Views on Internal and External Validity in Empirical Software Engineering

Janet Siegmund, Norbert Siegmund, and Sven Apel
University of Passau, Germany

***Abstract*—Empirical methods have grown common in software engineering, but there is no consensus on how to apply them properly. Is practical relevance key? Do internally valid studies have any value? Should we replicate more to address the tradeoff between internal and external validity? We asked the community how empirical research should take place in software engineering, with a focus on the tradeoff between internal and external validity and replication, complemented with a literature review about the status of empirical research in software engineering. We found that the opinions differ considerably, and that there is no consensus in the community when to focus on internal or external validity and how to conduct and review replications.**

I. Introduction

Empirical research in software engineering came a long way. From being received as a niche science, the awareness of its importance has increased. In 2005, empirical studies were found in about 2% of papers of major venues and conferences [31], while in recent years, almost all papers of ICSE, ESEC/FSE, and EMSE reported some kind of empirical evaluation (see Section III). Thus, the amount of empirically investigated claims has increased considerably.

With the rising awareness and usage of empirical studies, the question of where to go with empirical software-engineering research is also emerging. New programming languages, techniques, and paradigms, new tool support to improve debugging and testing, new visualizations to present information emerge almost daily, and claims regarding their merits need to be evaluated—otherwise, they remain claims. But, how should new approaches be evaluated? Do researchers focus on internal validity and control every aspect of the experiment setting, so that differences in the outcome can only be caused by the newly introduced technique? Or, do they focus on external validity and observe their technique in the wild, showing a real-world effect, but without knowing which factors actually caused the observed difference?

Both options, maximizing internal or maximizing external validity, have their benefits and drawbacks, which we illustrate by the example of evaluating the influence of using a new tool on the performance of beginning programmers: The first option (maximizing internal validity) allows researchers to exclude almost all influencing factors, so that they can observe in a highly controlled setting whether the new tool improves one aspect of the every-day work of beginning programmers. This way, researchers can draw sound conclusions about the reasons of improvement or degradation, but at the cost of generalizability. With the second option (maximizing external validity), researchers can observe whether the tool has any effect on different types of developers in an every-day setting, but at the cost of not being able to unambiguously understand why the new tool affects the work flow—maybe it is just because it is new.

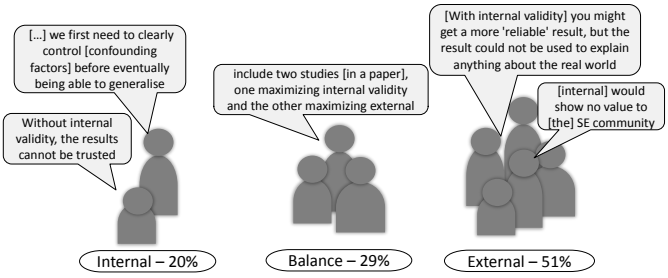

Fig. 1. Preferences for internal vs. external validity among program-committee and editorial-board members.

There is an inherent tradeoff in empirical research: Do we want observations that we can fully explain, but with a limited generalizability, or do we want results that are applicable to a variety of circumstances, but where we cannot reliably explain underlying factors and relationships? Due to the options' different objectives, we cannot choose both. Deciding for one of these options is not easy, and existing general guidelines, for example by Wohlin [36] or Juristo and Moreno [12], are too general to assist in making this decision.

With our work, we want to raise the awareness of this problem: *Should we focus on internal or external validity? Should we focus on one first and then on the other? Should we balance both kinds of validity, not maximizing one?* In the end, every time we are planning an experiment, we must ask ourselves: Do we ask the right questions? For example, is it better to ask principal questions, such as whether static type systems ease program comprehension compared to dynamic type systems, or is it better to ask broadly which commonly-used programming languages are more superior in what circumstances? Do we want pure, ground research, or applied research with immediate practical relevance? Is there even a way to design studies such that we can answer both kinds of questions at the same time, or is there no way around replications (i.e., exactly repeated studies or studies that deviate from the original study design only in a few, well-selected factors) in software-engineering research?

In the remainder of this paper, we present the results of a literature review to evaluate the kind and extent of empirical methods used in software engineering, and to get an impression of the role of internal and external validity and replications (Sec. III), followed by example studies maximizing one or the

other (Sec. IV). Thereupon, as a main contribution of this paper, we present the results of an online survey among 79 program-committee and editorial-board members—"key players" in their field—of 11 major software-engineering venues regarding their perception and opinion on how to address the tradeoff between internal and external validity (Sec. V to VII).

In a nutshell, we found large differences in the opinions regarding the importance of internally and externally valid studies and a lack of awareness of the tradeoff between the two, which we illustrate in Figure 1. Furthermore, many survey participants are aware of the need of replication, but there is substantial disagreement about the kind and extent of the delta that is necessary for proper replication. Thus, there are different reviewer expectations to a paper and there are no proper guidelines for reviewing a paper in this hindsight.

Our research is meant to stimulate researchers across multiple areas rethinking their expectations and standards of empirical research, including educating (young) software-engineering researchers, assisting researchers in evaluating their work, helping reviewers in judging the soundness of a research paper, and providing guidelines for planning empirical research.

In summary, we make the following contributions:

- Overview of the state of the art of empirical software engineering in three major (empirical) software-engineering venues, with a focus on the role of internal and external validity and replication.
- Overview of the opinions of the "key players" of the software-engineering community, based on a survey among 79 program-committee and editorial-board members of 11 major software-engineering venues.
- Suggestions on how to conduct empirical research in software engineering.
- A discussion of open issues, meant to initiate a discussion in the community.

All data from the literature review and survey are available on a supplementary Web site: http://www.infosun.fim.uni-passau.de/spl/janet/ese/ . As an overarching theme of our work, let us quote David Parnas on software-engineering research [22]:

"It is time to stop "exploring" and start experimenting."

II. Related Work

There is considerable work concerned with the status of empirical research or guidelines on how to conduct empirical research in the area of software engineering. However, we are not aware of any work surveying program-committee or editorial-board members to assess their opinion and suggestions addressing the tradeoff between internal and external validity in empirical software engineering.

