Carl von Ossietzky University of Oldenburg
DFG Graduate School "TrustSoft"
http://trustsoft.uni-oldenburg.de
26111 Oldenburg
Germany

Seminar "Research Methods", Summer Term 2005

The Role of Experimentation in Software Engineering

Heiko Koziolek

7th July 2005

Abstract

Research proposals need to be validated either by formal proofs or by applying empirical methods (e.g. controlled experiments). Many authors have pointed out that the level of experimentation in software engineering is not satisfactory. The quantity of experimentation is too low as a lot of software engineering publication do not contain any empirical validation at all. But also the quality of software engineering experiments conducted so far is often weak, because no proper methodological approach is applied and statistical methods are misused.

This paper provides an overview of the status of experimentation in software engineering. First, the role of experimentation among other types of research is clarified. Several research paradigms are introduced, a classification of different types of experiments in software engineering is provided, and a comparison with experimentation in other research disciplines is drawn. Afterwards the current state of experimentation in software engineering is analysed with more detail. Some discussion points from various researchers about the situation of experimentation are summed up. Their recommendations for improving the state of experimentation are provided as well as possible future directions of experimentation in software engineering.

1 Introduction

Any research proposal in computer science needs to be validated properly to check it claims, improvements and also its applicability in practice. To conduct such a validation either the proposal has to be proven formally or empirical methods have to be used to gather evidence. The fact that formal proofs are only seldom possible in software engineering (SE), has lead researchers to emphasize the role of empirical methods, which are traditionally more common in disciplines like physics, social sciences, medicine or psychology.

Several different empirical methods are known. Quantitative methods try to measure a certain effect, while qualitative methods search for the reasons of an observed effect. Examples for quantitative methods are experiments, case studies, field studies, surveys and meta-studies. Examples for qualitative methods are interviews and group discussions.

The *controlled experiment* is the method with the highest degree of confidence into the results. In such an experiment researchers try to control any variable influencing the outcome, except the variable they want to analyse. For example, this involves testing a product or method with a larger group of persons, so that the differences in the qualifications of the participants can be reduced by averaging and thus do not influence the result. A less strict method is the *case study*, in which a (possibly artificial) example is analysed and initial interpretations of the observed effects can be drawn, but the results are normally not generalizable to other examples. *Surveys* include searching the literature or passing out questionnaires to experts to gather evidence. *Meta-studies* analyse other studies and try to gain knowledge by comparing different approaches. A more detailed introduction into empirical methods in computer sciences can be found in several books, which have appeared recently [WRH+00, Pre01, JM01].

Many authors have pointed out that the level of experimentation in SE is not satisfactory. The quantity of experimentation is weak, as a lot of researchers are reluctant to validate their approaches empirically. The quality of experimentation is also often not sufficient, as statistical methods are used inappropriately or it is neglected to draw proper conclusions from experiments. Based on these observations this paper discusses the role of experimentation in SE in more detail.

This paper is organised as follows: The following section 2 contains an overview of general research approaches and describes how research is conducted in other disciplines. Afterwards, the focus shifts specifically to experimentation and its application in SE. Section 3 analyses the status quo of experimentation in SE, while section 4 sums up several common fallacies on this topic. Section 5 summarizes future direction of experimentation and presents some ideas how to improve the conduction of empirical studies. Section 6 contains a critical reflection of some of the statements found in the literature analysed before. Finally, Section 7 concludes the paper.

2 Research and Experimentation in SE

Evolution in science disciplines is based on encapsulating experience into models and their verification and validation based on experimentation [Bas96]. The science process is a cycle of applying ideas, analysing results and collection feedback. SE requires the same cycle of model building, experimentation and learning to become a matured scientific discipline.

Scientific research in SE is conducted under the assumption that "if we look long enough and hard enough, we will find rational rules that show us the best ways to build the best software" [Pfl99].

In the following, first different levels of knowledge are presented before four general research methods and the experimentation/learning cycle are introduced. To conduct empirical research, multiple data collection methods can be applied, which are summed up in the third subsection. Older scientific disciplines like physics and psychology have developed their own experimental

paradigms, which are shortly described in the fourth subsection. This section concludes with an argumentation why SE is different from other scientific disciplines and why an own experimental paradigm has to be developed.

2.1 Empirical Knowledge versus Theoretical Knowledge

Three levels of scientific investigations to gain knowledge can be identified [JM01]:

- **Survey inquiries** observe an object or process and try to find out, which variables affect other variables. The effect is neither quantified nor explained. Although other disciplines have identified the relationships between important variables, most of the influencing variables in SE are still not known.
- **Empirical inquiries** aim at quantifying how variables affect other variables. The goal of such inquiries is to construct an empirical model. According to Popper [Pop59] empirical inquiries cannot prove any theory, they can only fail to falsify it. From this viewpoint scientific knowledge is nothing more than a system of untrue statements and claims that are provisionally true as long as they are not contradicted. This insight should always be kept in mind when reasoning about experimentation. Apart from the inability to prove a theory, empirical inquiries are also not able to create an understanding of the observed phenomena.
- **Mechanistic inquiries** represent the highest level of scientific investigation. The aim is to explain why variables affect other variables in the observed manner and to construct a theoretical model. Such a theoretical model contributes to the understanding of a phenomena and additionally provides a basis for extrapolation, which is not possible with an empirical model. Thus, predictions can be made about certain phenomena based on an theoretical model. Theoretical models also provide a stricter representation of the response function to a research question.

As survey inquiries have hardly been conducted in SE and not enough background knowledge has been collected, it is not yet possible to construct theoretical models in SE. Empirical models are a necessary step towards theoretical models for SE.

