Fraunhofer Institut
Experimentelles
Software Engineering

On the Use of Empirical Studies in Software Engineering

Authors:
Andreas Jedlitschka
Christian Bunse

Published in
Workshop Series on Empirical
Software Engineering
Vol 1

IESE-Report No. 054.03/E
Version 1.0
August, 2003

A publication by Fraunhofer IESE

Fraunhofer IESE is an institute of the Fraunhofer Gesellschaft.
The institute transfers innovative software development techniques, methods and tools into industrial practice, assists companies in building software competencies customized to their needs, and helps them to establish a competitive market position.

Fraunhofer IESE is directed by
Prof. Dr. Dieter Rombach
Sauerwiesen 6
67661 Kaiserslautern

Abstract

Software process improvement (SPI) is instantiated in industry in many ways. Different models exist like CMM, CMMI, SPICE, ISO9000. All of these require experimentation and cycles for improvement. One approach that is used to achieve this is the use of empirical methods in conjunction with, for example, the Quality Improvement Paradigm (QIP). The EU-funded Experimental Software Engineering Network aims at supporting improvement maturity by providing an infrastructure for cross-organisational exchange by the means of empirical studies and the sharing of experience both on a personal and technical level. In order to foster the understanding of the use of empirical studies for organisation specific improvement management activities, the authors present a generic framework for strategic improvement management enabling the use of existing resources and the integration of these into an improvement cycle. The workshop on "Empirical Studies in Software Engineering" contributes to the common body of knowledge necessary to support decisions pro/contra (e.g., the introduction of a new technology), and is similar to the use of pair programming within established software development processes.

Table of Contents

Introduction

Software Process Improvement (SPI) plays a vital role in software engineering. The preconditions for running improvement programmes are established measurement of key indicators, alongside a method enabling conclusions to be drawn from the figures on the one hand, but also to feed them back into new projects on the other hand. One instantiation for this approach is the use of empirical methods to evaluate products, processes, people, and the Quality Improvement Paradigm (QIP) [BCR01a] as improvement cycle.

The EU-funded Experimental Software Engineering Network (ESERNET) assigns high priority to fostering the use of empirical studies [CW03] in order to remain ahead of developments in Software Engineering (SE). ESERNET itself includes members from both industry and research and provides measures to strengthen exchange of knowledge such as: exchange of personnel, cross-organisational web-based knowledge management infrastructure [JN03]; and, workshops, and tutorials. In order to disseminate the most recent findings from both research and industry across the borders of the project itself, ESERNET established the Workshop Series in Empirical Software Engineering.

One lesson from ESERNET is that it is important to have a common understanding of the use of empirical studies in the context of SPI. In reality, there are various SPI approaches from different models (e.g., CMM, CMMI, SPICE, ISO 9000) used in industry. The most common denominator of these approaches is a cycle for improvement based on measurement together with what can be loosely labelled, experimentation. Drawing upon some main lessons (for details see [JP03]), a generic framework for Strategic Improvement Management (SIM) is presented, this aims to foster a common understanding of SPI in conjunction with the use of empirical studies, as well as the use of exchange of experience across the borders of single organisations.

The remainder of the chapter is structured as follows. First the ESERNET project is briefly described, then the generic SIM framework is presented, and the papers from the workshop are summarized.

The Experimental Software Engineering Network (ESERNET)

ESERNET aims at supporting improvement maturity and competitiveness of European software intensive organisations. In order to improve, there is a need to efficiently share experience about best SE practices, their benefits as well as their context requirements and boundaries. Such knowledge enables companies to improve faster and at lower cost. ESERNET collects such knowledge, on the one hand knowledge that exists locally in individual companies (derived via case studies in projects), and on the other from research organisations (derived via controlled technology empirical studies).

