by briefly summarizing the position papers written by the group members. We then extract key themes from the group discussion and summarize the main messages from the workshop group.

2. Position Papers

Meta-analysis: Unnecessary or Unworkable? Brooks, Andrew (University of Strathclyde, UK) (Brooks, 1996)

Brooks notes that meta-analyses are relatively common in medical and psychological research, but questions whether meta-analysis is necessary and workable in the software

maintenance area at this time. He considers these questions from the perspective of subject-based empirical studies. Meta-analysis is necessary when clinical significance is high and multi-site trials are required to establish a frequency distribution for the size of the treatment effect. However, to do meta-analysis, the replications of an experiment should be similar. In the software maintenance area, this is particularly difficult due to differences in method, task, technology and subject. Thus, unless the clinical significance of a treatment effect is high and resources are available, Brooks concludes that meta-analysis is not currently practical in the software maintenance area.

Need for More Longitudinal Studies of Software Maintenance Kemerer, Chris F. (University of Pittsburgh), Slaughter, Sandra (Carnegie Mellon University) (Kemerer and Slaughter, 1996)

Software maintenance occurs over a relatively long period of time. However, with few exceptions, empirical work in software maintenance has been cross-sectional in focus. Kemerer and Slaughter argue that there is a need for more longitudinal research of software maintenance. Longitudinal studies could reveal whether certain development practices are effective in reducing costs over the software life cycle, and could provide insight into maintenance management concerns. Kemerer and Slaughter are conducting a longitudinal research study that examines detailed maintenance cost and change histories for information systems from two commercial organizations. Their study examines whether software changes follow predictable patterns and whether the point of software replacement can be predicted. Issues include identifying other important, relevant and interesting research questions that can be answered using longitudinal data and the types of methods applicable to longitudinal data.

Industrial-Strength Software Quality Modeling Khoshgoftaar, Taghi M., Allen, Edward B. (Florida Atlantic University) (Khoshgoftaar and Allen, 1996)

Systems developers today use systems that integrate software measurements, quality models and delivery of results. These systems depend upon valid models of software quality. The goal of research by Khoshgoftaar and Allen is to build a set of methodologies for development of accurate, robust software quality models that are suitable for routine industrial use. A number of issues affect building and validating empirical software quality models. One issue is the need for an accurate problem reporting system. Often problem reporting systems are vulnerable to factors such as disposition, consistency, attribution and others that compromise the accuracy of the quality model. Another issue is that multivariate models are more accurate and robust than are models based on individual software product metrics. Yet, much of the literature on software metrics focuses on individual metrics. A final issue is that process measures significantly improve the accuracy and robustness of quality models, but these measures are often not included.

Strategies for Studying Maintenance Lethbridge, Timothy C. (University of Ottawa), Singer, Janice (National Research Council) (Lethbridge and Singer, 1996)

Lethbridge and Singer consider how to systematically study the maintenance process so that tools can be built to improve the productivity of software engineers. They argue that a maintenance study should investigate the mental models of maintainers and decompose their activities in significant detail. The study should look at the macro and micro levels, and should pose questions about time consumption, cognitive load, knowledge requirements and enjoyment. Finally, the study should look at how these aspects differ among maintainers and environments. The research by Lethbridge and Singer applies these ideas to the study of a group of software engineers who are evolving a large telecommunications system.

Early Risk-Management by Identification of Fault-prone Models Ohlsson, Niclas (Linkoping University), Eriksson, Ann Christin (AXE Systems Management), Helander, Mary (Linkoping University) (Ohlsson, et al., 1996)

Early identification of fault-prone modules is desirable both from the developer and the customer perspective since it supports planning and scheduling activities that facilitate cost avoidance and improved time to market. Large scale software systems usually involve modification and enhancement of existing systems. This greatly enhances development planning, since knowledge of previous releases can be used to improve processes for further projects. Ohlsson, Eriksson, and Helander present results from empirical studies at Ericsson Telecom AB which examine the use of metrics to predict fault-prone modules in successive product releases. Results show that such prediction is possible and has potential to improve project maintenance. More detailed models that incorporate process and resource aspects together with product attributes need to be defined and tested. Future work needs to focus on identifying attributes that should be included in the models and on developing methods for analyzing additive and multiplicative effects of several variables.

The Study of Software Maintenance Organizations and Processes Seaman, Carolyn B., Basili, Victor R. (University of Maryland) (Seaman and Basili, 1996)

Seaman and Basili argue that an important, but often overlooked, component of software maintenance processes is the organizational context in which they are enacted. A variety of approaches must be employed to study organizational issues in maintenance, including both quantitative and qualitative approaches. Seaman and Basili discuss examples of both approaches from the studies of software maintenance organizations. They conclude that quantitative methods are most appropriate when it is easy to define a unit of analysis, variation between individual units is minimal or well-understood, and the goal of the study is to find extensive and strong evidence to support a hypothesis. Qualitative methods are a good alternative when the above conditions do not hold, particularly in an exploratory study of a problem in which little previous work has been done. Most studies of software

maintenance fall somewhere between these two extremes, and there are a number of ways to combine quantitative and qualitative methods in such cases.

