State-of-the-Art Survey

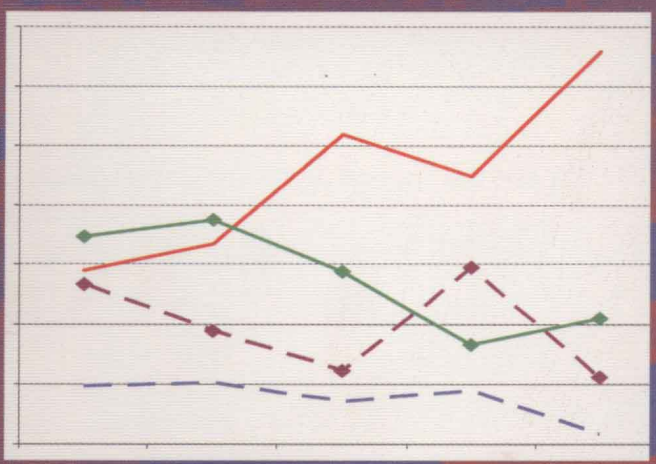
LNCS 4336

Victor R. Basili Dieter Rombach
Kurt Schneider Barbara Kitchenham
Dietmar Pfahl Richard W. Selby (Eds.)

Empirical Software Engineering Issues

Critical Assessment and Future Directions

International Workshop
Dagstuhl Castle, Germany, June 2006
Revised Papers



Springer

Victor R. Basili Dieter Rombach
Kurt Schneider Barbara Kitchenham
Dietmar Pfahl Richard W. Selby (Eds.)

Empirical Software Engineering Issues

Critical Assessment and Future Directions

International Workshop
Dagstuhl Castle, Germany, June 26-30, 2006
Revised Papers

Springer

Volume Editors

Victor R. Basili
Univ. of Maryland at College Park, 20742 College Park, MD, USA
E-mail: basili@cs.umd.edu

Dieter Rombach
Fraunhofer Institut, 67663 Kaiserslautern, Germany
E-mail: dieter.rombach@iese.fraunhofer.de

Kurt Schneider
University Hannover, 30167 Hannover, Germany
E-mail: kurt.schneider@inf.uni-hannover.de

Barbara Kitchenham
Keele University, ST5 5BG, Staffordshire, UK
E-mail: barbara.kitchenham@cs.keele.ac.uk

Dietmar Pfahl
University of Calgary, T2N 1N4 Calgary, Canada
E-mail: dpfahl@ucalgary.ca

Richard W. Selby
Northrop Grumman Space Technology, 90278 Redondo Beach, CA, USA
E-mail: rick.selby@ngc.com

Library of Congress Control Number: 2007922331

CR Subject Classification (1998): D.2, K.6

LNCS Sublibrary: SL 2 – Programming and Software Engineering

ISSN 0302-9743
ISBN-10 3-540-71300-X Springer Berlin Heidelberg New York
ISBN-13 978-3-540-71300-5 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12032105 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 4336

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Preface

Victor R. Basili, Dieter Rombach, and Kurt Schneider

Introduction

In 1992, a Dagstuhl seminar was held on "Experimental Software Engineering Issues" (seminar no. 9238). Its goal was to discuss the state of the art of empirical software engineering (ESE) by assessing past accomplishments, raising open questions, and proposing a future research agenda.

Since 1992, the topic of ESE has been adopted more widely by academia as an interesting and promising research topic, and in industrial practice as a necessary infrastructure technology for goal-oriented, sustained process improvement. At the same time, the spectrum of methods applied in ESE has broadened. For example, in 1992, the empirical methods applied in software engineering were basically restricted to quantitative studies (mostly controlled experiments), whereas since then, a range of qualitative methods have been introduced, from observational to ethnographical studies. Thus, the field can be said to have moved from experimental to empirical software engineering.

We believe that it is now time to again bring together practitioners and researchers to identify both the progress made since 1992 and the most important challenges for the next five to ten years.

Objectives

The purpose of this workshop was to gather those members of the software engineering community who support an engineering approach, based on empirical studies, to:

- Identify the progress of ESE since 1992 (Dagstuhl Seminar No. 9238*)
- Summarize the state of the art in ESE
- Summarize the state of the practice in ESE in industry
- Develop an ESE roadmap for research, practice, education and training

Three sessions were set up to discuss some of the most eminent challenges in ESE:

1 The Empirical Paradigm (Lead: Dieter Rombach)

The two topics addressed in this session were "Approaches for Empirical Validation" and "Exploration versus Confirmation." The session was complemented by a historical review.