2.2 Research Paradigms and Methods

Basili distinguishes between two main research paradigms [Bas93]. The experimental paradigm includes the scientific method with the engineering method and the empirical method as subsets, while the analytical paradigm includes the mathematical method:

- **Scientific method:** "Observe the world, propose a model or a theory of behaviour, measure and analyse, validate hypotheses of the model or theory, and if possible repeat the procedure."
  - **Engineering method:** "Observe existing solutions, propose better solutions, build or develop, measure and analyse, and repeat the process until no more improvements appear possible."
  - **Empirical method:** "Propose a model, develop statistical/qualitative methods, apply to case studies, measure and analyse, validate model and repeat the procedure."
- **Mathematical method:** "Propose a formal theory or set of axioms, develop a theory, derive results and if possible compare with empirical observations."

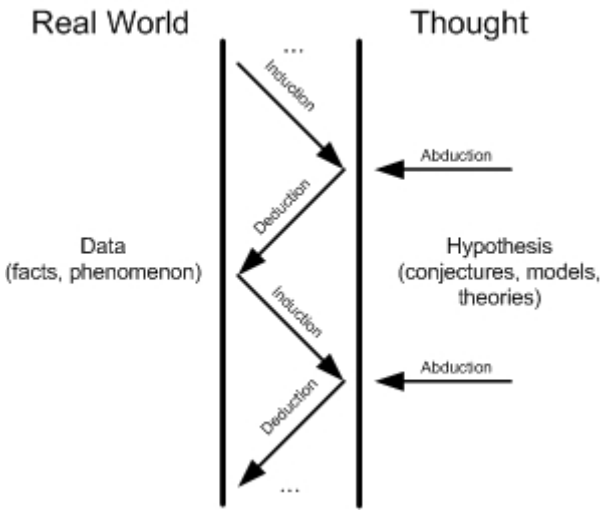

Figure 1: Iterative learning cycle [JM01]

The experimental paradigm is an *inductive* approach: it uses empirical evidence to formulate generalised knowledge from the observations made. On the other hand the analytical paradigm is *deductive*: it takes known theories and provides new evidence for special cases. In deductive reasoning evidence provided must be a set about which everything is known before the conclusion can be drawn. Since not all variables of SE (software artefacts, development processes, people, environment) are known formally, it is difficult to fully apply such an approach in this discipline. Thus, Basili concludes that "the inductive paradigm might be best used when trying to understand the software process, product, people, environment".

According to Glass [Gla94] the engineering, empirical and also the mathematical method are applicable in computer science, however the scientific method (although a superset of the first two methods) is problematic because of the phrase "observe the world". Software is intangible and has an invisible nature, so the phrase should be replaced by "observe the problem space".

To clarify the relationship between deduction and induction, Juristo et. al. [JM01] illustrate the iterative *experimentation/learning cycle* (Figure 1). Deduction proves something, induction shows that something is operational and abduction suggests that something could be. The cycle is composed as follows: a preliminary hypothesis is made about some phenomena. A process of deduction from the hypothesis leads to results, which are compared against real data. Discrepancies about the deduction results and reality can lead to a new hypothesis via induction. With the new hypothesis the cycle starts again.

The two variations of the experimental paradigm are rather contrary [Bas93]. The engineering method is *evolutionary*, it takes existing solutions and tries to improve them. For example the knowledge that a method is less cost-intensive than another or that a new tool preforms better than its predecessors might be the result of such an approach. Opposed to that, the empirical method can be a *revolutionary* approach not based on existing solutions. For example the proposal of a new method or model and the evaluation of its effect to the software development process or software products is a result of this approach. For both approaches careful analysis and measurements are critical for their success.

With these paradigms research activities and development can be differentiated. If someone is neither applying the experimental nor analytical paradigm, his work cannot be considered as research. For example, constructing a system or tool alone without validation is development and not research. Only if an understanding is gained how a system works or why it is useful and this is validated by experimentation, then a work qualifies as being research.

In his critique about the state of the art of computer science research Glass [Gla94] sums up some observations about the research methods introduced above. Few studies in fact have used the scientific method. Unfortunately, it is rather uncommon for computer scientists to formulate and validate hypotheses and to do evaluation in an iterative cycle. The engineering method has been used more frequently, as sometimes proposals aim to improve existing solutions. Yet the cycle "repeat, until no further improvement is possible" can hardly be found in most approaches. The empirical method is under-represented and only used by a few researchers. Glass criticizes that most researchers have no interest in this approach, and if they do, they tend to use only student experiments, not industrial case studies. The mathematical method is still the most common in computer science because of the mathematical heritage of the discipline. Within this method it is extremely rare that researchers compare their results with empirical observations. Glass concludes to characterize the most common research method as seriously flawed in practice.

2.3 Data Collection Methods

Data collection from reality for the experimental/learning cycle can be done in multiple ways. A taxonomy by Zelkowitz et. al. lists 12 different data collection methods:

- **Observational Methods:** project monitoring, case study, assertion, field study
- **Historical Methods:** literature search, legacy data, lessons-learned, static analysis
- **Controlled Methods:** replicated experiment, synthetic environment experiment, dynamic analysis, simulation

An overview of the empirical methods most common in computer sciences and SE (controlled experiments, case studies, survey, meta-studies) can be found in [Pre01, WRH$^+$00].