Methods for Studying Maintenance Activities Singer, Janice (National Research Council), Lethbridge, Timothy C. (University of Ottawa) (Singer and Lethbridge, 1996)

There are a number of methods for studying the work of software maintenance engineers. The goal of research by Singer and Lethbridge is to provide software engineers in an industrial telecommunications setting with a toolset to help them maintain the system more effectively. To do this, they began by intensely studying the software engineers. Their study involves a number of methods which they consider and evaluate. Possible methods include inquisitive techniques (brainstorming, questionnaires), observation techniques (think aloud protocols, shadowing, starship, fly on the wall), monitoring techniques (instrumenting systems, logbooks), historical analysis techniques (problem report analysis, comment/document analysis), and system/user modeling (system illustration, building a knowledge base). Each method has its advantages and disadvantages. The only way to get a truly accurate picture of maintainer's work is to use several methods at various points in the development cycle.

Qualitative Analysis of a Requirements Change Process El Emam, Khaled (Fraunhofer Institute), Hoelje, Dirk (Positron Inc.) (El Emam and Hoelje, 1996)

El Emam and Hoelje apply the qualitative analysis methodology to identify problems in an organization that maintains the requirements for a large real-time system. They identify the organization, process, and people problems using this methodology. Their intent is to set an improvement program based on the findings. Organizational problems included lack of communication of changes, and lack of information for critical decisions. Process problems included planning uncertainties, difficulties in getting participation from domain experts, and lack of documentation of the analysis. People related problems include turnover of key personnel, lack of organizational knowledge for new employees, and lack of system knowledge by domain experts. Interestingly, El Emam and Hoelje do not find any product type problems. They suggest a number of possible explanations for this result.

3. Themes

Research Questions

Although the primary agenda of this group was research methods, evaluation and choice of methods clearly depends upon the research question being asked, and therefore the discussion began with a discussion of research questions.

After some initial discussion, some agreement was reached around a suggestion by Nicholas Zvegintzov for the need to investigate what he termed "stale but dominant hy-

potheses". These are items that are taken as somewhat of articles of faith in the software maintenance community, despite the fact that they may have been the result of a single study, or, even in the case where the evidence is broader, in many cases such evidence was collected a sufficient number of years ago that it is not clear whether such evidence continues to accurately reflect the state of practice. Examples of such stale but dominant hypotheses would include:

- How much do organizations really currently spend on maintenance?
- How much code exists? What is its average age? What distribution of languages?
- Lehman/Belady related research. Rate of code growth? Rate of complexity growth? Decay/entropy rates? How much effect on maintenance productivity does a highly complex product have?
- 80/20 rules
- What tasks do people spend time on?
- Time spent on comprehension?
- Demographics of maintainers—age, technical experience, language experience, application experience?
- What skills are needed?
- What makes a great maintainer great?

These questions roughly group into three categories. The first set have to do with organizational maintenance activity. There are widely cited quotes about maintenance consuming 50% or more of organizational software activities. Is this still true? (Bannick, 1991) With common definitions of maintenance, what are typical averages? How do these activities break down into the Swanson and Lehman categories? How have these percentages changed over time? (Abran and Nguyenkim, 1991)

This latter question inspires a re-investigation of some of the early work of Lehman and Belady. (Chong Hok Yuen, 1987) How well do their "laws" predict the longitudinal behavior of software systems? Are there so-called "80/20" rules that would predict fault-prone or maintenance-prone components?

Finally, how do maintainers really spend their time? An early study by Fjelstad reported that half of all maintenance time was spent simply comprehending the existing system (Fjelstad and Hamlen 1983). Has this result been validated elsewhere? And, have years of greater attention to structured coding and machine generated code made code easier to comprehend? (Gibson and Senn, 1989) There has been a significant amount of perceived wisdom that good maintainers are many, many times better than average or poor maintainers. What skills/attitudes do they possess, and could these be taught/infused into other maintainers?

Research Approaches

This discussion of interesting research problems results in a list of factors that are potentially present in any research model of maintenance. These include:

- Product
- Task
- Processes
- Tools
- People

By product what is meant is the software artifact(s) upon which maintenance is to be performed. In contrast, the task is the work to be performed upon the product. For example, a researcher could be primarily interested in the product, e.g., how much effect on maintenance productivity does a highly complex product have? (Harrison, et al., 1982) versus a primary interest in the task, e.g., what is the relative residual error rate left by maintainers performing enhancement work versus repair work?