Approaches for Empirical Validation
Numerous types of approaches are used for empirical validation. Questions relevant in this context include:

Which approaches are useful in what situation? How do we combine quantitative and qualitative studies? Which new and innovative approaches have surfaced lately?

Exploration versus Confirmation
Empirical studies range from exploring new, badly understood software engineering approaches for the purpose of incremental learning to confirming well-defined software engineering approaches. This raises a number of questions:
Which of these study forms is appropriate under what circumstances? How do we deal with validity threads (especially for exploratory studies)?

Historical Review
Guest Speaker: Mike Mahoney, a science historian from Princeton, provided a historical review on how other disciplines have dealt with the issue of "exploratory versus confirmatory studies." He then suggested some ideas for software engineering.

2 Measurement and Model Building (Lead: Victor Basili)

The two topics addressed in this session were "Data Sharing" and "Effective Data Interpretation."

Data Sharing
We have not yet established clear rules for handling ownership of empirical data – like those in other disciplines such as physics. This raises the following questions:
What is the value of empirical data, testbeds, and other study artifacts? How should they be shared for replication? What are the limits of sharing? How should ownership be recognized in publications? How can the probability of misuse be minimized?

Effective Data Interpretation
Proper interpretation of empirical data is a challenging task. The most obvious questions in this context include:
What are scientifically acceptable means of interpretation? What new approaches (e.g., visualization, simulation) can help? How do we combine evidence from individual studies into more abstract evidence?

3 Technology Transfer and Education (Lead: Kurt Schneider)

The two topics addressed in this session were "Technology Transfer" and "Education." They are two facets of making an impact with empirical work.

Technology Transfer
Empirical studies can be used to facilitate and speed up technology transfer into practice. Relevant questions in this context include:
How do we package results for different purposes and contexts? How can we use empirical software engineering to speed up technology transfer? What information do practitioners need from empirical studies? What is most convincing for practitioners?

Education
Empirical methods have to be taught, on the one hand; on the other hand, empirical studies can be used to teach other computer science and software engineering topics better. This raises a number of questions:
What are the experiences for effectively teaching empirical study competence? What are the experiences and approaches for integrating empirical studies into computer science or software engineering curricula? What educational methods are suitable for teaching empirical or evidence-based software engineering? How can education be improved?

Workshop Organization

To address the aforementioned challenges, an international workshop on the topic of "Empirical Software Engineering" was organized and held at the International Conference and Research Center for Computer Science (IBFI) at Dagstuhl Castle in Germany. The motivation for this workshop was to provide a forum for a relatively small but representative group of leading experts in software engineering with an emphasis on empirical studies from both universities and industry to meet and reflect on past successes and failures, assess the current state of the practice and research, identify challenges, and define future directions. An Organizing Committee identified key topics and key people who should participate in the workshop. The topics were chosen for discussion along with the session chairs and the people who were to present introductory talks. A final session was aimed at devising a roadmap for future work.

After the selection of discussion topics, introductory talks, and session chairs, approximately 30 more participants were invited to submit position statements on the selected topics. The participants came from Europe, the USA and Canada, Asia, and Australia.

The workshop was scheduled to run from Tuesday, June 26, 2006, through noon on Friday, June 30, 2006. Three full-day sessions were devoted to the topics listed. A half-day session was devoted to the development of a roadmap.

Session Organization

During each session, the Chair set the stage, by summarizing the state of 1992 and introducing the two selected topics. Two introductory presentations per topic set the tone and raised issues for discussion by identifying the progress made since the 1992 Dagstuhl Workshop, raising issues and challenges for the next five to ten years, and providing a list of provocative statements. Based on the issues raised by the introductory presentations and additional issues raised by other seminar participants, lively discussions took place. The discussions were deepened in up to four parallel working groups. Each working group was asked to provide a summary of their discussion. The session was concluded with the presentations of the working groups and final discussions. The material contained in this volume includes, for each session, the keynote address, introductory talks, position papers, summaries of the working groups, and a discussion summary.

Results

In session 1, on the empirical paradigm, the topics of approaches for empirical validation and of exploration versus confirmation were discussed. Workshop participants agreed that the community has matured since 1992 but is still in a very early phase compared to other disciplines. Improvements were suggested, among others, with regard to types of studies (e.g., longitudinal case studies), complementary usage of quantitative and qualitative studies, and theories for aggregating results across studies.