In the context of inspection technology, a typical example of an empirical study might look as follows. A post-mortem analysis is first carried out on inspection techniques (using both literature and ESERNET industrial data). Initial studies with university students are then used to test the conclusions and hypotheses of the post-mortem analysis. The results are used in designing a replicated controlled empirical study with multiple groups of subjects in three universities in different countries resulting in about ten parallel studies. The outcomes of these studies are carefully analysed and used to draw the first conclusions on the validity of inspection techniques. A company applying inspection techniques can use these results to systematically introduce, or improve inspections techniques in an industrial environment. The results of the industrial study are packaged with context information and can then be used in other industrial studies. These new empirical studies are used to validate the first results and to explore how the results apply in different industrial environments. An essential element in each of these different types of empirical studies is packaging the results sharing them via web-based knowledge repositories (e.g., ESERNET: www.esernet.org).

One other lesson we learned from ESERNET is that, from an industrial point of view, research is often to far away from real industrial needs, e.g., by investing lots of effort in slightly "improved" methods with little impact instead of searching to prove evidence for existing methods. This somehow reflects the needs of industry for guidelines and templates to support the selection and introduction of technologies or improvement measures. The existing results from empirical studies are often far away to allow the collation and aggregation of more general guidance or to proof of evidence.

One reason for this may be that in comparison to other types of research (e.g., medical research), there are no common guidelines available for describing minimum standards of how to perform, report, and collate results of empirical studies in SE in a manner that allows to prove evidence. This would also include guidelines regarding how to describe the results and lessons learned in a way that can be understood by industry. From the authors experience additional research is necessary to understand what context really means, and what are the least common denominators to allow comparisons and common conclusions.

ESERNET aims at fostering the collaboration between research and industry, but the authors claim that this is not enough. To improve both, research and software development in industry it is necessary to close existing gaps by the means of fostering a common understanding of the needs of both sides. In the first case this might be obtained if research contributes with more relevant, reliable and evidence-based results. But this can only be achieved if industry is willing to provide information about their current and future needs, and also supports research in collecting evidence by the means of providing test beds or at least data from pilot projects. The positive impact for both sides is obvious. By having more influence on research activities industry will directly benefit through improved and usable results, and indirectly by potentially influencing education in the field. Research will benefit through engaging feedback and improved possibilities to "sell" their products and services.

The authors believe that independent research is absolutely necessary to develop and evolve new methodologies, technologies, etc. However, whilst this is necessary, it is not sufficient, there is also a need for applied research that closes the gap between industry and basic research. Here the authors claim that to prove evidence and to provide results that are useful for industry it is absolutely necessary to have studies in real industrial environments.

Additionally, these lessons confirm, and reiterate the findings of previous researchers in software engineering, e.g., Sommerville and Rodden [SR95]. There is a need to a more interdisciplinary approach, so to involve (organizational) psychologists, operation research, and education; firstly, to find solutions to overcome human and social barriers or at least take them into account, and secondly to better understand potential financial and structural impacts on the organisation. The first step in the direction sketched above will be described in the following section by introducing a framework for Strategic Improvement Management (SIM) that aims at fostering a common understanding of the need for, and the use of empirical studies in Software Process Improvement (SPI). In addition, this framework shows how empirical studies contribute to organizational learning and, in the future to cross-organizational learning.

Strategic Improvement Management (SIM)

It is widely accepted today that software development has to be performed in a more or less systematic and managed way. This acceptance does not necessarily hold for improvement initiatives. They often fail, i.e., do not yield the expected results (level of improvement or benefit), because they are:

- performed in an isolated, non-coordinated way,
- viewed outside their initial context; or,
- not supported by management or individuals.

Those were the reasons that prompted Birk [BR01] to propose a framework supporting the systematic management of improvement measures.

Empirical SE somehow faces similar challenges. Measurement and documentation are often seen as an extra effort, since they are not regarded as integral part of the daily work and are correspondingly cut back by management (e.g., in times of low markets). The framework described here aims at motivating empirical studies in SE on the one hand, and the extensive use of and need for empirical studies for systematic improvement management on the other hand. The framework covers other improvement approaches like assessments, measurement programmes, and competence management. It also incorporates the industrial view on improvement driven by the need for fast ROI for improvements. This is achieved by means of an integrated cost-benefit analysis approach.