Beyond these two distinctions would be a focus on the maintenance process. For example, this could include studying the so-called 'A-type' versus 'W-type' maintenance organizational structures and their impact on maintenance cost (Swanson and Beath, 1990). These processes then can be further explored, with more detailed examination of the components of the processes, typically tools and people. Tools include debugging aids, for example, and research questions involving people would be as described in the previous section (Kuvaja, 1993). Overall, the sense is that there is a tremendous amount of unresolved questions, and therefore work in all of these areas is to be encouraged.

Research Methods

Attendees tended to work in one of three dominant ways: quantitative field studies, where data are collected in actual maintenance organizations, and then analyzed; quantitative lab experiments, where maintainers, typically students, are given controlled tasks and the results monitored; or qualitative studies, where other data are collected, typically from interviews with maintainers (Zvegintzov, 1988; Hagemeister, et al., 1992).

There was wide agreement that all of these approaches, properly executed, can enhance the level of knowledge in the field. And, after some discussion, no clear consensus emerged as to the 'proper' mapping of methods to factors. Items to be considered in choosing a method included the quality of available measures, the likely scale of the effect to be researched, any likely learning curves in the process, the strength of the external validity requirement of the research, and the ease of abstraction of the task.

Some agreement was reached that all of the problems seem to benefit when multiple methods are applied. Given that researchers are typically trained in, at most, one area, this suggests the clear need for collaborative, inter-disciplinary research. Examples of including

qualitative research might include a scenario where the first phase involved a qualitative case study, which, for example, might highlight potential constructs of interest, followed by an in-depth quantitative field study that attempts to measure and analyze those constructs. Another scenario might be a lab experiment, followed by a qualitative de-briefing of the subjects. Such de-briefings may unearth issues not initially considered by the experimenter in the course of laying out the research design.

Research Presentation

With a gathering of software maintenance researchers such as this one it was natural that participants would share their concerns and suggestions about how work in the field should be presented. While many items were raised, there seemed to be a shared sense of interest in having published research do a better job of drawing out broader implications from results and clarifying their applicability. In particular, the group issued a shared request to all working in this area to more clearly define terms and variables, so as to allow proper interpretation and, when desired, replication (Kemerer, 1995). In addition, there was also a general request that researchers, in their discussion sections, offer their views on the perceived generalizability of their empirical results. For example, a study of COBOL maintainers at a bank—do the authors believe that all of the results are specific to COBOL (for example, use of certain COBOL language constructs) or to banking? Or are there some results that are expected to apply to MIS-style systems in general? Or all commercial systems? Or all non-real time systems? Or to software maintenance in general (for example, maintainer experience?

4. Summary

The sessions were summarized as follows:

- continue to address so-called “stale” research questions
- carefully define all research constructs and models
- offer insights on the perceived level of generality of the research results
- use research methods well-suited to the problem, and use them rigorously
- combine methods where this would add significant additional insight—collaborate where necessary to achieve this goal
- don’t ignore factors relating to maintainers (e.g., ability and experience) despite the known difficulties in their measurement
- maintain a strong linkage to practice to ensure the research’s continued relevance.

The group concluded with a high degree of agreement and encouragement that the field was moving in appropriate directions.

Acknowledgments

Participants from the 1996 International Workshop on Empirical Studies of Software Maintenance contributed greatly to our report, especially those in Working Group 2: Andrew Brooks, Taghi M. Khoshgoftaar , Edward B. Allen, Timothy C. Lethbridge, Janice Singer, Niclas Ohlsson, Ann Christin Eriksson, Mary Helander, Carolyn B. Seaman, Victor R. Basili, Khaled El Emam, Dirk Hoelje, and Lionel Briand.

References

Abran, A., and Nguyenkim, H. 1991. Analysis of maintenance work categories through measurement. *Conference on Software Maintenance*, Sorrento, Italy.

Banker, R. D., Datar, S. M., and Kemerer, C. F. 1991. A model to evaluate variables impacting productivity on software maintenance projects. *Management Science* 37(1): 1–18.

Bannick, K. A. 1991. *Breakdown of Software Expenditures in the Department of Defense, United States, and World*. Naval Postgraduate School.

Basili, V., et al. 1996. Understanding and predicting the process of software maintenance releases. *Eighteenth International Conference on Software Engineering*, Berlin, Germany.

Brooks, A. 1996. Meta-analysis: unnecessary or unworkable? *International Workshop on Empirical Studies of Software Maintenance*, Monterey, CA.

Chong Hok Yuen, C. K. S. 1987. A statistical rationale for evolution dynamics concepts. *Proceedings of the Conference on Software Maintenance*.