In session 2, the topics of data sharing and effective data interpretation were discussed in the context of measurement and model building. Clearly, as a community we need to build on each other's work in order to build models that represent the knowledge of the discipline better. We need effective approaches to interpret these models across multiple domains, environments, contexts, etc. Workshop participants agreed on these goals but did not always agree on how to achieve them. Discussions ran from debates on the necessity of protocols, to the use of open source data as opportunities of study, to ways of combining the results from individual studies, to building theories based on multiple studies.

In session 3, technology transfer and education were discussed. These issues are tightly interwoven. Good empirical education will lead to competent graduates. In their industrial careers, they are more likely to use and adopt empirical results than others. However, this will be a long-term effect. Workshop participants agreed there need to be explicit, short-term technology transfer mechanisms. Discussions and working groups discussed achievements and pointed to future research agendas.

Past Achievements

Since 1992, the community's understanding of how to perform empirical studies in the area of SE has improved. Consequently, the amount of empirical study activity has grown dramatically and there are many more sources of data and results from studies than there were then. Furthermore, the community was able to provide a body of knowledge from empirical studies in a few software engineering areas (e.g., inspections).

There have been a fair number of collaborations, some sharing of data, attempts at combining evidence from various studies and even the beginnings of the development of theories. Some of these results come from the enlargement of the community of researchers and some come from the availability of vehicles for publication and sharing of knowledge, e.g., ISERN: the International Software Engineering Research Network (started in 1993), the *Journal of Empirical Software Engineering* (started in 1996), and the International Symposium on Empirical Software Engineering (started in 2001). Some of the data have begun to be shared by different research groups, often involving some form of collaboration. But there are problems associated with successful sharing, e.g., regarding the overhead. Open source has become a major source of new data and opportunity for study. Various mechanisms are being studied for combining results of individual studies using experience bases and evidence-based approaches. Very little has been done in building theories from multiple studies but the problem has been identified as an area for further research.

Since 1992, empirical methods have been taught at many universities today. All courses discussed were created and designed after the 1992 workshop. The necessity to study empirical work in software engineering was seen on different levels of education, from Bachelor to graduate levels. Today, there is a clear shift to expecting empirical validation of results in software engineering publications. Papers are less likely to be accepted without them. PhD students, the potential next generation of researchers, are exposed to ESE techniques in a growing number of institutions.

Empirical validations do not necessarily require controlled experiments. During the last few years, well-prepared case studies in a realistic environment added an important facet. In a real industry project, a technique can prove its scalability and its fit to a specific industry context. Therefore, collaborations between academia and industry have been widely acknowledged in the meantime. They are a key to making an impact with empirical results. This awareness has grown significantly since 1992.

Key Points of Dissent

Probably the biggest source of dissent was the need for protocols in the sharing of data. Some felt that there should be some organized set of protocols for sharing data that would promote the sharing activity and protect the integrity of the data, while others felt a more free form of exchange was more appropriate. The concept of whether the experimenter owned the data was hotly debated and the view that programs are often considered the intellectual property of the developer was considered a detriment to data sharing.

Much empirical work is based on identifying problems, goals, or hypotheses, and then identifying the kind of design that should be applied and data that should be collected. There was some concern that studying open source was more of a bottom-up approach and that the data needed might not always be available, leading to a poor study or a change of focus for the study to satisfy the data that are there.

Although there is a general trend towards offering courses on empirical techniques, there is no consent on their contents or structure. Depending on the teachers' attitude, courses either tend to be focused on statistical methods or on conveying a more general view of research and evidence. Different books on empirical results emphasize different styles. Stimulating reflection in students is the common driver of both attitudes.

Collaboration with industry is seen as an important prerequisite. Researchers want to transfer their results to industrial use. Often, this is associated with funding through a company. Many see consulting and industry validation almost as synonyms. However, some participants stressed the two-way character of such a collaboration. Not only will practitioners learn from researchers, but researchers will also need to learn from practice: What are relevant research questions? What are valid arguments to sell a result? Researchers and industry experts need to meet on middle ground.

Important Topics for Future Work

A historian's perspective yielded the insight that ESE should not feel that it is behind, but there is a normal progress in the maturing of science that, as history shows, takes some time; ESE still needs to probe / explore what the important factors are in

software engineering. Much of science rests on engineering experience and ESE needs to accept that wrong models can be useful to advance science.