Basili provides a classification for several kinds of experiments [Bas96], which are identified by:

- Type of results
  - Descriptive: relationships among variables have not been examined
  - Correlational: variation of dependent variables is related to variation of independent variables
  - Cause-effect: independent variable is only possible cause for variation of dependent variables
- Type of participants
  - Novice: inexperienced students
  - Expert: practitioners experienced in the study domain
- Type of environment
  - In vivo: in the field (i.e. the software industry) under realistic conditions
  - In vitro: in the laboratory (i.e. the university) under controlled conditions
- Level of control
  - Controlled experiment: typically in vitro, mostly with students, strong statistically confidence in results, expensive, difficult to control
  - Quasi-experiment: typically in vivo, with practitioners, qualitative character

– Observational study: no treatment or controlled variables, possibly no set of study variables defined beforehand

Furthermore Basili examines, that in general two patterns of experiments have been performed so far: human factor studies and project-based studies. *Human factor studies* are normally cause-effect studies and try to find out how software engineers perceive and solve problems. *Project-based studies* are often correlational or descriptive studies and aim to help practitioners building models of software product and processes.

Still, there is a disagreement about the right methodology in empirical SE, as discussed in a panel session on the ICSE 2003 and summed up in a position paper by Walker [WBN+03]. Tichy favours controlled, quantitative, and statistically-analysable experiments, but like Popper he also states that no "silver-bullet experiments" exists, which could provide a final answer to a research question. Kitchenham also favours quantitative studies, but highlights the value of field studies in an industrial environment. For Seaman, qualitative studies are a valuable addition to quantitative studies. On one hand they can be used to replicate a study with a different methodology, thereby improving the trust in the results. On the other hand qualitative studies can help reducing the complexity of SE experiment resulting from the human factor, because they were originally developed to analyse human behaviour. Murphy (also in [MWB99]) points out that different evaluation treatments have to be applied according the degree of maturity of a technology under analysis. Briand adds that the discipline of SE needs to develop its own body of experience and strategies like other disciplines. Additionally, he states that qualitative and quantitative methods actually complement each other. Furthermore, in his opinion controlled experiments (with often high internal, but low external validity) and field studies (with often low internal but high external validity) both are necessary to create a body of evidence. Notkin states that empirical evidence is not always needed to transfer research results into practice. Because of the uniquely high rate of change in SE, the application of research results must be arranged differently. Although not each minor solution to a problem needs to be evaluated empirically, for deeper solutions this is necessary ("heavy claims require heavy evaluation").

2.4 Research in Other Fields

The science approach of the early Greeks was to observe something and then create knowledge by a chain of logical thought. Experimentation was hardly used by the Greeks for scientific purposes. One of the first popular scientific experiment in science history was performed by Galileo Galilei, who allegedly dropped balls from the tower of Pisa to study the free fall. Over the course of history each research field has developed experimental paradigms. Basili [Bas96] argues that SE needs to follow this model of other physical sciences.

- **Physics:** In physics researchers are usually either theorists or experimentalists. Models are built by theorists to explain the universe, while experimentalists observe and measure the environment with the goal of validating a theory or exploring new areas.
- **Medicine:** Persons in the medical profession are divided into researchers and practitioners, where the researcher aims at understanding the human body and the practitioner aims at curing other people. In medicine knowledge is often built by feedback from the practitioner to the researcher.
- **Manufacturing:** In manufacturing the relationship between process and product characteristics is generally well understood. Progress is made by experimenting with varying processes, building models of what occurs and measuring the effects on an over-worked product.

- **Psychology:** In psychology experiments have been used since the end of the 19th century, to base knowledge not on the personal experience of single human beings, but to check claims systematically and methodological controlled. Experiments are used to explain relationships of different variables and to predict certain events.

2.5 The Difference of SE

However, SE embodies *different* characteristics than other disciplines. Unlike in physics but like in manufacturing computer scientists are able to manipulate the essence of their product, because software and the corresponding development processes can easily be modified. But unlike in manufacturing the problem in SE is not production (because software can easily be copied) but development, because each new software product is different from the last [Bas96]. This implies that the mechanisms for model building are different in SE than in other disciplines.

A lot of computer science methods and technologies are influenced by human behaviour. Unlike in physics, the same experiment can produce different results based on the people involved. The human factor in SE prevents researchers from directly applying the causal-deterministic model of for example physics, because psychological and social influences are not completely determined by preceding facts [JM01]. Pfleeger [Pfl99] notes that researchers still often view SE as a natural process like in physics with deterministic effects to causes. But in fact software development is a social process, which leads to stochastic effects to causes. The occurrence of an effect can be described best by a probability function. For example, a software company with a high development certification level is not guaranteed to develop high quality software. Yet, the customers are assured, that they will actually get good software from this company with a high probability.

Another difference in SE is the amount of variables, which has to be considered on the outcome of an experiment. Software products, processes, the goals to be achieved and the context are all variable [Bas96]. A result of the high variability in SE is a lack of useful models for reasoning about software processes and products can still be observed. Because researchers have not been able to construct mathematical tractable models, like it is possible in physics and manufacturing, they have the tendency to not build models at all. Few attempts have been made to construct non-mathematical models, heuristics and models representing simple variable relationships, which for example is common in medicine.

3 Current State of Experimentation in SE

After introducing research approaches in the preceding section, the status of experimentation in SE shall be analysed in the following section. Two meta-studies about experimentation in SE are often cited and will be described more detailed. Additionally, several authors have written articles about the problems and trends in experimentation and some of their remarks will be summed up afterwards.