Once instantiated and populated by the organizational key figures and nominal values, it will also support decision-making for strategic improvements considering the situation in hand, the goals, and the experience of the organization. In addition external data, for example, that obtainable from empirical studies or experts, can be consulted. This requires at least minimum information about the context in which the study has been performed but also a standardized reporting structure, that enables the reader fast access to the main findings (quantitative data) but also to the main lessons learned (qualitative data). This is especially important for managers. ESERNET has developed such a common reporting structure, which is public available via the ESERNET portal.

Empirical studies themselves are, from an industrial point of view, often seen as kind of basic research. On the other hand, if a software development project fails, lots of efforts are spent in post-mortem-analyses, which are in fact one type of empirical study (i.e., in a reactive way). The most crucial point is to draw the right conclusions from the post-mortem, to make a plan to avoid the same faults in future, and to follow that plan. With some additions this leads to a more active or controlled (managed) way of improvement (for the proposed framework see figure 1). Additionally, many efforts are undertaken to control the projects via huge collections of figures. Using the results in a more analytical (research-based) way together with some small additions will lead to industrial case studies. These provide a firm initial basis with which to provide more systematic evidence as a whole, perhaps at a later stage.

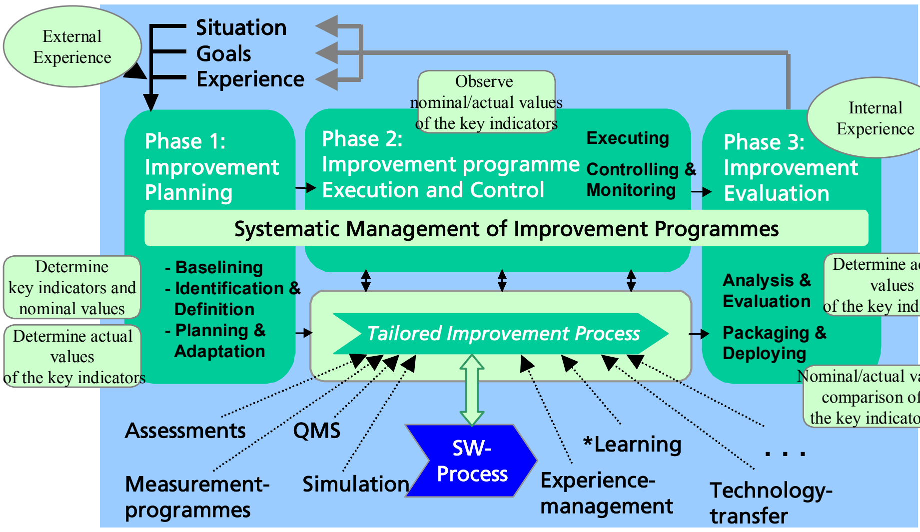

Figure 1: The Strategic Improvement Management Framework (SIM)

In figure 1 the connection between different types of experience organizational context, systematic planning, execution, and evaluation of improvement measures and potential measures is sketched. Additionally, figure 1 depicts the flow of continuous feedback. Considering the goals, the situation at hand and the experience of the organization, e.g., with certain technologies, in phase 1 (Improvement Planning) a base-lining exercise should be performed. Additionally, external experience, e.g., taken from empirical studies or even better from evidence reports, or consultation of external experts, can be taken into account in order to define a baseline. Based on the results the key indicators and their nominal values together with potentials towards the goals for improvement can be determined (e.g., by using the Goal-Question-Metric (GQM) approach [BCR01b]). An a-priori cost/benefit analysis (based on available empirical, accurate data and estimation) should be performed as well. Especially for industry, a systematic approach to cost/benefit analysis is crucial for making decisions according to, and in line with, the business goals of the company. The framework supports decision-making using techniques for constructing adequate key figures satisfying business needs and transparent decision-models. At this stage it is often necessary to choose between the provided alternatives. Apart from experts in the field it is almost impossible for managers, and practitioners in general, to know all the potential measures or the most suitable ones. The SIM framework in a final stage aims at supporting decision making by providing relevant information and assigning a ranking to the best fitting ones (e.g., case-based reasoning has shown some evidence in supporting similar approaches in other fields like product suggestion in the context of user modelling). The solution also attempts to use the organizational context (for ease of understanding, here context means (a) information about the organization like domain and country, (b) the goals of the organization, and (c) the internal experience) to find the most relevant external experience (e.g., about the introduction of a new technique in the same domain, same country, similar experience as study baseline). The results are also ranked taking into account cost-benefit prerequisites.