Guidelines: There is a long history of advocating and evaluating empirical research in software engineering. As early as 1986, Basili and others published guidelines on empirical research [3], comprising a framework to describe experimental work. Furthermore, Basili proposed the goal-question-metric approach to guide researchers in defining their research goals, such that the context of an experimental setting is clearly described [1]. Kitchenham and Charters proposed guidelines on how to conduct systematic surveys in software engineering, following guidelines from medical research [18]. Kitchenham and others complement this research with a study on systematic literature reviews [16] and on the repeatability of systematic literature reviews [17]. Ko and others present guidelines for conducting controlled experiments to evaluate software-engineering tools with human participants [19]. These guidelines arrange research activities along ten steps, including recruitment and training of participants as well as task design. Siegmund and Schuman provide an overview of confounding parameters that influence the outcome of an experiment and that need to be controlled for [28]. Sjøberg and others make the case for more realistic settings in software-engineering research, stressing the role of funding to pay professional developers [29]. Furthermore, they report on current problems of empirical research, for which the lack of practical relevance is still an issue, among others [30]. As solutions, they suggest to give more competence to empirical researchers (e.g., by training) or to improve the collaboration between industry and academia. Their vision of empirical research in 5 to 10 years is striving for more practical relevance, more synthesis of knowledge, and more theory building. Tichy and others reported on the status of experimental research in software engineering compared to optical engineering and neural computation, concluding that there is only little empirical research in software engineering [34]. Consequently, Tichy stated that computer scientists should experiment more [35]. He also provided guidelines for reviewing empirical research, which describe common arguments that reviewers use to reject a paper, and explanations for why these are not valid for rejection [33].

Replication: There is considerable work in the direction of replication (i.e., a repetition of an experiment under similar conditions, but with specified variation, such as a new sample [36]). Basili and others stated that "too many studies tend to be isolated and are not replicated, either by the same researchers or by others" [2]. They describe a framework for categorizing related studies, which can then be viewed in context, rather than viewing each study in isolation. Shull and others describe the role of exact and conceptual replication in software engineering, which both are standard in behavioral science [27], but not in software-engineering research, as our literature review and our survey show. Juristo and Vegas describe the role of non-exact replications, explaining that exact replications are almost impossible to conduct in software-engineering research, because the context is so complex (e.g., how techniques were applied as well as the knowledge of participants and how they were trained) [13]. Thus, many researchers give up (e.g., [20]) or do not publish their efforts because of contradicting results. To improve this situation, Juristo and Vegas suggest to loosen the restrictions for the exactness of replication studies, so that some obstacles of replication studies can be removed.

Status of empirical research: Do all these guidelines and insights affect the status of empirical research? There is evidence that the amount of empirical research has increased: While Sjøberg and others found that in major software-engineering venues from 1992 to 2002, only 1.9% of the papers reported a controlled experiment [31], this fraction has increased in recent years, for example, as observed by Ivarsson

and Gorschek in the domain of requirements engineering [9]. In our literature review, we found a large number of papers that conducted some sort of empirical evaluation (Section III). But does that also count for the quality? Ivarsson and Gorschek found that, in requirements engineering, the rigor of empirical studies has improved, but practical relevance has not [9]. Nagappan and others found that selected subject systems cover a wide range of different dimensions, such as team size and project size, which positively affects external validity [21]. Sjøberg and others [31] as well as Dybå and others [5] noted, among others, that the reports of empirical studies often lack important details. For example, threats to validity are often vague and unsystematic despite the numerous guidelines on how to describe empirical studies [4], [10], [11], [15], [36]. Kampenes and others analyzed the conduct of quasi experiments and found that the design, analysis, and reporting can be improved [14].

Thus, despite the long history of advocating empirical research in software engineering, there is still much room for improvement, which Zannier and others nicely phrased [37]:

> "[G]iven the numerous clear and repeated messages of [numerous researchers], which date back almost 20 years and provide results that date even further in history, we must ask ourselves, at what point will the message become clear?"

Our work—in particular, the analysis of the survey results—strives for making this message clearer.

III. State of the Art: A Literature Review

As not being addressed by previous work (cf. Section II), we conducted a literature review of three of the major (empirical) software-engineering venues, to get an overview of the current status of empirical research in software engineering. Our sample consisted of all 405 full technical papers of ICSE (2012, 2013), ESEC/FSE (2011 to 2013), and EMSE (2011 to 2013), the major venues in (empirical) software engineering. While this selection is limited, it still gives a good impression of the state of the art. We manually examined each paper regarding the use of empirical methods, recruitment of human participants (students or professionals), replication, and presentation of validity. To this end, we skimmed each paper and searched with a set of keywords[1]. In Figure 2, we provide an overview of the process and findings of the literature review.

First, we determined whether an empirical method (e.g., case study, controlled experiment) was applied, which happened in overwhelmingly 381 (94 %) papers. This seems like a large increase compared to the 1.9 % that Sjøberg and others found about 10 years earlier [31]; but, to be fair, they only included controlled experiments with human participants.

Second, we also determined whether a study was conducted with or without human participants. Of all 405 papers, 87 (21 %) recruited human participants, and 294 (73 %) had no human participants, but evaluated other properties, such as performance or test coverage. Now, we can draw a more close comparison with the Sjøberg study, indicating that the human factor is nowadays considered as more important.

[1]Keywords: empirical, student, profession, developer, subject, participant, human, repeat, replicat, further.

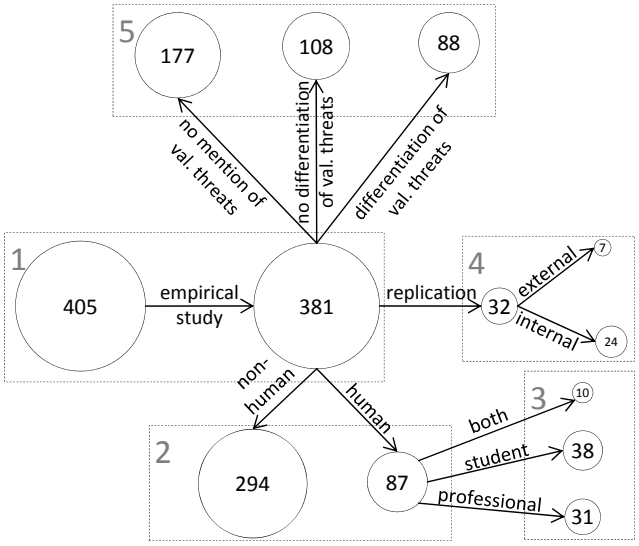

Fig. 2. Fraction of empirical studies that meet certain criteria. Numbers in circles represent the absolute number of papers, to which the circle area is proportional. Gray numbers refer to the paragraphs in the text.

Third, there is a diversity in the selection of human participants. Of the 87 studies involving human participants, 38 recruited students, 31 professionals, and 10 both. In 7 papers, the participants were not specified closer, and in 1 study, researchers used Mechanical Turk. Thus, relying on professional programmers is not the exception.

Fourth, we determined whether a paper reported on a replication: Of the 381 papers, 347 (91 %) did not conduct a replication; 32 (8 %) did a replication. Of these, 24 did an internal replication (i.e., by the same group) and 7 reported on an external (i.e., by another group) replication (one paper was not clear on the kind of replication). This result suggests that replication studies, especially external ones, are underrepresented in software engineering.

