Difficulty Factors of Obtaining Access for Empirical Studies in Industry

Lutz Prechelt, Franz Zieris, Holger Schmeisky
Freie Universität Berlin, 14195 Berlin, Germany
{prechelt,zieris}@inf.fu-berlin.de, holger.s@fu-berlin.de

***Abstract*—*Context:* The difficulty (not just effort) of obtaining access for software engineering empirical studies in industry varies greatly. Supposedly, some of this variance in difficulty is particular, stemming from properties of individual contexts (the industrial partners and their work), while the rest is repeatable, related to properties of the research question and research design. *Question:* What are these recurring difficulty factors that arise from research question and research design? What mechanisms produce their influence? *Method:* We use ideation and knowledge extraction from research experience to identify potential difficulty factors, use expert discussion to understand their mechanisms, and use concept analysis to arrange them into a taxonomy. We evaluate the result by comparatively applying it to two research efforts pursued by the same research group. *Results:* We find six scope factors, five problematic intervention effects factors, and seven helpful intervention (side-)effects factors. *Conclusion:* Considering these factors systematically during the formulation of a research question and the design of a research method can help with balancing data collection difficulty with results validity and relevance.**

I. Introduction

After a few decades of empirical work in software engineering there is now a literature on what types and strengths of evidence to expect from what types of research method [4], on how to design specific types of study (such as controlled experiments [19], surveys [5], interview studies [16, Sec. 2.2], or case studies [13]), on qualitative data analysis [16, Sec. 3], and even specifically on empirical work in industrial settings [1], [2].

The focus of these methodological contributions is usually on how to design a study such that the effort for carrying it out remains bearable and such that the results will be valid and relevant, how to avoid pitfalls during the study's execution, or how to perform the data analysis. However, for studies in industrial contexts, in between the preparation of even the most sound and efficient research design and the most brilliantly flawless execution of this design, researchers need to obtain access to a suitable industrial context and need to be allowed to perform their data collection – and enough of it.

Many of the articles that have appeared in the two previous instances of the CESI workshop talk about some aspects of this problem, but they usually do this on a specific level: for a particular case or at least a particular research topic [7], [9], [11]. The present article discusses this problem in a generic fashion.

A. *Research questions*

- What are the factors that influence the difficulty of obtaining access to a suitable industrial context for a study and that arise from the study's research question and research design (as opposed to individual properties of the researchers or of the industrial partners)?
- What mechanisms are responsible for their influence?

B. *Terminology*

Difficulty in this sense is not to be confused with effort. By difficulty we mean the amount of ingeniousness (to adjust the research design), quick wit (to understand the industrial partner's constraints within the short time they will be willing to explain them), relationship-building talent (to gain the partner's trust), and luck (for encountering the right partner at the right time in the first place) that is required for obtaining access. High difficulty will mean high effort, but high effort alone cannot guarantee overcoming the difficulty, because it can not replace the ingeniousness, quick wit, or relationship-building talent, only the luck.

C. *Related work*

Lethbridge et al. [10] present a set of benefits, costs, and risks related to university-industry collaborations. They consider both parties of such collaborations, and all factors (controllable or not) of the whole time span, that is, from negotiation over execution to analysis and publication. The goal of our work, in contrast, is to inform the research design phase in order to reduce the amount of ingeniousness required and hence the practical difficulty. We are neither concerned with procedures for actually executing the study nor with design techniques for maximizing its validity or relevance. Thus, in comparison with the work of Lethbridge et al., there are
(a) factors we do not consider, because they are independent from research question and design (such as their "indirect benefits" for the company to recruit employees from the research staff);
(b) factors that are equivalent, but named differently (such as their cost "consumption of employee time" and our scope factor *practitioner effort* described below);
(c) generic factors (such as their risk of "unknown consumption of employee time") which we split up into several (in this case the scope factor *time extent* and the problematic intervention effects of *distraction*, *complication*, and *need to learn*); and finally,
(d) factors that are completely new such as our *required technology*.