3.1 Study by Tichy et. al. (1994)

In 1994 Tichy et. al [TLPH95, LHPT94] conducted a quantitative study surveying over 400 research articles and looked for the amount of experimental validation. Included in this study for instance were articles from the ACM Transactions on Computer Systems (TOCS), ACM Transactions on Programming Languages (TOPLAS), IEEE Transactions on Software Engineering (TSE), and a random sample of articles published from the ACM in 1993. Also included for comparison with other disciplines were journals entitled Neural Computing (NC) and Optical Engineering (OE). The area NC was chosen because it overlaps with computer sciences (CS) and has a similar level

of youth like CS. OE in contrast was chosen, because it has lots of immediate applications like CS, but also has a longer history.

The metric used for measuring the amount of experimentation in design and modelling articles was simply the physical space the authors reserved for experimental evaluation. This metric was chosen because the physical space correlates with the importance the authors attached to their experimental evaluation and also the quality of the experiments.

Interestingly, over 40% of CS papers about design and modelling completely excluded experimentation, in software engineering papers the number was even higher at 50%. In contrast, only 14% of the NC and OE articles did not contain experimental evaluations. The CS papers contained a significantly lower amount of purely empirical papers than in NC and OE. Hypothesis testing articles were rare in all samples (1%). The number of papers dedicating more than 20% of their space to experimental validation was significantly lower in CS (31%) than in NC and OE (67%).

The authors of the study used these results to disprove a common perception in CS that the lack of experimentation is resulting from the youth of the discipline. During the conduction of the study the field of NC was only six years old, but had a much better level of experimentation established that was comparable to the much older discipline of OE. Although NC overlaps with CS, computer scientist are actually a minority among the NC community.

An explanation the authors offer for the lack of experimentation in CS is the lack of well established measuring techniques. Another problem is the fact, that a lot of conference committees accept papers without experimental validations. The authors encourage conference committees to set higher standards for accepting papers and also viewing empirical work as first class science.

3.2 Study by Zelkowitz et. al. (1997)

Zelkowitz and Wallace [ZW97] conducted a similar study like Tichy and analysed 612 SE papers and 137 papers from other sciences (for example physics, management science and behaviour theory). SE papers were from three different years (1985, 1990, 1995) and were taken from IEEE Transactions on Software Engineering, IEEE Software and the Proceedings of the International Conference on Software Engineering (ICSE) of the corresponding year. The articles from other disciplines were taken from various journals of the respective sciences from the years 1991-1996.

Similar to the study by Tichy, here, about one-third of the SE articles had no experimental validation at all. But the percentage was decreasing over the years (1985: 36.4%, 1990: 29.2%, 1995: 19.4%). About one-third of the articles contained only assertions as experimental validation. In assertions the developer of a technolgy becomes the experimenter and the subject of the study and usually does not perform any kind of control. These are often preliminary forms of validation and are potentially biased. 5% of the articles used simulation techniques for result validations, while the other data collection methods (see section 2.3) were found in only 1-3% of the articles.

Concerning other sciences, the authors observed specific patterns of experimentation approaches in each fields. Physical publications commonly contained dynamic analyses and simulations. In psychological articles mainly replicated and synthetic experiments could be found, while anthropological papers used passive techniques on historical data like legacy data and literature search.

Additionally to the confirmation of Tichy's result, that experimental validation is under-represented in CS, in this study the authors found out, that one-third of papers which actually contained experimental validation only did it in an insufficient and weak way (with assertions). This means that authors are starting to realize the need for experimental validation, yet still do not do it with strong methods. Encouraging is the fact, that the authors observed a development with fewer papers without experimentation comparing the numbers from 1985, 1990 and 1995.

Some qualitative observations of the study are mentioned in [ZW98]. Often, researcher do not succeed in stating their goals explicitly and clearly. Sections containing experimental validation

were sometimes not labelled accordingly and the terms "case study", "controlled experiment" were used very loosely with different meanings.

3.3 Further analyses

Fenton et. al. [FPG94] also discuss the current state of experimentation in SE. Although Basili et. al. had made several recommendation for experimentation in SE [BSH86], most of the experiments documented later did not follow their recommendation. The authors make five observations:

- **Intuitive research:** Too many concepts in SE are still based on so-called "analytical advocacy research", meaning that concepts have been described, analysed for their benefits informally and recommended for practical use but that rigorous, quantitative experimentation of them is missing. For example, formal methods in SE, which use mathematical precise specifications of software (e.g. the language Z [Spi88]) to prove its correctness, have been used for decades. It has been claimed that these methods are cost saving and that they reduce the amount of product failures. However, quantified evidence to support these claims does not appear in the respective publications. Nevertheless, counter-examples, in which methods have become a standard in SE because of empirical analysis, also exist (for example the use of inspections to uncover code-defects).

- **Experimental design flaws:** Experimental designs in SE are still often flawed. For example, Shneiderman showed with an experiment that program flowcharts did not increase the comprehension of program behaviour better than pseudocode. Years later, Scanlan made a comparison of flowcharts and pseudocode and investigated the amount of time needed to understand a program and the amount of time needed to make appropriate changes to it. In both dimensions using flowcharts was superior to using pseudocode. Fenton et. al. claim that flaws in experimental designs are a result of the thin representation of experimentation, statistical analysis and measurement principles in the computer science curricula of universities.

- **Toy analysis:** Empirical investigations in SE too often only analyse "toy projects in toy situations". The costs of large-scale industrial experiments lead most researchers to only experiment with student groups on small artificial examples. It is often unknown how the results of such experiments scale up and whether the results can be generalized at all. Because of this a better cooperation of research institutes and the software industry is recommended.

- **Inappropriate statistics:** Many experiments use inappropriate measures and misuse statistical methods. For example, several experiments can be found that use a nominal scale for their data and apply means and standard deviations to that data, although this data can only be analysed in terms of frequency and mode.