El Emam, K., and Hoelje, D. 1996. Qualitative analysis of a requirements change process. *International Workshop on Empirical Studies of Software Maintenance*, Monterey, CA.

Fjelstad, R. K., and Hamlen, W. T. 1983. Application program maintenance study: report to our respondents. G. Parikh and N. Zvegintzov, eds. *Tutorial on Software Maintenance*. Los Angeles, CA: IEEE Computer Society Press: 11–27.

Gibson, V. R., and Senn, J. A. 1989. System structure and software maintenance performance. *Communications of the ACM* 32(3): 347–358.

Gill, G. K., and Kemerer, C. F. 1991. Cyclomatic complexity density and software maintenance productivity. *IEEE Transactions on Software Engineering* 17(12): 1284–1288.

Gode, D. K., Barua, A., and Mukhopadhyay, T. 1990. On the economics of the software replacement problem. *Proceedings of the 11th International Conference on Information Systems*, Copenhagen, Denmark.

Hagemeister, J., et al. 1992. An annotated bibliography on software maintenance. *Software Engineering Notes* 17(2): 79–84.

Harrison, W., Magel, K., Kluczny, R., and DeKock, A. 1982. Applying software complexity metrics to program maintenance. *IEEE Computer* 15: 65–79.

Kemerer, C. F. 1995. Empirical research on software complexity and software maintenance. *Annals of Software Engineering* 1(1): 1–22.

Kemerer, C. F., and Slaughter, S. 1996. Need for more longitudinal studies of software maintenance. *International Workshop on Empirical Studies of Software Maintenance*, Monterey, CA.

Khoshgoftaar, T. M., and Allen, E. B. 1996. Industrial-strength software quality modeling. *International Workshop on Empirical Studies of Software Maintenance*, Monterey, CA.

Kuvaja, P. 1993. Productivity of CASE technology implementation in SW development and maintenance on the third maturity level. *IFIP Conference on Diffusion, Transfer and Implementation of Information Technology*, Champion, Pennsylvania.

Lethbridge, T. C., and Singer, J. 1996. Strategies for studying maintenance. *International Workshop on Empirical Studies of Software Maintenance*, Monterey, CA.

Ohlsson, N., Eriksson, A. C., and Helander, M. 1996. Early risk-management by identification of fault-prone models. *International Workshop on Empirical Studies of Software Maintenance*, Monterey, CA.

Seaman, C. B., and Basili, V. R. 1996. The study of software maintenance organizations and processes. *International Workshop on Empirical Studies of Software Maintenance*, Monterey, CA.

Singer, J., and Lethbridge, T. C. 1996. Methods for studying maintenance activities. *International Workshop on Empirical Studies of Software Maintenance*, Monterey, CA.

Swanson, E. B., and Beath, C. M. 1990. Departmentalization in software development and maintenance. *Communications of the ACM* 33(6): 658–667.

Zvegintzov, N. 1988. High noon III: Continuing the quest for a true test of software maintenance tools. *Software Maintenance News* 6(1): 6–7.

Chris F. Kemerer's research interests include management and measurement issues in information systems and software engineering, and he has published articles on these topics in the leading professional and academic journals, including Communications of the ACM, IEEE Transactions on Software Engineering, Information Systems Research, Management Science, Sloan Management Review, and others.

Prior to accepting his current position as the David M. Roderick Chair in Information Systems at the University of Pittsburgh he was an associate professor at MIT's Sloan School of Management. He received the B.S. degree from the Wharton School at the University of Pennsylvania and the Ph.D. degree from Carnegie Mellon University.

He is a former Principal of American Management Systems Inc., the Arlington, Virginia-based software development and consulting firm. He has been invited to address audiences in eleven countries and numerous US cities. Dr. Kemerer serves or has served on the editorial boards of the Annals of Software Engineering, Communications of the ACM, Empirical Software Engineering, Information Systems Research, the Journal of Organizational Computing, the Journal of Software Quality, and MIS Quarterly.

Sandra Slaughter is an Assistant Professor at Carnegie Mellon University in the Graduate School of Industrial Administration, and teaches strategic management of information technology, systems analysis and design, and software project management. She received her doctorate in information systems from the University of Minnesota. Her research focuses on productivity and quality improvement in developing and maintaining information systems and on effective management of the information systems function. She has published a number of papers examining these issues in Management Science, Communications of the ACM, and Information, Journal of Software Maintenance and Technology and People. Her dissertation, Software Development Practices and Software Maintenance Performance: A Field Study, received the ICIS Best Dissertataion Award in 1995. Slaughter has more than a decade of industry experience in information systems, having worked for Hewlett-Packard as a financial systems planning analyst and for Allen-Bradley Company and Square D Corporation as an information systems project manager and analyst.