Wohlin et al. [18] are also concerned with the collaboration as a whole. They surveyed university researchers and industry representatives experienced in industry-academia collaboration and asked them to rank 14 success factors. A few factors, such as "buy-in and support from company management" and having "a collaboration champion on site", were identified as crucial. So compared to existing work our contribution arises from our sharp focus on formulating the research question and designing the research method.

D. Structure of this article

We will now explain our research method, including the limitations that arise from it (Section II) and the difficulty factors we found (scope factors in Section III, problematic intervention effects in Section IV, helpful intervention effects in Sections V and VI.). We go on to evaluate the factors by applying them to two rather different attempts at industrial studies (Section VII) and conclude (Section VIII).

II. Method, Limitations

To arrive at the results described below, we have not applied a formalized research procedure; nor have we worked through any large volume of formalized data. Rather, in step 1 we have relied on sourcing initial factor ideas from the research experience of the three authors plus a fourth empirical researcher. As step 2, we have then discussed these ideas to make sure they actually fall into the scope of this article (or identify the aspect of them that does) and to obtain consensus on the mechanisms that are at work for each of them to make it into a source of difficulty.

During these discussions plus the subsequent writing process we also sorted them into the categories described below such as to form a taxonomy (step 3). As step 4, we somewhat validated our results by applying them to two ongoing streams of research of ours, looking whether each major experience of high or low difficulty we had in each would be (a) addressed at all and (b) properly explained and predicted by a difficulty factor from the taxonomy.

This style of results production (steps 1 to 3) cannot harvest any of its credibility from formal sources such as a known-to-work research method; its initial credibility depends entirely on convincing argumentation which in turn will have to appeal to the readers' own experience with industrial empirical studies. This can (and hopefully does) work well for the validity of the results, but there is no way to ensure their completeness and indeed the reader should expect that a few factors may be missing.

Our case examples (step 4) can strengthen the credibility somewhat with respect to validity, but are obviously very limited in breadth, again pointing to potential incompleteness of our set of factors.

The factors could easily be split into many more factors or aggregated into many fewer. We have settled for what we perceive as a medium level of granularity; it likely reflects our main research interest which is human-related (as opposed to technology-focused) aspects of the software process (as opposed to software products).

III. Factors Category 1: Scope

We will present each difficulty factor in turn, each in a separate subsection. That subsection will define the factor and shortly explain the mechanisms that underly it, that is, *how* adjustments in the research question and/or design affect the difficulty of obtaining access. Many of the factors correlate somewhat, even across categories; do not expect them to be orthogonal – we are interested in their useful differences not in their similarities. Important: All factors use the perspective of the industrial organization[1], not the perspective of the researcher. In particular, all effort discussed is practitioner effort, not researcher effort.

TABLE I. Overview

1 Scope
- Practitioner Effort
- Loss of Confidentiality
- Required Technology
- Number of Participants
- Diversity of Roles
- Time Extent

2 Intervention Effects

A Problematic Intervention Effects
- Schedule Risk
- Quality Risk
- Distraction
- Complication
- Need to Learn

B Helpful Intervention Side-Effects
- Action Research Mode
- Must-pay-for Activity

C Helpful Intervention Effects
- Capability Improvement Expected
- Tooling Improvement Expected
- Insights Expected
- Image Benefits
- Altruistic Benefits

The notion of *scope* comprises issues having to do with the breadth and/or volume of obstacles that will have to be overcome to execute the study. Note that for obtaining access, expected scope is more important than actual scope.

A. Practitioner Effort

The most straightforward and ubiquitous scope-type difficulty is the amount of work time (person hours) the organization expects to invest into participating in the study.

B. Loss of Confidentiality

The amount of information that the organization perceives to be confidential and that it expects will be exposed to the researchers and/or the scientific public, as well as the degree of confidentiality assigned to this information represents a scope-type difficulty.