- **No long-term view:** Some experiments are conducted over a too short period of time and thus omit long-term effects. For example, the comparison of the benefits of the programming languages Ada and Fortran at first showed that Ada programmers were less productive and delivered programs with less quality than the Fortran programmers. However, the study did not take into account that Ada has a long learning curve and that the actual benefits of Ada can only be observed after the programmers had at least implemented three projects with this language.

Perry et. al. [PPV00] mention that the amount of empirical studies and also their quality is rising over the last 10-20 years. Empirical Validation is still not a standard part of research papers, yet a powerful addition. Especially in the testing community empirical studies are quite common. US funding agencies such as the National Science Foundation (NSF) and the National Academy

of Sciences are realizing the importance of empirical studies and are sponsoring for example the Experimental and Integrative Activities program or a workshop on the topic of statistics and software engineering [Pre96]. Furthermore, Perry et. al. observe, that the awareness for empirical studies is growing, as can be seen in tutorials, panels and presentations at major conferences such as ICSE, FSE and ICSM.

The authors notice that the discipline is still suffering from several systemic problems. Most of them originate from misunderstandings about the reasons for experiments and their proper conduction. Many researchers only use empirical studies retrospectively for early validation of their research results, but do not remember, that studies can also be used pro-actively to direct research.

Additionally, it is often tried to make the perfect study without flaws, which in the authors' opinion is impossible. Instead, more interest should be directed to the conclusions drawn from studies, which is often a weak part of such studies. Many studies only observe the obvious, thereby encouraging an argument by intuition approach. More studies should investigate unintuitive relationships.

Data collection and analysis is sometimes done very extensive and precise in empirical studies today, but the use of data to answer questions is neglected. In these cases no conclusions are drawn making it difficult to learn anything from such a study. Often studies simply lack hypotheses, do not ask insightful questions and therefore contain no well-defined end. Because the drawing of conclusions is disregarded, researchers are reluctant to generalize their results. But without generalized results it is hard to make any progress in SE.

Concluding this section, it can be stated that experimentation is still under-represented in SE, but a positive trend can be observed. Although the awareness for the need of experimentation is starting to increase, lots of the experiments conducted today are still flawed in their designs and in the ability to draw proper conclusions.

4 Common Fallacies on Experiments

Researchers often try to justify their lack of experimentation. Tichy [Tic98] wrote an article refuting common comments given by computer scientist when they are asked about why they neglect the experimental validation of their research results:

- **"Traditional scientific method isn't applicable":** A major difference in computer sciences to other disciplines is that information is neither energy nor matter. Several researchers conclude that traditional scientific methods are thus not applicable in computer science. This is a fallacy, because researchers can use the same methods as in traditional sciences, like observation of phenomena, formulation of explanation and theories and testing, to gain an understanding on the nature of information processes. The fact that the information itself is different from energy or matter does not alter the treatment of information processes from other sciences.

- **"The current level of experimentation is good enough":** This can easily be refuted with the studies mentioned earlier [TLPH95, ZW98], which analysed the amount of experimentation in SE papers. Experimentation has the purpose of reducing uncertainty about untested theories, may serve to start new areas of research and eliminates fruitless approaches thus directing the science process.

- **"Experiments costs too much":** Tichy admits, that experiments cost a lot of effort, but he states that this effort is justified, if the research question asked is of importance and the gained insights outweigh the costs. For example, Isaac Eddington undertook an expensive

expedition to West Africa in 1919 to observe a total solar eclipse. Doing so he was able to check Einstein's theory that gravity bends light when passing large stars. In this case the experiment required a lot of effort, but the importance of the answer to the research question was tremendous.
A way to overcome the high costs of experiments might be involving software industry. Companies may be able to get competitive advantages out of experiments and thus may be interested in sponsoring such research. Furthermore, Tichy mentions that a possible cheaper substitute to experiments might be simulations techniques.

- **"Demonstrations will suffice":** Demonstrations only provide proof of concepts, but not hard evidence to research questions. Demos only illustrate a potential and are dependent on the authors to generalize the results. The need for replication of results is not stressed by demonstrations.

- **"There is too much noise in the way":** Often researchers complain, that too many variables have to be controlled in an experiment and that the results are hard to interpret because they are swamped by noise. Tichy advises researchers to use benchmarks to simplify repeated experiments. Noise created by human subjects may be reduced by using techniques from areas like psychology and sociology.

- **"Progress will slow":** As experiments require a lot of effort to be conducted, many argue that experimenting slows the flow of ideas down, if every idea must be extensively validated with experiments. Contrary, Tichy replies, that experimenting might in fact speed up the science progress, because fruitless approaches would be discarded earlier and science would focus more on the most promising approaches.

- **"Technology changes too fast":** Problematic in computer science is the fact that technology changes so fast that ideas might not be relevant anymore if experimental validations are finally available. If a research question falls into this category it is obviously formulated too narrow. Experimentation techniques should be applied on deeper, fundamental questions and not on the newest fashion of tools and methods.

- **"You will never get it published":** Theoretical computer scientists usually expect perfection and absolute certainty in results published. But because experiments are always flawed in some way [PPV00], it is difficult to get experimentation papers accepted at important conferences. Tichy replies that several journals exists, that would welcome more experimental papers, but that the supply is actually too low.

Tichy also states that intuition and personal experience is not sufficient to claim the applicability of a product or process in a matured engineering discipline. Several examples are known in computer science in which intuition falsely favoured an opinion (the need for meetings in code reviews, the lesser failure probabilities of multi-version programs). It is also dangerous to simply trust well-known experts and not to demand hard evidence of their claims. A fundamental precondition of science is, that it is based on healthy scepticism.