Generalization remains a major issue for controlled experiments performed under laboratory conditions. Generalization can be approached by building (logical) models. Extend the body of knowledge from empirical studies by incorporating the whole range of empirical evidence, ranging from controlled experiments to longitudinal case studies, by also integrating quantitative and qualitative research methods.

To better support reuse of empirical knowledge (combination of results), standardized ways, not only for performing empirical studies but also for reporting the results, have to be agreed upon.

Combining the results of various studies is necessary to evolve our understanding of the software engineering discipline. There is a need to share data and artifacts across research groups so that results of varying studies can be analyzed, models built and evolved, and influencing variables identified. An article offering various protocols for the sharing of data and artifacts has been published in the *Journal of Empirical Software Engineering* (2007). The future work involved here is to test out the protocols and evolve them based on experience.

In many ways, empirical study in software engineering is opportunistic, and we need to find those opportunities for data and artifacts. Open source is one such opportunity. We need to experiment with mechanisms that allow us to use open source projects in more effective ways, e.g., by letting an empiricist be part of the study from the beginning.

Identifying and experimenting with ways to combine data from multiple studies are one of the most important areas of research if we are to be successful in building effective models in the software engineering discipline. We still struggle with how to do it effectively.

Abstracting from models to theories is a natural progression for most disciplines. Research is needed in approaches to building theories, representing them, documenting them, schematizing them, and evaluating them.

Teaching empirical techniques requires good teaching material, such as textbooks. Sample documents and free data would be of great help to try established empirical techniques. Participants suggested exchanging study material on specific Web sites. These could be shared among the empirical research community.

Establishing and maintaining collaborations between industry and academia will be even more important in the future. Many (junior) researchers find it challenging to approach a company successfully. Even senior researchers admit there can be many failures and disappointments before a fruitful collaboration emerges. More guidance and more examples in the area of collaborations would lower the threshold.

How to select rewarding topics for ESE research was discussed several times. While research had important successes in fields such as reviews and inspections in the past, many believe the agenda needs to be broadened. Different schemes were proposed to visualize current areas of work. Industry will have a say in defining future research roadmaps.

Established, well-understood areas of empirical work, such as inspections, make good candidates for a common repository of education material. Every student who wants to become a mature empirical researcher should replicate some of the classical inspection studies!

Acknowledgements

It would have been difficult to organize a workshop like this without help from the community. First, we would like to thank all the workshop participants for their effort in submitting position papers and participating in the discussions. Next, the presenters of the introductory talks and the session chairs as well as the chairs of the working groups deserve a special thank you for their contributions to the success of the workshop.

Clearly, an effort such as this could not have been successful without financial support. We would like to acknowledge and sincerely thank IBFI for supporting accommodation and providing excellent meeting facilities and the Fraunhofer Institute for Experimental Software Engineering (IESE) for additional general purpose funds. Our special thanks go to Marcus Ciolkowski from University Kaiserslautern, Sira Vegas from the Universidad Politécnica de Madrid, and Andreas Jedlitschka from Fraunhofer IESE for having taken notes of all discussion sessions. Furthermore, we would like to thank Andreas Jedlitschka and Marcus Ciolkowski for their splendid job in providing organizational support, arranging the social event, and last but not least for editing these proceedings.

September 2006

Victor R. Basili
Dieter Rombach
Kurt Schneider

Reference

[1] Rombach, H.D., Basili, V.R., Selby, R.W. (Eds.): Experimental Software Engineering Issues: Critical Assessment and Future Directions, International Workshop Dagstuhl Castle, Germany, September 1992, Springer Verlag, LNCS 706

Lecture Notes in Computer Science

For information about Vols. 1–4310

please contact your bookseller or Springer

Vol. 4429: R. Lu, J.H. Siekmann, C. Ullrich (Eds.), Cognitive Systems. X, 161 pages. 2007. (Sublibrary LNAI).

Vol. 4424: O. Grumberg, M. Huth (Eds.), Tools and Algorithms for the Construction and Analysis of Systems. XX, 738 pages. 2007.

Vol. 4420: S. Krishnamurthi, M. Odersky (Eds.), Compiler Construction. XIV, 233 pages. 2007.

Vol. 4415: P. Lukowicz, L. Thiele, G. Tröster (Eds.), Architecture of Computing Systems - ARCS 2007. X, 297 pages. 2007.

Vol. 4414: S. Hochreiter, R. Wagner (Eds.), Bioinformatics Research and Development. XVI, 482 pages. 2007. (Sublibrary LNBI).