The list of potentially confidential information is long: product elements, product status information (metrics), process data (bug tracker etc.), aspects of organizational structure or culture, etc. Potential partners may not be open with respect to what they consider confidential. Research designs that do not require planned exposition of confidential information and provide mechanisms for minimizing unplanned exposition will more likely lead to access.

[1] at whatever level is relevant: company, division, department, team, or any other structure.

C. Required Technology

The number of technological elements the study requires the organization to have in place and in use, e.g. a particular programming language, IDE, or version management system. Every factor for which a study design manages to accommodate two or more choices for any such factor will increase the likelihood of access. This is the only difficulty factor that tends to represent an unsurmountable (rather than gradual) obstacle.

D. Number of Participants

The number of different people that need to participate in the study. It tends to be much harder[2] to get (at one organization) 12 people to participate 3 hours each than to get 3 people to participate 12 hours each, because these people need to be found, their concerns addressed, and their time constraints accommodated.

E. Diversity of Roles

The number of different types of people that need to participate in the study. It tends to be much harder to obtain participation from 4 developers, 2 testers, 1 architect, and 1 team lead than it is to obtain participation from 8 developers, because convincing them that participation is worthwhile works differently for each type.

This may not be true if they happen to form a team that decides participation at the team level. In this case, team diversity takes the place of role diversity. Note that a team assigned to the study by some boss (as opposed to deciding themselves) will not remove the role diversity difficulty, because without convincing the members data collection may not actually work out.

F. Time Extent

The number of calendar days between study start and study end, including days (possibly an overwhelming majority of days) of waiting time on which no data collection is intended. Organizations know that the longer the time frame, the more (unexpected) events may intervene to wreck the original collaboration plans.

Study designs that manage to compress the overall time frame have higher likelihood of obtaining access. A particularly nasty case of long time extent is the need to observe rare events that only occur unforeseeably.

IV. Factors Category 2a: Problematic Intervention Effects

Any empirical study performed in an industrial organization may influence that organization's work somehow, whether that happens now or later or both. Except for the ones covered by *scope* above, we call such influences *intervention effects*, even for studies that do not aim at intervention at all. Note that again for obtaining access we are concerned with expected effects, not actual effects.

Negative expectations are discussed here, positive expectations will be discussed in Sections V and VI.

[2]The opposite may be true for very low-effort studies such as 5-minute interviews.

A. Schedule Risk

The organization may expect that the study will or may inflict unforeseeable effort that may wreck a project schedule. Study designs that allow the organization to withdraw from the study quickly and without losing work in case of such problems will have a better likelihood of overcoming this difficulty.

B. Quality Risk

The organization may expect that the study will or may damage the quality of the organization's software product. The more any negative quality impact that the study's intervention may have will be immediately obvious once it happens and easy to fix, the better the chance the study design will overcome this difficulty.

C. Distraction

The organization may expect that the study will distract its practitioners from their normal work *beside* the explicit participation effort. For instance, there may be context switching losses if the study involves interrupting practitioners in the middle of work. Also, the presence of observers or the activity of other data collection mechanisms may produce irritation, or practitioners may develop feelings of embarrassment for some of their normal behaviors. Studies that ensure a high degree of voluntariness of *all* participants and a high degree of informedness about research procedures will have a better likelihood of overcoming this difficulty.

D. Complication

The organization may expect that the study will require steps that are outside normal work routines *and* outside the core aspects of participating in the study. Even if such steps are neither difficult nor time-consuming they may still be perceived as complications and thus represent a substantial motivational obstacle.

Examples might be the need to obtain permission from higher management for deviating from a standardized work routine or the need to collaborate with system administrators for accommodating requirements of the tools applied in the study. Study designs that are flexible enough to expressedly work around such steps if and when they are perceived as complications will have a better likelihood of overcoming this difficulty.

E. Need to Learn

If a study involves introducing a new tool or method, practitioners will typically need to understand these and learn how to work with them first before productive work can begin. This remains true even if the research question is more interested in the learning process than the later usage process.