5 Future Direction of Experimentation in SE

As the need for experimentation in SE has been emphasized before and problems of the discipline have been examined, the following section deals with the possible future of experimentation. Most of the authors, who have complained about the lack of experimentation in SE, have also made recommendations about what could be improved (next subsection). An approach still neglected

in SE experiments is the repetition of empirical studies and the creation of families of studies to answer larger research questions (last subsection).

5.1 General Recommendations

Basili [Bas96] advocates the use of the Goal-Question-Metric (GQM) method (ref Klaus Krogmanns Ausarbeitung) for a better direction of experimentation. Especially the level of sophistication of the goals towards experiments are designed must improve. The results of experiments need to be shared more and the results by one group need to be used by other groups. An organization called International Software Engineering Research Network (ISERN) has been established especially for this purpose. A forum for empirical researchers is provided by the International Journal of Empirical Software Engineering by Kluwer.

Perry et. al. [PPV00] state that in order to improve the quality of empirical studies in SE, more clarity about the goals of the studies is needed. Researchers must ask important and sophisticated questions and establish causal, actionable and general principles. Causality implies constructing a chain of factors influencing each other, while actionable principles require factors which can be controlled effectively. Factors also have to be so much general, that the results are relevant to multiple persons in multiple contexts.

Credible studies have a high degree of confidence in the results. To create such studies the internal, external and construct validity have to be checked explicitly. Enough data has to be published to let other researcher recheck the validity of a study. Additionally, studies should always be conducted on the basis of hypotheses. If a study is not sufficient enough to create a causal relationship of factors, several alternative explanations may be proposed and data from other source might be used to discredit certain alternatives.

Glass [Gla94] emphasises the need for a close cooperation between practitioners and researchers to improve the state of research in computer sciences. For him, the Software Engineering Laboratory (SEL) [BCM+92] as an institution involving academic (University of Maryland), industrial (Computer Sciences Coporations) and governmental (NASA-Goddard, as sponsors) organizations is an exemplary model which should be replicated by other researchers and countries to stimulate the exchange between practice and research.

Recently, Kitchenham et. al. [KPP+02] have proposed a set of guidelines for empirical research in SE, which are based on a review of guidelines for medical researchers. The guidelines cover six topics:

- **Experimental context:** It is important for studies to include information about the industrial context they were conducted in. The research hypotheses have to be discussed and also how they have been derived. The main guidelines about the context are to state and discuss the goal of the study and to include sufficient details for researchers as well as practitioners.

- **Experimental design:** Studies must describe the population under analysis (e.g. students, practitioners, ...) and the sampling technique used for it. It has to be documented what intervention have been conducted and what method has been used to reduce bias and to determine the sample size.

- **Conduct of the experiment and data collection:** Because the measures for the outcome of a study are not standardized they have to be documented in sufficient detail. The entity, attribute, unit and counting rules of measures must be defined. Quality control methods used on the data and data about subjects who dropped out of the study should be presented.

- **Analysis:** Procedures used to control for multiple testing should be specified and a sensitivity analysis should be performed. Additionally, the data must not violate the assumptions of the

tests used on them and for the verification of the results appropriate quality control methods should be applied.

- **Presentation of results:** A reference should be provided for all statistical procedures used and the statistical package used should be reported. The magnitude of effects and the confidence limits of quantitative results should be presented as well as confidence levels. If possible, the raw data should be provided with the study, so that independent researchers can draw their own interpretations from them. Descriptive statistics should be used in an appropriate way. Graphics can increase the degree of understandability of a study if used correctly.

- **Interpretation of results:** Researchers should differentiate between statistical significance and practical importance. The type of the study needs to be define and limitations of it should be discussed.

5.2 Repeatability and Families of Studies

Perry et. al. [PPV00] note, that it is often not possible to design studies for complex issues and difficult questions, especially considering the effort and costs needed for empirical studies. In this case it is necessary to focus on smaller problems and to create multiple studies, which results possibly can be combined to answer more deep questions. The credibility of empirical studies is improved drastically if other researcher are provided with enough information to reproduce the results.

The same is mentioned by Lewis et. al., who state: "The use of precise, repeatable experiments is the hallmark of a mature scientific or engineering discipline." [LHKS92]. In the history of science several examples for experiments can be found, which could not be repeated by other scientists as summarized by Juristo et. al. [JM01]. For example, the psychodynamic theories developed by Freud are criticised as being unscientific because they cannot be verified or disproved empirically. In 1989 two physicians claimed they had successfully conducted a cold fusion during an experiment, but after they published the design of their experiment, other scientist were not able to reproduce their result.

Apart from external replication run by independent researchers, also internal replications run by the original experimentator are necessary to improve the trust in an experiment. This might be hard to do in practice because a software project cannot be repeated precisely, yet it is not an excuse for not experimenting at all.

Basili also emphasises the importance of replicated experiments [BSL99]. Too many SE experiments stay isolated and do not lead to a larger body of knowledge. This can only be achieved by a set of unifying principles that allows the combination and generalization of results. The authors propose a framework including the GQM method for a family of experiments.

Additionally the authors classify three major categories for replicated experiments:

- **Replication without a variation of the research hypothesis:** These include strict replications and replications that alter the way an experiment is run. Strict replications aim at a very accurate replication of the original experiment and ensure the repeatability of an observation. Replications that are run with a different experimental set-up than in the original case but with the same hypothesis might reveal flaws of the internal validity of an experiment.