Fifth, we looked at the discussion of validity. Of the 381 papers using an empirical method, surprisingly 177 (46 %) papers did not explicitly mention threats to validity at all. In 108 (28 %) papers, authors discussed threats to validity, but did not differentiate between internal or external (or other kinds of) validity. In a few papers (5), the discussion was not explicit, but hidden in a paragraph of the discussion or conclusion. The remaining 88 (23 %) papers differentiated between different kinds of validity (mostly internal, external, construct, and conclusion validity). While this result may be biased by the selection of venues (i.e., 2 conferences and 1 journal; conferences impose a strict page limit to which the discussion of validity is often sacrificed), it nevertheless suggests that there is considerable room for improvement in the discussion and documentation of threats to validity.

To summarize, empirical research seems to be an integral part of software-engineering research nowadays. However, from the methodological point of view, the individual standards differ considerably.

IV. Maximizing Internal or External Validity

To illustrate the merits of internal and external validity as well as to provide a foundation for the survey, we introduce two studies as running examples, one maximizing internal, the other maximizing external validity.

A study set up that maximizes internal validity was designed by Hanenberg [6]. He evaluated whether static type systems, as compared to dynamic type systems, influence development time. To control for confounding parameters—which might influence the result beside the merits of the two kinds of type systems—he developed a language and corresponding IDE, solely for the purpose of this experiment. The language and IDE differed only in the type system used, nothing else. Furthermore, Hanenberg recruited students with similar programming experience as participants and let them implement two small tasks. In this highly controlled setting, he was able to conclude that a difference in the performance of student programmers is caused only by the type system, nothing else.

However, how can one generalize the result of such a controlled experiment? Should developers switch to another type system? Obviously, giving a recommendation is difficult, because the setting of Hanenberg's experiment was artificial.

So, how about another setting, in which several different professional developers from different companies work with different programming languages on every-day tasks? Röhm and others used such a setting to observe how professional programmers work with source code [23]. While being realistic, this set up has a lot of confounding parameters that have not been controlled for, such as the complexity of the task, programming language, and programming experience of the developers. Thus, while such a general setting produces general, potentially practically relevant results, it is unclear how the results emerged—many factors could have affected the results

Both kinds of study setting are viable and lead to interesting results, but which one is preferable and in which situation? We conducted an online survey among 79 program-committee and editorial-board members, to provide answers to this and related questions.

V. Survey Setup

In this section, we give a detailed overview of our survey following the guidelines provided by Jedlitschka and others [10].

A. Objective

With our survey, we targeted several research objectives:[2]

RO_1 Assess the awareness of the community of the tradeoff between external and internal validity.

RO_2 Assess the opinion of the community regarding how to address this tradeoff.

RO_3 Assess the opinion of the community regarding the role of replication.

These objectives emerged from discussions with researchers at different conferences and workshops as well as from reviews of empirical research papers. We experienced that, sometimes, there is a lack of appreciation for internally valid studies, and that external validity or practical relevance of a study is seen as most important. Thus, we assess the awareness of the community (RO_1) as well as their suggestions on how to address this tradeoff (RO_2). Furthermore, in other disciplines, replicating studies is a commonly accepted way to address this tradeoff—in medicine or physics, only replicated results are accepted. Thus, we asked the community about what they think of replication to address this tradeoff in software-engineering research (RO_3).

[2]We refer to "objectives" rather than "questions" to avoid any confusion with the actual questions of the survey.

B. Participants

As participants, we contacted the program-committee and editorial-board members of major (empirical) software-engineering venues. We decided to balance venues with empirical focus and venues with a general software-engineering focus to reduce the bias toward empirically interested researchers. This way, we can assess the opinion of renowned researchers and experts of their area (empirical and not empirical). Clearly, the "key players" shape the future of software-engineering research by deciding on the acceptance of research papers, guide young researchers, and advise funding agencies.

To ensure that the participants have been reviewing current papers, we extracted the e-mail addresses of members active in the years 2010 to 2013 from the following venues:

- ASE (Automated Software Engineering)
- EASE (Evaluation and Assessment in Software Engineering)
- ECOOP (Object-Oriented Programming)
- EMSE (Empirical Software Engineering)
- ESEC/FSE (Foundations of Software Engineering)
- ESEM (Empirical Software Engineering and Measurement)
- GPCE (Generative Programming)
- ICPC (Program Comprehension)
- ICSE (Software Engineering)
- ICSM (Software Maintenance)
- OOPSLA (Object-Oriented Programming)
- TOSEM (Software Engineering and Methodology)
- TSE (Software Engineering)

On average, a participant was on the program committee or editorial board of 3.6 ($\pm$ 2) different venues, with a minimum of 1 and a maximum of 9 different venues.

C. Questionnaire and Conduct

We designed a questionnaire that covers several aspects of empirical research, in particular, focusing on internal and external validity and replication. We included several closed questions, for each of which we additionally asked the participants to elaborate on their decision. Furthermore, we included several open questions asking for suggestions, for example, "Do you have any suggestions on how empirical researchers can solve the dilemma of internal vs. external validity of empirical work in computer science?". All questions were optional.

To ensure that the participants knew what a highly internally and highly externally valid study looks like, we described a research question inspired by Hanenberg's study (Sec. IV) and two settings to evaluate the corresponding research question, one maximizing internal validity and one maximizing external validity. In Table I, we list all survey questions and map them to our research objectives.

TABLE I
QUESTIONS OF THE SURVEY TO ANSWER THE RESEARCH OBJECTIVES (RO). BEFORE THE QUESTIONS, WE DESCRIBED A RESEARCH QUESTION AND TWO SCENARIOS FOR ITS EVALUATION, ONE MAXIMIZING INTERNAL AND THE OTHER EXTERNAL VALIDITY.