Vol. 4405: L. Padgham, F. Zambonelli (Eds.), Agent-Oriented Software Engineering VII. XII, 225 pages. 2007.

Vol. 4403: S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, T. Murata (Eds.), Evolutionary Multi-Criterion Optimization. XIX, 954 pages. 2007.

Vol. 4397: C. Stephanidis, M. Pieper (Eds.), Universal Access in Ambient Intelligence Environments. XV, 467 pages. 2007.

Vol. 4396: J. García-Vidal, L. Cerdà-Alabern (Eds.), Wireless Systems and Mobility in Next Generation Internet. IX, 271 pages. 2007.

Vol. 4394: A. Gelbukh (Ed.), Computational Linguistics and Intelligent Text Processing. XVI, 648 pages. 2007.

Vol. 4393: W. Thomas, P. Weil (Eds.), STACS 2007. XVIII, 708 pages. 2007.

Vol. 4392: S.P. Vadhan (Ed.), Theory of Cryptography. XI, 595 pages. 2007.

Vol. 4390: S.O. Kuznetsov, S. Schmidt (Eds.), Formal Concept Analysis. X, 329 pages. 2007. (Sublibrary LNAI).

Vol. 4389: D. Weyns, H.V.D. Parunak, F. Michel (Eds.), Environments for Multi-Agent Systems III. X, 273 pages. 2007. (Sublibrary LNAI).

Vol. 4385: K. Coninx, K. Luyten, K.A. Schneider (Eds.), Task Models and Diagrams for Users Interface Design. XI, 355 pages. 2007.

Vol. 4384: T. Washio, K. Satoh, H. Takeda, A. Inokuchi (Eds.), New Frontiers in Artificial Intelligence. IX, 401 pages. 2007. (Sublibrary LNAI).

Vol. 4383: E. Bin, A. Ziv, S. Ur (Eds.), Hardware and Software, Verification and Testing. XII, 235 pages. 2007.

Vol. 4381: J. Akiyama, W.Y.C. Chen, M. Kano, X. Li, Q. Yu (Eds.), Discrete Geometry, Combinatorics and Graph Theory. XI, 289 pages. 2007.

Vol. 4380: S. Spaccapietra, P. Atzeni, F. Fages, M.-S. Hacid, M. Kifer, J. Mylopoulos, B. Pernici, P. Shvaiko, J. Trujillo, I. Zaihrayeu (Eds.), Journal on Data Semantics VIII. XV, 219 pages. 2007.

Vol. 4378: I. Virbitskaite, A. Voronkov (Eds.), Perspectives of Systems Informatics. XIV, 496 pages. 2007.

Vol. 4377: M. Abe (Ed.), Topics in Cryptology – CT-RSA 2007. XI, 403 pages. 2006.

Vol. 4376: E. Frachtenberg, U. Schwiegelshohn (Eds.), Job Scheduling Strategies for Parallel Processing. VII, 257 pages. 2007.

Vol. 4374: J.F. Peters, A. Skowron, I. Düntsch, J. Grzymała-Busse, E. Orłowska, L. Polkowski (Eds.), Transactions on Rough Sets VI, Part I. XII, 499 pages. 2007.

Vol. 4373: K. Langendoen, T. Voigt (Eds.), Wireless Sensor Networks. XIII, 358 pages. 2007.

Vol. 4372: M. Kaufmann, D. Wagner (Eds.), Graph Drawing. XIV, 454 pages. 2007.

Vol. 4371: K. Inoue, K. Satoh, F. Toni (Eds.), Computational Logic in Multi-Agent Systems. X, 315 pages. 2007. (Sublibrary LNAI).

Vol. 4370: P.P Lévy, B. Le Grand, F. Poulet, M. Soto, L. Darago, L. Toubiana, J.-F. Vibert (Eds.), Pixelization Paradigm. XV, 279 pages. 2007.

Vol. 4369: M. Umeda, A. Wolf, O. Bartenstein, U. Geske, D. Seipel, O. Takata (Eds.), Declarative Programming for Knowledge Management. X, 229 pages. 2006. (Sublibrary LNAI).

Vol. 4368: T. Erlebach, C. Kaklamanis (Eds.), Approximation and Online Algorithms. X, 345 pages. 2007.

Vol. 4367: K. De Bosschere, D. Kaeli, P. Stenström, D. Whalley, T. Ungerer (Eds.), High Performance Embedded Architectures and Compilers. XI, 307 pages. 2007.