If organizations expect a *need to learn* they are usually correct, yet their estimate of the learning *effort* (time as well as motivation) may be pessimistic. So bringing evidence for actual learning effort and possible learning enjoyment may help overcoming this difficulty.

V. Factors Category 2b: Helpful Intervention Side-Effects

To have any hope of admission at all, a study will usually need to promise at least some sort of benefit to the organization or some of its inhabitants (see Section VI). In order to present such a promise believably, the researchers will need to possess at least a basic level of "street credibility": The expectation that they are not only capable researchers, but also have an at least roughly realistic idea of how software engineering practice works. We call this the *competency assumption*.

The present section presents one study design attribute that will reduce the required level of assumed competence and another that can strengthen the competence assumption. Section VI describes the types of benefit a study design may promise once the competence assumption is given.

A. Action Research Mode

If the study involves action research, that is, the *joint* solving of a problem posed by the organization [3], the organization will be much more confident to have enough control to ensure that the activities performed during the study fit with their constraints. This confidence reduces the level of competence the organization needs to assume of the researchers before they are willing to provide access: With action research, it is sufficient that the researchers are capable researchers and *able to understand* issues of practice when those are explained to them. They need not know and understand those issues in advance.

Researchers should be aware that there is no sharp boundary between predesigned studies and action research. Even if they do not intend to present their research as action research afterwards, explicitly introducing action research elements in their design when presenting it to an organization may help with obtaining access.

B. Must-pay-for Activity

If the study design involves a researcher activity of a type companies are used to pay for (typically training, possibly consulting), the mere fact of *having* a price can serve as a proxy for service quality [20] and will therefore help leaping the competency assumption hurdle.

VI. Factors Category 2c: Helpful Intervention Effects

Once the competency assumption has been established, the following categories of benefit to the organization should be considered as possible parts of the marketing story for the study design. These are anti-difficulties and need to outweigh the difficulties.

A. Capability Improvement Expected

Constructive studies usually involve interventions that promise some improvement of engineering capabilities. Study designs aiming at quantitative measurement of benefits are easier to get access for in this respect than exploratory studies (but the latter tend to be much nicer in terms of the scope difficulties). Ideally, there is even prior quantitative evidence of the benefits, however weak.

B. Tooling Improvement Expected

This is a special case of the former: The simplification of work steps that are performed anyway. It tends to be more convincing for technology-oriented partners, because the improvement is easier to imagine.

C. Insights Expected

The benefits expectation may be unspecific: "We are not sure what it will be, but we expect to learn *something* beneficial." In this case, the benefit arises from insights and those insights might be expected to arise from the study execution, the study results (perhaps only in the form of the eventual article, which may include results from other organizations as well), or both. The insight or its usefulness might be expected by individuals only or by the organization. Researchers should work to explain their study design such that they emphasize various kinds of possible insights in order to apply this helpful effect to their advantage.

D. Image Benefits

Beside the engineering benefits, companies may also find it attractive to be named as research partners in research publications because they perceive this to be positive for their public image, e.g. with respect to hiring young talent.

E. Altruistic Benefits

The study results will not only help the organization, they will also help the rest of the world. This altruistic motive is a valid argument for many individuals as well as some fraction of organizations at least on the team level.

VII. Case examples

In the present section we will analyze two research efforts of our work group in terms of the difficulty factors described above. We will compare the thusly-expected difficulty between the two efforts and compare it to the actual sequence of events with respect to obtaining access to industrial work contexts.

This serves two purposes: First (and foremost) to validate that difficulty effects actually exist as postulated; second to illustrate the meaning of the factors some more.

A. Case 1: Pair Programming (PP)

Pair programming (PP) is the practice of having two people work jointly on one programming problem in close collaboration using only one computer. Pair programming involves a lot of verbal dialog, which makes it nicely investigatable [6].