RO	Questions	Answer options
1, 2	Which option would you prefer for an evaluation? [We asked this question two times, for human and non-human studies]	○ Max. internal validity, ○ Max. external validity ○ No preference
1	Would it be a reason to reject a paper that does not choose your favorite option?	○ Yes, ○ No
1, 2	In your opinion, what is the ideal way to address research questions like the one outlined above?	Open
1	Did you recommend to reject a paper in the past mainly for the following reasons?	□ Int. validity too low, □ Ext. validity too low
1, 2	For research questions like the one presented above (FP vs. OOP), do you prefer more practically relevant research or more theoretical (ground) research?	○ Applied, ○ Basic, ○ No preference
1	Have you changed how you judged a paper regarding internal and external validity?	○ Yes, ○ No
1, 3	What do you think about a reviewing format with several rounds, but with publication guarantees?	Open
1, 2	Do you have any suggestions on how empirical researchers can solve the dilemma of internal vs. external validity of empirical work in computer science?	Open
3	During your activity as a reviewer, how often have you reviewed a replicated study?	○ Never, ○ Sometimes, ○ Regularly
3	In general, how were the replications rated by you... by your fellow reviewers?	○ Accept, ○ Borderline, ○ Reject
3	During your activity as a reviewer, did you notice a change in the number of replicated studies?	○ Yes, increase, ○ Yes, decrease, ○ No
3	Do you think we need to publish more experimental replications in computer science?	○ Yes, ○ No
3	As a reviewer of a top-ranked conference, would you accept a paper that, as the main contribution,...	
	...exactly replicates a previously published experiment of *the same group*?	○ Yes, ○ No, ○ I do not know
	...exactly replicates a previously published experiment of *another group*?	○ Yes, ○ No, ○ I do not know
	...replicates a previously published experiment of the *same group*, but *increases external* validity?	○ Yes, ○ No, ○ I do not know
	...replicates a previously published experiment of *another group*, but *increases external* validity?	○ Yes, ○ No, ○ I do not know
	...replicates a previously published experiment of the *same group*, but increases *internal* validity?	○ Yes, ○ No, ○ I do not know
	...replicates a previously published experiment of *another group*, but *increases internal* validity?	○ Yes, ○ No, ○ I do not know

We used SurveyGizmo for our survey. In May 2014, we e-mailed each program-committee and editorial-board member, and asked them to complete the survey within three weeks. Of the 807 people we contacted, 94 completed the questionnaire, leading to the typical 10% response rate. Some members preferred to have all questions on one page, so we created an according version for them.

VI. RESULTS AND DISCUSSION

To analyze the answers of the survey, we used an open card-sorting technique [8]. To this end, we looked for higher-order themes in the open answers of participants for each question. Overall, we spent 19 (open questions) $\times$ 2 hours (per question) $=$ 38 hours on categorizing 776 answers. We identified several categories per question, several of them occurred across questions.

Instead of discussing all identified categories, we structure this section along our research objectives. For each objective, we present descriptive statistics of the closed questions (if applicable), followed by a summary of the categories we found with as minimal interpretation as possible (to separate data analysis from interpretation). On the supplementary Web site, we provide all identified categories per question, including their frequency of occurrence. We conclude this section with an interpretation and insights we gained.

A. RO_1: Awareness of the tradeoff between external and internal validity.

For this objective, there are no closed questions, so we directly start with the categories we identified in the free-text responses of the participants.

1) Categories: The responses show a mixed picture. In particular, we found answers indicating that participants are aware of this issue, but also statements lacking this awareness.

Awareness of tradeoff: Participants stated that both kinds of validity should be balanced, which we found 14 times across all questions related to RO_1.

Unawareness of tradeoff: By contrast, we also found a profound lack of awareness regarding the tradeoff. One reviewer stated s/he would reject a paper that describes a study that maximizes internal validity, because it

> "[w]ould show no value at all to SE community".

Another participant stated that his/her opinion regarding the kind of validity changed, such that s/he now can appreciate studies with external validity more, and that s/he has "come to loathe ivory tower toy examples".

Other interesting insights: We also often found that reviewers stated that "it depends" (35) on different aspects, for example, on the research question, on the study subjects, or on the claims, indicating that the kind of validity plays a minor role in judging the merits of a study.

Human and non-human studies: There is a disagreement on whether, for human and non-human studies, the same (6) or different (11) criteria regarding validity should be applied. The reasons for different criteria lie, among others, in the effort of human studies:

> "Non-human experiments are be able to scale up to realistic situations at reasonable cost, in contrast to human experiments.",

they lie in the bias caused by human studies:

> "Removing humans from the exercise reduces the challenges for internal validity. In that context, knowing how general the approach was would seem a more important issue to address.",

or researchers should maximize internal validity for non-human studies (because this is possible in the first place):

> "[...] systems, unlike humans, can be inspected and explained fully. We can produce extremely precise theories about the behavior of software that we create and we should.".

Arguments in favor of applying the same criteria for human and non-human studies arise, among others, from the fact that adoption for industry is the key point of software-engineering research:

> "[...] assess the potential for industrial adoption.";

or that, independent of the kind of study, both kinds of validity are necessary to get a thorough understanding:

> "[...] we need both studies (and possibly more) to get a thorough understanding".

Interestingly, one even stated the equality of human and non-human studies as ground truth:

> "[...] It makes no difference with or without humans! We are talking about software technologies...".

2) Consequences: The magnitude of difference in the opinions surprised us, starting from the view that internally valid studies would have no value to software-engineering research, to the view that only a combination of internally and externally valid studies lets us understand a problem in detail. What can be learned from this result is that researchers should be aware that there *is* a tradeoff and that both kinds of validity add valuable information to our body of knowledge. Furthermore, we would like to point researchers to the fact that there are strong differences in opinions of key players in software-engineering research. If there is no consensus—some might not even be aware of this situation—it is difficult to properly shape the future of software-engineering research. Currently, getting a paper on a study published seems like a game of chance: If authors get a reviewer who is not open to the kind of study that authors report on, chances are that the reviewer will argue strongly against the paper, possibly leading to the rejection of a methodologically sound study.

Generally speaking, there are *no transparent community standards on empirical research*. On the contrary: Different program-committee and editorial-board members have strongly different opinions about internal and external validity without even knowing it. This is partly reflected in the large number of participants stating that the kind of study depends on several factors. One participant even stated that it also depends on the resources of the authors of the paper:

> "...what resources did the authors have? What I expect from a paper out of Cisco is different from a paper out of a university. [...]"

Exaggerating this statement, it could mean that it is ok to recruit students as participants in studies conducted by researchers at universities, because they lack the resources to recruit professionals; however, studies conducted by or in companies, such as Cisco, should recruit professionals, because they have according resources. Clearly, knowing the authors would help in understanding certain tradeoffs regarding resources, but would also prohibit conducting double-blind reviews, which is current practice for several conferences, such as SIGCSE or ECOOP (2014).

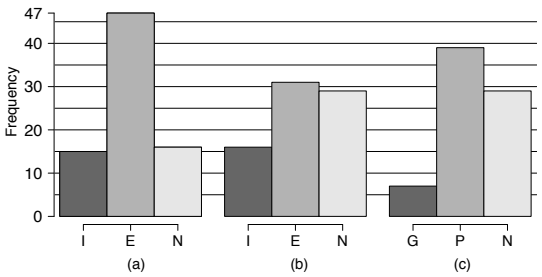

Fig. 3. Frequency distribution of answers. (a)/(b): Which option would you prefer for an evaluation? (a): human studies, (b): non-human studies. [I]nternal validity, [E]xternal validity, [N]o preference. (c): Do you prefer more practically relevant research or more theoretical (ground) research? [G]round research, [P]ractical research, [N]o preference.