Vol. 4366: K. Tuyls, R. Westra, Y. Saeys, A. Nowé (Eds.), Knowledge Discovery and Emergent Complexity in Bioinformatics. IX, 183 pages. 2007. (Sublibrary LNBI).

Vol. 4364: T. Kühne (Ed.), Models in Software Engineering. XI, 332 pages. 2007.

Vol. 4362: J. van Leeuwen, G.F. Italiano, W. van der Hoek, C. Meinel, H. Sack, F. Plášil (Eds.), SOFSEM 2007: Theory and Practice of Computer Science. XXI, 937 pages. 2007.

Vol. 4361: H.J. Hoogeboom, G. Păun, G. Rozenberg, A. Salomaa (Eds.), Membrane Computing. IX, 555 pages. 2006.

Vol. 4360: W. Dubitzky, A. Schuster, P.M.A. Sloot, M. Schroeder, M. Romberg (Eds.), Distributed, High-Performance and Grid Computing in Computational Biology. X, 192 pages. 2007. (Sublibrary LNBI).

Vol. 4358: R. Vidal, A. Heyden, Y. Ma (Eds.), Dynamical Vision. IX, 329 pages. 2007.

Vol. 4357: L. Buttyán, V. Gligor, D. Westhoff (Eds.), Security and Privacy in Ad-Hoc and Sensor Networks. X, 193 pages. 2006.

Vol. 4355: J. Julliand, O. Kouchnarenko (Eds.), B 2007: Formal Specification and Development in B. XIII, 293 pages. 2006.

Vol. 4354: M. Hanus (Ed.), Practical Aspects of Declarative Languages. X, 335 pages. 2006.

Vol. 4353: T. Schwentick, D. Suciu (Eds.), Database Theory – ICDT 2007. XI, 419 pages. 2006.

Vol. 4352: T.-J. Cham, J. Cai, C. Dorai, D. Rajan, T.-S. Chua, L.-T. Chia (Eds.), Advances in Multimedia Modeling, Part II. XVIII, 743 pages. 2006.

Vol. 4351: T.-J. Cham, J. Cai, C. Dorai, D. Rajan, T.-S. Chua, L.-T. Chia (Eds.), Advances in Multimedia Modeling, Part I. XIX, 797 pages. 2006.

Vol. 4349: B. Cook, A. Podelski (Eds.), Verification, Model Checking, and Abstract Interpretation. XI, 395 pages. 2007.

Vol. 4348: S.T. Taft, R.A. Duff, R.L. Brukardt, E. Ploedereder, P. Leroy (Eds.), Ada 2005 Reference Manual. XXII, 765 pages. 2006.

Vol. 4347: J. Lopez (Ed.), Critical Information Infrastructures Security. X, 286 pages. 2006.

Vol. 4346: L. Brim, B. Haverkort, M. Leucker, J. van de Pol (Eds.), Formal Methods: Applications and Technology. X, 363 pages. 2007.

Vol. 4345: N. Maglaveras, I. Chouvarda, V. Koutkias, R. Brause (Eds.), Biological and Medical Data Analysis. XIII, 496 pages. 2006. (Sublibrary LNBI).

Vol. 4344: V. Gruhn, F. Oquendo (Eds.), Software Architecture. X, 245 pages. 2006.

Vol. 4342: H. de Swart, E. Orłowska, G. Schmidt, M. Roubens (Eds.), Theory and Applications of Relational Structures as Knowledge Instruments II. X, 373 pages. 2006. (Sublibrary LNAI).

Vol. 4341: P.Q. Nguyen (Ed.), Progress in Cryptology - VIETCRYPT 2006. XI, 385 pages. 2006.

Vol. 4340: R. Prodan, T. Fahringer, Grid Computing. XXIII, 317 pages. 2007.

Vol. 4339: E. Ayguadé, G. Baumgartner, J. Ramanujam, P. Sadayappan (Eds.), Languages and Compilers for Parallel Computing. XI, 476 pages. 2006.

Vol. 4338: P. Kalra, S. Peleg (Eds.), Computer Vision, Graphics and Image Processing. XV, 965 pages. 2006.

Vol. 4337: S. Arun-Kumar, N. Garg (Eds.), FSTTCS 2006: Foundations of Software Technology and Theoretical Computer Science. XIII, 430 pages. 2006.