Our research aims at finding out (a) what behaviors pair programmers use to steer the process and (b) which behaviors work well or not so well and why [15].

The research approach is Grounded Theory Methodology [17]: We record (as desktop video, talking heads web cam video, and audio) complete PP sessions (typically between one and three hours long) of natural pair work (the real task of that day, the pair's own decision to use PP). We analyze and compare these recordings in depth at home over many months, conceptualizing the behaviors we see. We also offer a reflection interview held the day after a recording where we learn about the participants' perception of the session and the pair receives feedback from us to reflect on their work. We rarely record more than two sessions of the same pair (often only one) and strive for a diverse collection of recordings involving many companies, application domains, technological domains, task types, pairs, developer skill levels, skill level pairings, PP-skill levels, and so on.

B. Case 2: Agile Offsharing (AgOg)

Agile Offsharing (AgOg) is a process idea for reducing the pain of distributed software development: It is difficult to get distributed sub-teams to feel and work as one joint team. Further, if there is a "home" team that has access to the application domain, the users, and hence the requirements, and a "remote" team that does not, avoiding requirements misunderstandings is a major difficulty. Agile Offsharing attempts to solve both of these problems at once by introducing distributed pair programming (with one home and one remote participant) into the process as a regular (if not pervasive) practice. Pairs and tasks are selected such that joint requirements discussion is maximized [12].

Our research aims at (a) determining the details of how to put the AgOg idea into practice and (b) finding out how well it works for which aspects of the problem.

The research approach is Action Research [3]: The work is a joint problem solving process using a semi-structured hypothesis-experiment-evaluation-decision cycle. The researchers' main roles are supporting ideation and decision-making and performing the data collection for the evaluation.

C. Expected difficulty: PP

The PP research ought to have low difficulty of getting access (per company) because of the fine granularity and the lack of intervention: Only two people in only one role (small *number of participants*, low *diversity of roles*) are required at a time; their additional effort is only about 4 person hours total (small *practitioner effort*) over only two days (small *time extent*). The study is technology-neutral (no *required technology*) and can usually avoid confidential areas easily (no *loss of confidentiality*).

Schedule risk, *quality risk*, and *need to learn* are non-issues, because the pair simply does what it would have done anyway. The only relevant obstacles are the pair members' willingness to be scrutinized (some *distraction*) and the need to ask the boss for permission (possible *complication*).

We can explain to the practitioners why they should expect the reflection interviews to produce *insights* or even lead to *capability improvements*.

D. Expected difficulty: AgOg

The AgOg research ought to be a lot harder: It occurs at the team level and so has a higher *number of participants* from *diverse roles*. It concerns a project and so has a *time extent* of months and a lot more *confidential* items may become known to the researchers. Action research requires heavy involvement of the practitioners and so may create at least the impression of a substantial *practitioner effort*. The study depends on the use of editors/IDEs for which a suitable tool[3] for distributed pair programming (DPP) is available (a *required technology*). Both the tool and DPP itself *need to be learned*.

The whole arrangement could be suspected of side-effects that pose *schedule risk* and/or *quality risk*. The presence of researchers and the unusual work activities can be seen as *distracting* and the severe intervention that AgOg requires produces multiple *complications*, including the sysadmin type, the boss type, the legal department type (contract issues with the offshoring partner) and possibly others.

On the positive side, the intended *action research mode* of the study should by itself lower the required level of researchers' *compentence assumed* by the industry partner, produce the expectation of *insights* along the way, and the expectation of *capability* and *tooling improvements*.

E. Actual difficulty: PP

The getting-admission difficulty for the PP research is indeed low[4]: We went to a single practitioner conference, presented what we wanted to do, and immediately found several interested developers. Over the course of several weeks, these contacts resulted in 22 recordings at 4 different companies.

Pair partner unwillingness to be recorded (a *distraction* factor) and *loss of confidentiality* fears sometimes get in the way. But since the *scope* is narrow, potential partners are easier to find, so we can skip one and work with the next. Our collection of session recordings has been growing steadily since. Recorded pairs often report that they find the reflection session highly useful, which has strengthened the factor of *expected insights* when approaching new partners. What a difference to the getting-admission difficulty for the AgOg research!