Overall, these different opinions show the fundamental need for a community-agreed standard on how to conduct empirical research in software engineering.

Key insights:

- There is a mixed degree of awareness of the tradeoff between internal and external validity.
- The opinions on how to handle the tradeoff differ to a large extent.
- There are different points of view whether the same or different criteria regarding internal and external validity for human and non-human studies should be applied.
- There are no transparent community standards for handling the tradeoff between internal and external validity.

B. RO_2: Opinion of the community regarding how to address this tradeoff.

1) Descriptives: In Figure 3, we show the answers to the three closed questions for RO_2. They indicate a tendency toward externally valid studies with practical relevance.

2) Categories: Again, we got a mixed picture of which questions researchers should ask, but with a clear preference for externally valid studies. We also found that several reviewers prefer balancing internal and external validity. The most important reason to favor external validity is practical relevance:

> "[...] external validity is very important since it provides indications about the potential for industrial adoption."

> "Leave the ivory tower. If actual insights for people's lives are supposed to be the outcome of research, it better be applied to such problems."

> "[...] experience from professional developers seems more relevant."

These and further statements indicate that external validity and practical relevance are seen as equivalent. However, this is not entirely true, as we will discuss shortly.

In addition to focusing on one study, some participants stated that researchers should replicate studies or conduct multiple studies on the same topic, as inspired by other sciences. For example, to declare the discovery of the Higgs-Boson particle, many replications had to be conducted.

3) Consequences:

External validity vs. practical relevance: Several participants equated external validity with practical relevance, leading us to two interpretations:

- First, external validity describes how the results obtained in one experimental setting can be applied to different settings [25], for example, to different programming languages, tasks, or participants. Many answers indicate that a study conducted with professional programmers automatically has higher external validity than a study with students. However, if researchers use professional programmers in their every-day work, the results cannot necessarily be applied to students in a university context. Thus, a practically relevant study does not necessarily have high external validity. Instead, practical relevance is described by the term ecological validity [25]. Admittedly, we might have slightly influenced our participants by the way we asked the questions, as we discuss in Section VIII.
- Second, studies involving students are not seen as practically relevant, because the results are applicable to professionals only to a limited extent. While it is true that much research is conducted to improve the life of the professional programmer, also students (or beginning programmers) are an important population to be studied, especially, when they have considerable programming experience. Furthermore, there are studies showing that, in certain scenarios, students are comparable to professionals [6], [7], [32].

Practical impact of studies: Second, some participants stated that studies should have an immediate practical impact:

> "My preference towards external validity is only slight. I am worried that maximizing internal validity easily creates overly academic papers that provide little impact.[...]".

Thus, a single study is not seen as a piece of the puzzle, but each study needs to immediately lead to general conclusions.

Some reviewers suggested to look at the standards in others sciences, specifically referring to replications being common.

> "[studies in medicine or biology] have hundreds/thousands of participants, over several years, and address very narrow issues (e.g. is medicine X better than Y). We don't see there studies that use 20 participants, are done in 2 months, and attempt to answer questions of the caliber 'is CT better than MRI'.".

Looking at other sciences, it is certainly advisable to get away from the view that a single study must provide a definite answer to a substantial research question. Instead, combining different kinds of studies, for example, a case study to explore hypotheses and controlled experiments to evaluate these hypotheses, is a feasible strategy to address the tradeoff.

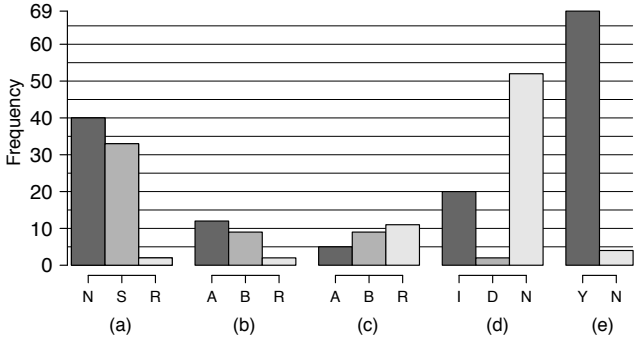

Fig. 4. Frequency distribution of answers. (a): How often have you reviewed a replication? [N]ever, [S]ometimes, [R]egularly. (b)/(c): How were the replications rated... (b): ...by you? (c): ...by others? [A]ccept, [B]orderline, [R]eject. (d): Did you notice a change in the number of replicated studies? I: Increase, [D]ecrease, [N]o (e): Do you think we need to publish more experimental replications in computer science? [Y]es, [N]o.

Key insights:

- There is a misconception of the relation between external validity and practical relevance.
- A single study is not seen as piece of the puzzle, but requires immediate practical impact; this is in contrast to the view that studies provide incremental insights into a complete big picture.
- Replication studies have proved successful in other sciences and should be considered more in software-engineering research.

C. RO_3: Opinion of the community regarding the role of replication

1) Descriptives: In Figure 4, we show the answers to the closed questions regarding RO_3. In essence, many participants think that there are too few replications in our field.

2) Categories: Even though replications are common in other sciences to increase the credibility of results, they are not as accepted in software engineering. For example, some participants stated that there should always be something novel in a study; one even stated:

> "Getting a publication accepted that doesn't contribute anything but a new experiment while assessing the same question (not even adding artifacts) is a good example of hunting for publications just for the sake of publishing. Come on."

However, the majority of the participants stated that we need more replication in software engineering, showing awareness of this issue. They gave several reasons for the need and the lack of replication, as we discuss next.

Delta of a replication: Participants who appreciate replication studies said that replications are useful, as long as they add information to the body of knowledge. However, participants do not agree on what "add information" means. In general, there are many different points of view regarding how to conduct a replication. Many participants say that a replication should add something new or improve an aspect of the original study, for example, not to make the same methodological mistakes again. Some say that a replication should increase

external validity of a study, while others state that internal validity should be increased, or that, at least, the replication has to be done by a different group. So, apparently, there is no agreement on the delta of a replication compared to its original study in the community.

Reasons for lack of replications: The participants mentioned several reasons for why we do not see many replications in software-engineering research, including that they are difficult to publish and that incentives, platforms, guidelines, standards, as well as replication packages are missing:

> "I have seen few replications (and perform myself a few) because they are too difficult to publish: there will always be a (dumb) reviewer to say 'this is not novel!'..."

> "It seems that replication is rarely done since it is costly, hard to do (often not all details, tools, software, or datasets involved in an earlier study are available), and it carries a low-impact factor (at least, in certain venues)."