- **Replication with a variation of the research hypothesis:** Included are variations of independent, dependent and context variables. Varying *independent* variables means changing variables intrinsic to the object under study, for example changing an attribute of the process or product under study. This form of replication is only possible if the experimental design

has been made explicit by the original authors. *Dependent* variables are variables regarding the focus of the study. An example of a replication with a change of the dependent variables may be using other metrics or measurements for the effects that are be studied. Varying the *context* of an experiment might help in identifying influencing environmental factors. For example an experiment might be carried out with a group of professionals opposed to a group of students to analyse the influence of personal experience to the studied effects.

- **Replications that extend the theory:** Replications in this category change a large part of the process or product under analysis to determine its limit of effectiveness.

6 Critical Reflection

After reviewing the literature about the role of experimentation in SE, some critical remarks about this topic are summed up in this section.

Tichy tried to refute common comments by researchers, who neglect empirical evaluations of their work, and tried to encourage scientists to put more emphasis on experimentation. However, overcoming the organisational effort for proper controlled experiments is still a major problem. Lots of research is conducted by PhD-students, who simply do not have the means and time to conduct elaborate experiments. Including the software industry into experimentation as suggested by Tichy is also very difficult, because practitioners are hard to motivate to put resources and money on testing unproven new methods, if the direct value for their customers cannot be made evident. Also, as pointed out by Tichy, it is not necessary to evaluate every small research proposal with large controlled experiments. But criterias for necessity of experiments are still informal and hard to determine.

One problem sometimes mentioned by the authors cited here is the researcher's inappropriate knowledge of empirical methods in software engineering. This point is clearly underestimated by the empirical software engineering community. The researchers' knowledge is inappropriate because universities do not teach empirical methods, and such courses are not part of the standard curriculums. If experiments are conducted, the methods of experimentation have often only been learned ad-hoc by the researchers, if they have laid emphasis on a methodological approach at all. Young researchers are seldomly taught the proper conduction of an experiment and have to collect experience about it on their own. The facts that books about methods like experiments or case studies for SE have appeared only recently (in the last 5 years) and that the methods are still not established well enough also contribute to this situation. Additionally the inappropriate application of statistical methods in experimentation can also be seen as a result of the missing courses in the computer science curriculum.

As seen in the literature analysed above, researchers often criticize experiments that are carried out with students as the subject under analysis, claiming that the results are not transferable to experienced practitioners. Apart from the organisational difficulties and high effort of including practitioners, there are other reason to counter this criticism. First, experienced under-graduate students often become practitioners just a short time later, so that their qualification and performance is not as different to practitioners as it seems. Second, the experience by practitioners might actually distort the results of an experiment, because specialists might have an unusual advantage over common developers. Thus, the results obtained with some experienced specialists might not be generalisable for the average developer.

Another reason why researchers are reluctant to experiment, which has been neglected in the literature reviewed, is simply the fact that some researchers do not like to experiment and consider it an inconvenient but necessary task. Most scientists rather want to create new ideas than spending time on validating old ones. Computer scientists in particular are more interested in technical problems and how to solve them. If their solutions are intuitively correct, most of them do not

bother to conduct further evaluation on them. They even might be scared to prove their solutions empirically wrong. A stronger motivation for experimentation has to be created, possibly by documenting popular experiments in SE, which revealed unexpected results.

In the future experiments will be conducted with a higher quality and also more experiments will conducted. But whether a level of experimentation can and needs to be established in SE like it has been in other disciplines (physics, medicine, psychology) remains doubtful.

7 Conclusions

In this paper the situation of experimentation in SE in the past, present and future has been analysed. "Experimentation is central to the scientific process" [Tic98]. This statement is especially true for SE, because most research proposals cannot be proven formally. It has been discussed that experimentation is vital for SE to become a matured scientific discipline. Multiple data collection methods are known for empirical SE, controlled experiments are the method with the highest degree of confidence in the results. When analysing experimentation in SE, the special characteristics of the discipline (like the human factor and the high variability) have to be kept in mind.

In the past experimentation in SE has not been sufficient as two studies from the mid-nineties have shown, although an improving trend can be observed. The reluctance of researchers to experiment can be refuted with multiple arguments. Researchers in SE should always remember that empirical validation is an essential part of their work and that their proposals are not valid unless empirical evidence has been provided. In the future hopefully not only the quantity but also the quality of experiments can be improved. It should be easier to conduct good empirical studies in SE because an increasing body of literature about the topic has been published (e.g. [WRH+00, Pre01, JM01]. Additionally, replicated experiments and families of studies should help to create larger bodies of knowledge.

References

[Bas93] Basili, V. R.: The Experimental Paradigm in Software Engineering. In: *Proceedings of the International Workshop on Experimental Software Engineering Issues: Critical Assessment and Future Directions*, London, UK: Springer-Verlag, 1993, ISBN 3-540-57092-6, pp. 3–12

[Bas96] —— The Role of Experimentation in Software Engineering: Past, Current, and Future. In: *ICSE*, 1996, pp. 442–449

[BCM+92] Basili, V.; Caldiera, G.; McGarry, F.; Pajerski, R.; Page, G.; Waligora, S.: The software engineering laboratory: an operational software experience factory. In: *ICSE '92: Proceedings of the 14th international conference on Software engineering*, New York, NY, USA: ACM Press, 1992, ISBN 0-89791-504-6, pp. 370–381, doi:http://doi.acm.org/10.1145/143062.143154

[BSH86] Basili, V. R.; Selby, R. W.; Hutchens, D. H.: Experimentation in software engineering. In: *IEEE Trans. Softw. Eng.* 12 (1986), № 7, pp. 733–743, ISSN 0098-5589