Since the framework will only be able to support the decision process by proposing possible measures, the last choice is left to the managers. In future guidelines and templates to additionally support the process of decision-making will be available.

In order to clarify the approach, the authors sketch the following example, which demonstrates the “similarities” that exist with the discipline of dentistry where for one treatment (goal) different alternatives are possibly available. However, but the last final choice (e.g., if one takes dental gold, amalgam or ceramics – i.e., different measures), is up to the patient and will be met based on the dentist’s (expert) advise concerning the advantages and disadvantages of the different measures, cost-benefit, and other more considerations (e.g., cosmetic reasons).

After having selected a measure this has to be adapted, with respect to the organizational speciality. Proper planning and adaptation of the improvement measure to the needs of the organization is the last but not least important step of phase 1. In phase 2 (Improvement program execution and control) the tailored improvement process is continuously observed and measured against the plans by comparing the nominal with the actual values of the key figures. This enables active controlling and risk management by the means of internal feedback. The improvement process influences the software development process after its successful evaluation in phase 3 (Improvement evaluation). The whole improvement measure is analysed and evaluated by determining the actual values of the cost/benefit key figures again. If they are in the expected range and the a-posteriori cost/benefit analysis is also successful the improvement measure can be fixed in the processes of the organization. In both cases, successful, but also unsuccessful evaluation packaging is essential to foster the organizational experience and to further improve future strategic improvement activities. This also contributes to the body of knowledge by providing evidence for at least the organization, but in the case of publication, also allows more generalisable conclusions to be drawn.

Summary and Conclusions

The goal of the workshop 'Empirical Studies in Software Engineering' has been to explore the use of empirical studies in research practice (e.g., practical experience and studies) and to discuss the key issues involved (e.g., problems concerned with justifying the practical use of empirical assessment and evaluation in software engineering). It has been a unique forum dedicated to the presentation and discussion of research and practical experiences addressing all aspects of empirical assessment and evaluation in software engineering. Several practitioners and researchers contributed to the workshop and helped to enable an exchange of experience. The papers submitted to this workshop show that although the concepts and ideas behind empirical studies are becoming widely accepted, there are still many issues to be resolved and studies to be performed.

Within the workshop there have been a number of papers which reported on actual empirical studies on different software development techniques and processes. The paper by *Ciolkowski* and *Schlemmer* reports an experiment on 'Pair Programming', which resulted in weak support that 'Pair Programming' results in higher-quality products but also requires more effort. The paper by *Abrahamsson et. al.* describes an experiment on the 'Personal Software Process' (PSP). Results showed that using PSP did not had an impact on size- and time estimation skills as well as on productivity but resulted in a higher product quality. The paper by *Serrano et al* describes a study on data warehouse design methods (i.e., the STAR and traditional approaches). It concludes that the experiment did not show any difference in the comprehensibility of schemas designed with either method. However, first trends seem to be promising and will be further examined by replications of the study. The paper by *Punter* discussed the rationale of a goal-oriented method for specifying user perceived software quality and experiences from its use. The paper concludes that domain experts can define factors that determine the quality and importance of these factors by using the proposed method. The paper by *Oliver* discusses ideas and trends in modelling software processes with UML-SPEM. Furthermore, it presents some experience in using UML-SPEM in real-world projects and identifies some issues which have to be (empirically) evaluated in the future. The paper by *Westerheim et. al.* presents a research design to evaluate the extension of lightweight post-mortem analysis in order to enable comparison between different analyses. The paper by *Del Rosso* and *Maccari* presents an industrial validation (i.e., case study) of software architecture assessment methods. The results showed that the applied interview-based approach showed promising results and was been perceived as useful by the practitioners. The paper by *Russo et. al*. introduces a procedure for the analysis of software modification requests to help managers in better estimation and allocation of resources, which is then evaluated by a case-study. Results show that 'S-shaped reliability models' are superior in predicting modification requests than 'Concave models'