> "I am not sure though [replication studies] would be appropriate for conferences. A replication study is appropriate for conference if new findings arise. I think that journals are the right outlets for replication studies."

Interestingly, one participant in our survey stated that "[r]eplications are common".

Several rounds of reviewing: One question in our survey was: "What do you think about a reviewing format with several rounds, but with publication guarantees? That is, the paper is guaranteed to be published (independent of the results), if the authors conduct a further, sound empirical evaluation that improves either internal or external validity." We got mixed reactions to this suggestion, some stating that *only* the quality of the conducted study counts:

> "Multiple rounds is a good idea, but approving publication must be based on the quality of the research and presentation. It should not be related to the outcome of the study.",

Others fear a degradation of quality, and that authors will abuse this publication guarantee:

> "Regrettably, my experience is that some authors will undermine this process. It isn't viable."

Nevertheless, while the participants fear that the process will be misused, decreasing research quality, they are not against it in general. Suggestions of our participants to implement this process include providing templates for authors, reviewers, papers, and the process itself.

3) Consequences:

Mismatch: The participants mostly agree that there should be more replication in our field, but they also argued that this is unlikely to happen. A possible resume is that doing or reading a replication is boring and there is no payoff for neither the authors nor the reviewers. It seems that there is a certain hypocrisy in that everybody agrees that replications are important, but not many researchers want to conduct, read, and accept them, as two participants nicely stated:

> "I think that this is a big problem in our discipline. However, in my experience, people are inclined to say that replications are important but then reject replication studies for not presenting "new" problems/questions."

> "I say "yes" [to accepting replication studies] but, like everyone else I know, I wouldn't actually like to do so. So it probably won't happen, even though we pay lip service to it."

Interestingly, we could observe a related pattern in our survey (see Figure 4, (b) and (c)): The number of participants stating that they would accept a replication (b) is as high as the number of participants stating that fellow reviewers tend to reject a replication (c). We see three possible explanations: First, this contradiction might be caused by the possible selection bias in our sample in that mostly participants with an affinity to empirical research responded to our survey, which may tend to accept a replication, while the majority who did not respond tend to reject it. Second, the distribution of answers might indicate a certain mismatch of views or even hypocrisy, in that the participants believe they tend to accept more replication than their fellow reviewers, which, however, might be a biased view. Third, participants have different expectations about how to conduct a replication leading to disagreement in the process of review. In any case, to increase the appreciation of replications, we need to encourage and motivate reviewers to rate them more positively, independently of the novelty of the results.

Incentives for replication: Several participants made suggestions on how to change the current situation to support replication: In essence, we need to create incentives for authors *and* reviewers. Many participants have the impression that authors do not want to do replications, because they expect difficulties getting them accepted. Also, reviewers do not want to review replications, because there is nothing new to learn. Furthermore, reviewing a replication also means more work for reviewers, who would also have to look at the original study to give an informed recommendation about the quality and delta of the replication. However, program-committee and editorial-board work is already a rather ungrateful job, and increasing the workload for reviewers is unlikely to improve the situation. Thus, without incentives for authors and reviewers, it is unlikely that we will see more replications.

A suggestion of some participants is to have a special platform for replication. There is already a workshop series specifically for replication; Replication in Empirical Software Engineering Research (RESER). But, this does not appear to be sufficient, because our participants stated that we need more replication, and workshops typically have limited impact. Furthermore, a designated workshop series gives reviewers the opportunity to reject a replication that does not have novel or contradicting results, based on the argument that it is out of scope and better fits to RESER. This way, unpopular replications become banished from mainstream venues, so that they will still face a niche existence. To make replications more accepted, there needs to be a place for them in renowned conferences and journals. This could mean a special track, issue, or paper categories. However, the community is certainly

well advised to be honest to itself: Who wants to attend a session about replication studies, of which the results may be well known? Thus, accepted replications may face a difficult role in conferences. Some of our participants mentioned that replications should be published in journals, whereas conferences are for presenting novel results. A special track at renowned software-engineering venues, such as ICSE, could raise the awareness for the value of replications.

A second suggestion was that there should be standards or guidelines for reviewers and authors on how to rate replications. For example, for a replication, the methodological soundness should have more influence on acceptance than the novelty of the results (i.e., it would be ok to confirm a previous result). This way, we can counteract the expectation of exciting results:

> "It depends [...] whether the findings contradict the previous ones [...]".

Authors could instead focus on the soundness of the study design, and they should provide replication packages, so that a lack of information does not hinder researchers to replicate a study. One of the participants suggested to consult a member of the original team to conduct the replication, because

> "[...] many details of the experiment are not properly described or not published."

In fact, there is an experience report of a group of researchers who originally planned to replicate a study, but due to the difficulties they encountered (despite conversing with the original team), they could not conduct the replication [20]. Instead, they published their experience with exact replication. Standards on which information to share in which way—also learning from other disciplines—will help authors when replicating other studies.

Third, there are considerable differences in the expectation of the delta a replication must provide. Should a replication study count as such only when conducted by a group different than the original group or only when it adds new information or improve the methodology? Is it enough to change the sample to different students/subjects or the room/daytime of a conduct? We cannot give an answer to these questions based on the survey, and we do not think that there is a general answer, because software engineering incorporates numerous different subfields of different maturity and with different requirements. For example, measuring performance is different than a human-based study on program comprehension. Thus, each subcommunity needs to define its own standards, which need to be communicated clearly to the authors and reviewers.

Finally, our suggestion of multiple reviewing rounds with publication guarantees—given sufficient quality of a study—received mixed reactions, mostly because of the fear of degrading the quality of research and of undermining the whole process. With a well-defined review and publication process, we might mitigate this problem, as one participant stated:

> "Sounds interesting but has to be outlined and studied in detail."

Key insights:

- There is a certain mismatch in the participants' view on replication studies: Most participants appreciate replications, but see that they are hard to conduct and publish.
- Neither researchers nor reviewers seem to like to conduct, read, or accept replications.
- There is disagreement on the delta a replication must provide.
- Suggestions to improve the situation include setting up special platforms and guidelines for reviewers and authors, which need to be defined and communicated by the respective subcommunity.

VII. Further Insights

In addition to answering our research objectives, we gained several further insights we want to share with the community.

A. Paper = Experiment?

An interesting point that came up across all questions was whether a study should map 1-to-1 to a paper, or whether there should be an n-to-m mapping, in particular, multiple (replication) studies making up a single, substantial paper. During our analysis, we learned that we and others tend to think of a study and a paper as interchangeable concepts. However, is this really the way to go? One participant asked:

> "Excuse me, but are we discussing science and the way it should be done, or how to prepare papers to be accepted?"