Vol. 4336: V.R. Basili, D. Rombach, K. Schneider, B. Kitchenham, D. Pfahl, R. W. Selby (Eds.), Empirical Software Engineering Issues. XVII, 193 pages. 2007.

Vol. 4335: S.A. Brueckner, S. Hassas, M. Jelasity, D. Yamins (Eds.), Engineering Self-Organising Systems. XII, 212 pages. 2007. (Sublibrary LNAI).

Vol. 4334: B. Beckert, R. Hähnle, P.H. Schmitt (Eds.), Verification of Object-Oriented Software. XXIX, 658 pages. 2007. (Sublibrary LNAI).

Vol. 4333: U. Reimer, D. Karagiannis (Eds.), Practical Aspects of Knowledge Management. XII, 338 pages. 2006. (Sublibrary LNAI).

Vol. 4332: A. Bagchi, V. Atluri (Eds.), Information Systems Security. XV, 382 pages. 2006.

Vol. 4331: G. Min, B. Di Martino, L.T. Yang, M. Guo, G. Ruenger (Eds.), Frontiers of High Performance Computing and Networking – ISPA 2006 Workshops. XXXVII, 1141 pages. 2006.

Vol. 4330: M. Guo, L.T. Yang, B. Di Martino, H.P. Zima, J. Dongarra, F. Tang (Eds.), Parallel and Distributed Processing and Applications. XVIII, 953 pages. 2006.

Vol. 4329: R. Barua, T. Lange (Eds.), Progress in Cryptology - INDOCRYPT 2006. X, 454 pages. 2006.

Vol. 4328: D. Penkler, M. Reitenspiess, F. Tam (Eds.), Service Availability. X, 289 pages. 2006.

Vol. 4327: M. Baldoni, U. Endriss (Eds.), Declarative Agent Languages and Technologies IV. VIII, 257 pages. 2006. (Sublibrary LNAI).

Vol. 4326: S. Göbel, R. Malkewitz, I. Iurgel (Eds.), Technologies for Interactive Digital Storytelling and Entertainment. X, 384 pages. 2006.

Vol. 4325: J. Cao, I. Stojmenovic, X. Jia, S.K. Das (Eds.), Mobile Ad-hoc and Sensor Networks. XIX, 887 pages. 2006.

Vol. 4323: G. Doherty, A. Blandford (Eds.), Interactive Systems. XI, 269 pages. 2007.

Vol. 4322: F. Kordon, J. Sztipanovits (Eds.), Reliable Systems on Unreliable Networked Platforms. XIV, 317 pages. 2007.

Vol. 4320: R. Gotzhein, R. Reed (Eds.), System Analysis and Modeling: Language Profiles. X, 229 pages. 2006.

Vol. 4319: L.-W. Chang, W.-N. Lie (Eds.), Advances in Image and Video Technology. XXVI, 1347 pages. 2006.

Vol. 4318: H. Lipmaa, M. Yung, D. Lin (Eds.), Information Security and Cryptology. XI, 305 pages. 2006.

Vol. 4317: S.K. Madria, K.T. Claypool, R. Kannan, P. Uppuluri, M.M. Gore (Eds.), Distributed Computing and Internet Technology. XIX, 466 pages. 2006.

Vol. 4316: M.M. Dalkilic, S. Kim, J. Yang (Eds.), Data Mining and Bioinformatics. VIII, 197 pages. 2006. (Sublibrary LNBI).

Vol. 4314: C. Freksa, M. Kohlhase, K. Schill (Eds.), KI 2006: Advances in Artificial Intelligence. XII, 458 pages. 2007. (Sublibrary LNAI).

Vol. 4313: T. Margaria, B. Steffen (Eds.), Leveraging Applications of Formal Methods. IX, 197 pages. 2006.

Vol. 4312: S. Sugimoto, J. Hunter, A. Rauber, A. Morishima (Eds.), Digital Libraries: Achievements, Challenges and Opportunities. XVIII, 571 pages. 2006.

Vol. 4311: K. Cho, P. Jacquet (Eds.), Technologies for Advanced Heterogeneous Networks II. XI, 253 pages. 2006.