There have also been a number of submissions devoted to the more fundamental or theoretical aspects of empirical software engineering. The paper *by Bhushan* and *Kaushik* discusses the transfer and use of the 'Entropy' Measure, developed for thermo-dynamic systems in the context of software development to measure disorder. In addition, it compares the measure with SEI standard measures and concludes that the 'Entropy' measure is superior. The paper by *Bevier* describes a method for evaluating the specifications of small database systems. It argues that the introduced ML-Method enables 'Power Users' to specify application better suited toward s the problem of service and maintenance. However, this hypothesis has to be evaluated by some form of empirical study.

In addition to the papers and their presentations in the first half of the workshop, the afternoon session was devoted to discussion and practical exchange of experience whereby discussions focused on the following topics:

- Advanced statistical and knowledge engineering techniques for analysing the results of the experiments.
- Meta-Analysis.
- Students vs. Professionals as experimental subjects
- How to package, store, and exchange empirical experience (i.e., do we need knowledge repositories)?
- How should experience be captured and stored (level of abstraction, granularity, technical details, etc.)? Technology experiences are often within one type of domain experts: engineers. Conflicting opinions of domain experts might exist (e.g., between software architects and engineers). What about integrating those different views?

The major conclusion of this discussion was that empirical studies are a powerful tool to evaluate SE techniques and practices. They are especially useful to examine of new techniques (e.g., agile development) are indeed as beneficial as promised. However, many problems still remain. In order to increase the awareness, to enable cross-company learning and to avoid replication a forum is needed which does not only provide knowledge on how to conduct empirical studies (i.e., the 'Tools') but also stores such studies to make them internationally available (One candidate being the ESERNET repository (http://www.esernet.org).

Acknowledgements

The work presented has partly been funded by the Information Societies Technology Programme of the European Union under grant IST-2000-26754. The complete list of partners and members can be found at www.esernet.org.

Special thanks go to Michael Ochs, Patrick Waterson and Dietmar Pfahl for many fruitful discussions.

References

[BCR01a] Basili, V.R.; Caldiera, G.; Rombach, H.D., Experience Factory; in: Marciniak JJ (ed.), Encyclopedia of Software Engineering, Vol. 1, pp. 511-519, John Wiley & Sons, 2001.

[BCR01b] Basili, V.R., Caldiera, G., and Rombach, H.D.; Goal Question Metric Paradigm; in: Marciniak JJ (ed.), Encyclopedia of Software Engineering, Vol. 1, pp. 528–532, John Wiley & Sons, 2001.

[BR01] Birk, A. & Rombach, D.: A Practical Approach to Continuous Improvement in Software Engineering; in Wieczorek, M. Meyerhoff, D. (Eds.) Software Quality, State of the art in Management, Testing, and Tools; Springer , 2001 ISBN 3-540-41441-X

[CW03] Conradi, R.; Wang, A.I.; *Empirical Methods and Studies in Software Engineering – Experiences from ESERNET*; Springer LNCS 2765, 2003

[JN03] Jedlitschka, A. and Nick, M.; Software Engineering Knowledge Repositories in [CW03]

[JP03] Jedlitschka, A. & Pfahl, D.: Experience-Based Model-Driven Improvement Management with Combined Data Sources from Industry and Academia, accepted for ACM/IEEE International Symposium on Empirical Software Engineering 2003.

[SR95] Sommerville, I. and Rodden, T.: 'Human, Social and Organisational Influences on the Software Process', in Trends in Software Processes, A. Fuggetta and A. Wolf (Eds). 1995, John Wiley and Sons: New York.

Document Information

Title:	On the Use of Empirical Studies in Software Engineering
Date:	August 2003
Report:	IESE-054.03/E
Status:	Final
Distribution:	Public