This issue indicates that empirical research in software engineering has come to a point that, when designing a study, researchers also think in terms of getting a corresponding paper published. But this is not just a problem in software engineering, but in many more disciplines, as the slogan "publish or perish" describes. A possible solution to this dilemma is exercised by PLOS ONE (http://www.plosone.org/static/publication), in which the evaluation of the worthiness of a result is left to the reader; the purpose of the review process is quality assurance, such that the conclusions drawn from a study are justified.

B. Internal/External Validity vs. Artificiality/Practicality

There seems to be a misconception about the relationship of validity and practicality. We believe that the reasons lie in the close relationship of both concepts: An internally valid study is only rarely realistic, because many confounding parameters need to be controlled for, which easily results in artificial experiment settings. Externally valid studies often are realistic, because the lack of control for confounding parameters can lead to several different values for them (e.g., novice to expert programmers). However, internally valid studies can also be realistic and produce generalizable results, for example, if the selected programming language shares similar properties with other often-used programming languages. Thus, we should avoid equating external validity with practicality, and internal validity with artificiality: If internally valid studies are only seen as artificial, toy examples, or ivory-tower research, it is hard to raise the appreciation for internally valid studies, which

are an important way to understand effects in depth. Likewise, if only externally valid studies are accepted, how can we ever pinpoint the precise factors causing the observed effect?

C. Software Engineering = Engineering Discipline?

We were surprised by the number of answers stating that participants expect a practical impact of each study, because software engineering is an engineering discipline rather than science—practically relevant studies are inherent to software engineering. However, there is no reason for why software-engineering research should not follow standards of natural science, where internally valid studies can add valuable information to our knowledge base. Maybe the view of software engineering "solely" as engineering discipline (which is still discussed [26]) is one reason for the lack of appreciation of internally valid and replication studies?

D. Empirical Research not for its Own Sake

Several participants expressed their concern not to do empirical research only for its own sake. In a world where publishing papers decides over careers, there is certainly the danger that people start conducting (replication) studies just for increasing their publication count. Given that replication studies will be more accepted in the future, one could imagine that it is quite easy to grab "low-hanging fruit" by replicating existing studies. Which degree of replication is healthy? In some sense, we trade confidence in our results by conducting replication studies with the danger of being swamped by studies that have been conducted only of their own sake.

VIII. Threats to Validity

A. Internal Validity

There is a possible selection bias, as it may be that only those program-committee and editorial-board members responded who have experience with empirical research. This could mean that there is a bias toward the awareness of the tradeoff between internal and external validity and the appreciation for internally valid and replication studies. Thus, one could assume that the software-engineering community as whole is less aware and appreciative of these issues. While the selection bias is relevant, it does not affect the big picture that there are many different opinions, about which the researchers are not necessarily aware.

Another threat arises from the Rosenthal effect [24]: The wording of the questions might have influenced the participants, for example, regarding the misconception of external/internal validity vs. artificiality/practicality, and the relation between studies and papers (1:1 or n:m). Hence, both insights might not follow to this extent from our sample, but we believe they would have occurred anyway: Only one participant mentioned that external validity and practicality are not the same, and one other stated that we should not design studies to get papers accepted.

B. External Validity

Reflecting on the insight about the mapping from studies to papers, we revisited our design decisions for the survey. Admittedly, we also thought about how these decisions would affect acceptance chances, especially, contacting only program-committee and editorial-board members, which threatens external validity. We cannot say whether the big picture would change when including further researchers. But, as our goal was to get insights from the "key players", we sufficiently controlled this threat with respect to the scope of our study.

IX. Conclusion

As empirical research has grown common in software engineering, it is time to agree on how it should be conducted and how to address the tradeoff between internal and external validity and replications. Reviewing papers that were recently published in major software-engineering venues, we found that 91 % presented an empirical study, but only 54 % discussed threats to validity, and only 23 % differentiated between different kinds of validity. Given that we include EMSE as major *empirical* software-engineering journal, this is an alarmingly high number of authors who do not seem to be aware of the threats to validity to their study.

To get a deeper understanding of the view of the community's "key players", we asked program-committee and editorial-board members of major software-engineering venues about their opinions on these and related issues. We found that many reviewers are *not aware* of the tradeoff between internal and external validity, but at the same time have strong opinions on *maximizing one kind of validity*, which indicates a *lack of community standards* on conducting and reviewing empirical studies. This leads to the situation that getting a paper accepted is a game of chance rather than based on quality or value added to the community. Interestingly, a considerable number of participants stated that *only externally valid studies*, best with *immediate practical impact*, have value.

Regarding the role of replication, we also found a mismatch: Most participants wish to see *more replications* but, at the same time, are *reluctant to conduct, read, or accept* them. Apparently, in software engineering, there is a *lack of incentives* for conducting replication studies (e.g., low impact, low acceptance chance, high effort) and a *lack of standards* on how to design and review replications (in particular, on the delta of a replication). Thus, software engineering does not seem to be comparable to other engineering or social disciplines. So, we must ask ourselves: How can we shape and promote empirical software engineering if we cannot agree on what it should like? Having made these different points of view explicit, we hope that they initiate a discussion in the community and provide a starting point for guidelines and standards of empirical software engineering, both for authors and reviewers.

Finally, we would like to stress that our goal is not to judge or offend any reviewers or authors. On the contrary, we highly appreciate the time and effort the participants took to answer our questions, which documents their interest in this issue.

Acknowledgments: Thanks to Lutz Prechelt for fruitful comments on this paper. Also thanks to all program-committee and editorial-board members who shared their opinion with us. This work has been supported by the DFG grants AP 206/4, AP 206/5, and AP 206/6.

REFERENCES

[1] V. Basili. Software Modeling and Measurement: The Goal/Question/Metric Paradigm. Technical Report CS-TR-2956 (UMIACS-TR-92-96), University of Maryland at College Park, 1992.

[2] V. Basili, F. Shull, and F. Lanubile. Building Knowledge through Families of Experiments. *IEEE Trans. Softw. Eng.*, 25(4):456–473, 1999.

[3] V. R. Basili, R. W. Selly, and D. Hutchens. Experimentation in Software Engineering. *IEEE Trans. Softw. Eng.*, 12(7):733–743, 1986.

[4] D. Budgen, B. A. Kitchenham, S. M. Charters, M. Turner, P. Brereton, and S. G. Linkman. Presenting Software Engineering Results Using Structured Abstracts: A Randomised Experiment. *Empirical Softw. Eng.*, 13(4):435–468, 2008.

[5] T. Dybå, V. B. Kampenes, and D. Sjøberg. A Systematic Review of Statistical Power in software Engineering Experiments. *J. Information and Software Technology*, 48(8):745–755, 2006.