[BSL99] Basili, V. R.; Shull, F.; Lanubile, F.: Building Knowledge through Families of Experiments. In: *IEEE Trans. Softw. Eng.* 25 (1999), № 4, pp. 456–473, ISSN 0098-5589, doi:http://dx.doi.org/10.1109/32.799939

[FPG94] Fenton, N.; Pfleeger, S. L.; Glass, R. L.: Science and Substance: A Challenge to Software Engineers. In: *IEEE Softw.* 11 (1994), № 4, pp. 86–95, ISSN 0740-7459, doi:http://dx.doi.org/10.1109/52.300094

[Gla94] Glass, R. L.: The Software-Research Crisis. In: *IEEE Softw.* 11 (1994), № 6, pp. 42–47, ISSN 0740-7459, doi:http://dx.doi.org/10.1109/52.329400

[JM01] Juristo, N.; Moreno, A. M.: *Basics of Software Engineering Experimentation.* Kluwer Academic Publishers, 2001

[JM03] —— *Lecture Notes on Empirical Software Engineering*, vol. 12 of *Series on Software Engineering and Knowledge Engineering.* World Scientific, 2003

[KPP+02] Kitchenham, B. A.; Pfleeger, S. L.; Pickard, L. M.; Jones, P. W.; Hoaglin, D. C.; Emam, K. E.; Rosenberg, J.: Preliminary guidelines for empirical research in software engineering. In: *IEEE Trans. Softw. Eng.* 28 (2002), № 8, pp. 721–734, ISSN 0098-5589, doi:http://dx.doi.org/10.1109/TSE.2002.1027796

[LHKS92] Lewis, J. A.; Henry, S. M.; Kafura, D. G.; Schulman, R. S.: On the relationship between the object-oriented paradigm and software reuse: An empirical investigation. In: *Journal of Object-Oriented Programming* 5 (1992), pp. 35–41

[LHPT94] Lukowicz, P.; Heinz, E. A.; Prechelt, L.; Tichy, W. F.: *Experimental Evaluation in Computer Science: A Quantitative Study.* tech. rep., Fakultät für Informatik, Universität Karlsruhe, August 1994

[MWB99] Murphy, G. C.; Walker, R. J.; Baniassad, E. L. A.: Evaluating Emerging Software Development Technologies: Lessons Learned from Assessing Aspect-Oriented Programming. In: *IEEE Trans. Softw. Eng.* 25 (1999), № 4, pp. 438–455, ISSN 0098-5589, doi:http://dx.doi.org/10.1109/32.799936

[Pfl99] Pfleeger, S. L.: Albert Einstein and Empirical Software Engineering. In: *Computer* 32 (1999), № 10, pp. 32–38, ISSN 0018-9162, doi:http://dx.doi.org/10.1109/2.796106

[PK04] Port, D.; Klappholz, D.: Empirical Research in the Software Engineering Classroom. In: *CSEET '04: Proceedings of the 17th Conference on Software Engineering Education and Training (CSEET'04)*, Washington, DC, USA: IEEE Computer Society, 2004, ISBN 0-7695-2099-5, pp. 132–137

[Pop59] Popper, K. R.: *The Logical of Scientific Discovery.* London: Hutchinson, 1959

[Pot93] Potts, C.: Software-Engineering Research Revisited. In: *IEEE Softw.* 10 (1993), № 5, pp. 19–28, ISSN 0740-7459, doi:http://dx.doi.org/10.1109/52.232392

[PPV00] Perry, D. E.; Porter, A. A.; Votta, L. G.: Empirical studies of software engineering: a roadmap. In: *ICSE - Future of SE Track*, 2000, pp. 345–355

[Pre96] Pregibon, D.: *Statistical Software Engineering.* National Academy of Sciences: Washington D.C., 1996

[Pre01] Prechelt, L.: *Kontrollierte Experimente in der Softwaretechnik.* Springer Verlag, 2001

[Spi88] Spivey, J. M.: *Understanding Z: a specification language and its formal semantics.* New York, NY, USA: Cambridge University Press, 1988, ISBN 0-521-33429-2

[Tic98] Tichy, W. F.: Should Computer Scientists Experiment More? In: *IEEE Computer* 31 (1998), № 5, pp. 32–40

[TLPH95] Tichy, W. F.; Lukowicz, P.; Prechelt, L.; Heinz, E. A.: Experimental evaluation in computer science: a quantitative study. In: *J. Syst. Softw.* 28 (1995), № 1, pp. 9–18, ISSN 0164-1212, doi:http://dx.doi.org/10.1016/0164-1212(94)00111-Y

[WBN+03] Walker, R. J.; Briand, L. C.; Notkin, D.; Seaman, C. B.; Tichy, W. F.: Panel: empirical validation: what, why, when, and how. In: *ICSE '03: Proceedings of the 25th International Conference on Software Engineering*, Washington, DC, USA: IEEE Computer Society, 2003, ISBN 0-7695-1877-X, pp. 721–722

[WRH+00] Wohling, C.; Runeson, P.; Höst, M.; Ohlsson, M.; Regnell, B.; Wesslen, A.: *Experimentation in Software Engineering – An Introduction.* Kluwer Academic Publishers, 2000

[ZW97] Zelkowitz, M. V.; Wallace, D. R.: Experimental Validation in Software Engineering. In: *Information and Software Technology* 39 (1997), pp. 735–743

[ZW98] —— Experimental Models for Validating Technology. In: *Computer* 31 (1998), № 5, pp. 23–31, ISSN 0018-9162, doi:http://dx.doi.org/10.1109/2.675630