Table of Contents

Session 1 *The Empirical Paradigm*

The Empirical Paradigm *Introduction* 1
Dieter Rombach

Approaches for Empirical Validation

Techniques for Empirical Validation 4
Marvin V. Zelkowitz

Status of Empirical Research in Software Engineering 10
Andreas Höfer and Walter F. Tichy

Position Papers

Aggregation of Empirical Evidence 20
Marcus Ciolkowski

Empirical Evaluation in Software Engineering: Role, Strategy, and Limitations 21
Lionel C. Briand

New Opportunities for Empirical Research 22
Markku Oivo

Empirical Paradigm: Position Paper 23
Carolyn B. Seaman

The Value of Empirical Evidence for Practitioners and Researchers 24
Austen Rainer

Exploration Versus Confirmation

Empirical Paradigm – The Role of Experiments 25
Barbara Kitchenham

The Role of Controlled Experiments in Software Engineering Research ... 33
Victor R. Basili

Position Papers

Creating Real Value in Software Engineering Experiments 38
James Miller

From Silver Bullets to Philosophers' Stones: Who Wants to Be Just an Empiricist? 39
Guilherme H. Travassos

Social and Human Aspects of Software Engineering 40
Helen Sharp

Longitudinal Studies in Evidence-Based Software Engineering 41
Tracy Hall

The Use of Grounded Theory in Empirical Software Engineering 42
Jeffrey Carver

Historical Review

Exploration and Confirmation: An Historical Perspective 43
Michael S. Mahoney

Working Group Results

Combining Study Designs and Techniques *Working Group Results* 50
Carolyn B. Seaman

Optimizing Return-On-Investment (ROI) for Empirical Software Engineering Studies *Working Group Results* 54
Lutz Prechelt

The Role of Controlled Experiments *Working Group Results* 58
Andreas Jedlitschka and Lionel C. Briand

Discussion and Summary

The Empirical Paradigm *Discussion and Summary* 63
Marcus Ciolkowski, Barbara Kitchenham, and Dieter Rombach

Session 2 *Measurement and Model Building*

Measurement and Model Building *Introduction* 68
Victor R. Basili

Data Sharing

Data Collection, Analysis, and Sharing Strategies for Enabling Software Measurement and Model Building 70
Richard W. Selby

Knowledge Acquisition in Software Engineering Requires Sharing of Data and Artifacts 77
Dag I.K. Sjøberg

Effective Data Interpretation

Effective Data Interpretation 83
Jürgen Münch

Software Support Tools and Experimental Work 91
Audris Mockus

Position Papers

Measurement and Interpretation of Productivity and Functional Correctness 100
Hakan Erdogmus

Synthesising Research Results 101
Barbara Kitchenham

On the Quality of Data 102
Thomas Ostrand

Working Group Results

Potential of Open Source Systems as Project Repositories for Empirical Studies *Working Group Results* 103
Nachiappan Nagappan

Data Sharing Enabling Technologies *Working Group Results* 108
Marvin V. Zelkowitz

Documenting Theories *Working Group Results* 111
Dag I.K. Sjøberg

Discussion and Summary

Measurement and Model Building *Discussion and Summary* 115
Sira Vegas and Vic Basili

Session 3 *Technology Transfer and Education*

Technology Transfer and Education *Introduction* 121
Kurt Schneider

Technology Transfer

Empirical Studies as a Basis for Technology Transfer 125
Elaine J. Weyuker

Position Papers

Relationships and Responsibilities of Software Experimentation 128
Giovanni Cantone

The (Practical) Importance of SE Experiments 129
Tore Dybå

How to Improve the Use of Controlled Experiments as a Means for Early Technology Transfer 130
Andreas Jedlitschka

Extending Empirical Studies to Cover More Realistic Industrial Development and Project Management Issues 131
Marek Leszak

Empirical Case Studies in Industry: Some Thoughts 132
Nachiappan Nagappan

Software Process Simulation Frameworks in Support of Packaging and Transferring Empirical Evidence 133
Dietmar Pfahl

Structuring Families of Industrial Case Studies 134
Laurie Williams

Education

Empirical Software Engineering: Teaching Methods and Conducting Studies 135
Claes Wohlin

Educational Objectives for Empirical Methods 143
Natalia Juristo

Position Papers

On "Landscaping" and Influence of Empirical Studies 151
Frank Houdek

Involving Industry Professionals in Empirical Studies with Students 152
Letizia Jaccheri and Sandro Morasca

Working Group Results

Industry-Research Collaboration *Working Group Results* 153
Lutz Prechelt and Laurie Williams

Teaching Empirical Methods to Undergraduate Students *Working Group Results* 158
Austen Rainer, Marcus Ciolkowski, Dietmar Pfahl, Barbara Kitchenham, Sandro Morasca, Matthias M. Müller, Guilherme H. Travassos, and Sira Vegas