[6] S. Hanenberg. An Experiment about Static and Dynamic Type Sytems: Doubts about the Positive Impact of Static Type Systems on Development Time. In *Proc. Int'l Conf. Object-Oriented Programming, Systems, Languages and Applications (OOPSLA)*, pages 22–35. ACM Press, 2010.

[7] M. Höst, B. Regnell, and C. Wohlin. Using Students as Subjects: A Comparative Study of Students and Professionals in Lead-Time Impact Assessment. *Empirical Softw. Eng.*, 5(3):201–214, 2000.

[8] W. Hudson. Card Sorting. In *Guide to Advanced Empirical Software Engineering*. The Interaction Design Foundation, 2013.

[9] M. Ivarsson and T. Gorschek. A Method for Evaluating Rigor and Industrial Relevance of Technology Evaluations. *Empirical Softw. Eng.*, 16(3):365–395, 2011.

[10] A. Jedlitschka, M. Ciolkowski, and D. Pfahl. Reporting Experiments in Software Engineering. In *Guide to Advanced Empirical Software Engineering*, pages 201–228. Springer, 2008.

[11] A. Jedlitschka and D. Pfahl. Reporting Guidelines for Controlled Experiments in Software Engineering. In *Int'l Symposium Empirical Software Engineering (ISESE)*, pages 95–104. IEEE CS, 2005.

[12] N. Juristo and A. Moreno. *Basics of Software Engineering Experimentation*. Kluwer, 2001.

[13] N. Juristo and S. Vegas. The Role of Non-exact Replications in Software Engineering Experiments. *Empirical Softw. Eng.*, 16(3):295–324, 2011.

[14] V. Kampenes, T. Dybå, J. Hannay, and D. Sjøberg. A Systematic Review of Quasi-Experiments in Software Engineering. *Information and Software Technology*, 51(1):71–82, 2009.

[15] B. Kitchenham, H. Al-Khilidar, M. A. Babar, M. Berry, K. Cox, J. Keung, F. Kurniawati, M. Staples, H. Zhang, and L. Zhu. Evaluating Guidelines for Reporting Empirical Software Engineering Studies. *Empirical Softw. Eng.*, 13(1):97–121, 2008.

[16] B. Kitchenham, P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman. Systematic Literature Reviews in Software Engineering: A Systematic Literature Review. *J. Information and Software Technology*, 51(1):7–15, 2009.

[17] B. Kitchenham, P. Brereton, Z. Li, D. Budgen, and A. Burn. Repeatability of Systematic Literature Reviews. In *Proc. Int'l Conf. Evaluation and Assessment in Software Engineering (EASE)*, pages 46–55. IET Software, 2011.

[18] B. Kitchenham and S. Charters. Guidelines for Performing Systematic Literature Reviews in Software Engineering. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report, 2007.

[19] A. Ko, T. LaToza, and M. Burnett. A Practical Guide to Controlled Experiments of Software Engineering Tools with Human Participants. *Empirical Softw. Eng.*, pages 1382–3256, 2013. Online first.

[20] J. Lung, J. Aranda, and S. Easterbrook. On the Difficulty of Replicating Human Subjects Studies in Software Engineering. In *Proc. Int'l Conf. Software Engineering (ICSE)*, pages 191–200. ACM Press, 2008.

[21] M. Nagappan, T. Zimmermann, and C. Bird. Diversity in Software Engineering Research. In *Proc. Europ. Software Engineering Conf./Foundations of Software Engineering (ESEC/FSE)*, pages 466–476. ACM Press, 2013.

[22] D. Parnas. Point: Empirical Research in Software Engineering: A Critical View. *IEEE Software*, 26(6):56–59, 2009.

[23] T. Roehm, R. Tiarks, R. Koschke, and W. Maalej. How Do Professional Developers Comprehend Software? In *Proc. Int'l Conf. Software Engineering (ICSE)*, pages 255–265. IEEE CS, 2012.

[24] R. Rosenthal and L. Jacobson. Teachers' Expectancies: Determinants of Pupils' IQ Gains. *Psychological Reports*, 19(1):115–118, 1966.

[25] W. Shadish, T. Cook, and D. Campbell. *Experimental and Quasi-Experimental Designs for Generalized Causal Inference*. Houghton Mifflin Company, 2002.

[26] M. Shaw. Research Toward an Engineering Discipline for Software. In *Proc. Int'l FSE/SDP Workshop on Future of Software Engineering Research (FoSER)*, pages 337–342. ACM Press, 2010.

[27] F. Shull, J. Carver, S. Vegas, and N. Juristo. The Role of Replications in Empirical Software Engineering. *Empirical Softw. Eng.*, 13(2):211–218, 2008.

[28] J. Siegmund and J. Schumann. Confounding Parameters on Program Comprehension: A Literature Survey. *Empirical Softw. Eng.*, pages 1–34, 2014. Online first.

[29] D. Sjøberg, B. Anda, E. Arisholm, T. Dybå, M. Jørgensen, A. Karahasanovic, E. F. Koren, and M. Vokác. Conducting Realistic Experiments in Software Engineering. In *Proc. Int'l Symposium Empirical Software Engineering and Measurement (ESEM)*, pages 17–26. IEEE CS, 2002.

[30] D. Sjøberg, T. Dybå, and M. Jørgensen. The Future of Empirical Methods in Software Engineering Research. In *Future of Software Engineering*, pages 358–378. IEEE CS, 2007.

[31] D. Sjøberg, J. Hannay, O. Hansen, V. B. Kampenes, A. Karahasanovic, N.-K. Liborg, and A. Rekdal. A Survey of Controlled Experiments in Software Engineering. *IEEE Trans. Softw. Eng.*, 31(9):733–753, 2005.

[32] M. Svahnberg, A. Aurum, and C. Wohlin. Using Students as Subjects: An Empirical Evaluation. In *Proc. Int'l Symposium Empirical Software Engineering and Measurement (ESEM)*, pages 288–290. ACM Press, 2008.

[33] W. Tichy. Hints for Reviewing Empirical Work in Software Engineering. *Empirical Softw. Eng.*, 5(4):309–312, 2000.

[34] W. Tichy, P. Lukowicz, L. Prechelt, and E. Heinz. Experimental Evaluation in Computer Science: A Quantitative Study. *Journal of Systems and Software*, 28(1):9–18, 1995.

[35] W. F. Tichy. Should Computer Scientists Experiment More? *Computer*, 31(5):32–40, 1998.

[36] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in Software Engineering: An Introduction*. Kluwer Academic Publishers, 2000.

[37] C. Zannier, G. Melnik, and F. Maurer. On the Success of Empirical Studies in the International Conference on Software Engineering. In *Proc. Int'l Conf. Software Engineering (ICSE)*, pages 341–350. ACM Press, 2006.