F. Actual difficulty: AgOg

The getting-admission difficulty for the AgOg research is indeed very high. Here is the story: We have been investing in building an industrial-strength DPP tool, Saros [14], since 2006 and most of our contacts find it interesting and trustworthy and expect related benefits (*capability improvement* and sometimes *tooling improvement*).

When we present the AgOg idea at practitioner conferences, we regularly obtain a handful of contacts who find it plausible and applicable to their context (*capability improvement expected*) and none of them has ever voiced *quality risk* concerns or substantial *distraction* concerns. One such contact even proceeded to sponsor a full-time developer for six months to develop a version of Saros for IntelliJ IDEA in order to remove the dependency on Eclipse which he found unacceptable (thus alleviating one *required technology* factor). We have twice been invited (by managers) to give company-internal presentations of AgOg in order to find concrete research partner projects.

Nevertheless, the broad scope of an AgOg study and the massive process intervention it incurs have foiled all candidate

[3]If the pair members use different editors, the tool must even be interoperable. Screen sharing is insufficient.

[4]The data analysis difficulty is very high, though.

collaborations so far: Some potential partners lost interest for reasons such as somebody having doubts regarding the general efficiency of pair programming, considering it a *schedule risk* (this is often a by-product of the required *diversity of roles*). For others, a weird aspect of the candidate's prescribed infrastructure broke Saros' usefulness (aspect of the *required technology*), a minor usage mistake when trying out Saros gave the tool a bad image (*tooling improvement expected* not working out), or the remote team shied away (because of the expected researcher induced *distraction* in the form of irritation and embarrassment, or the *loss of confidentiality*, we are not always sure which).

Several other contacts are still interested but the heavy bulk of an AgOg intervention has made them postpone and postpone again (high *number* and *role diversity of participants*, the *required technology*, and the *practitioner effort*) although one of these contacts had already decided to want us to hold a paid AgOg workshop (a *must-pay-for activity*). Since these partners' *competence assumption* is not a problem, the *action research* nature of the AgOg work has so far not helped.

As these descriptions show, the difficulty factors predict the different levels of empirical getting-access difficulty well. Also, one of the anonymous reviewers of this article applied the factors to a study s/he had been doing and wrote "I [...] found that [the] factors proposed in the paper allow evaluating my case and, somehow, predicting the acceptance to run the experiment we proposed."

VIII. Conclusion and Further Work

We have presented aspects of research designs that raise or reduce the difficulty of being admitted by an industrial partner to perform a study with them (difficulty factors). We found six "scope" factors that revolve around the breadth or volume of the steps the partner expects to have to do before and during the study. We found five "problematic intervention effects" factors that revolve around risks and complications the partner expects to perhaps arise from attempting the study. We found two "helpful intervention side-effects" factors that help establishing the expectation of a sufficient level of researcher competence and five "helpful intervention effects" factors that revolve around the benefit the partner may ascribe to the study and that constitute anti-difficulty. Even though we do not attempt to quantify the factors (and do not believe it is worth trying), obviously the anti-difficulty needs to outweigh the difficulty in the potential partner's overall view of the suggested research design.

We have performed a small initial validation of the factors by using them to explain the history of two different strands of our own industrial-collaboration research. Further work should be concerned with the following issues:

- What factors are still missing?
- Do the factors hold when validated against a much broader set of in-industry studies? Their current empirical support is rather narrow.
- In particular, are there culture dependencies? Obviously, cultural differences will influence the *strength* of influence for several of the factors (which was therefore explicitly not our topic here). For instance, in terms of Hofstede's Cultural Dimensions [8], diversity of roles will likely have *less* influence in collectivistic cultures; several factors will likely have *less* influence in cultures strong on Long-Term Orientation; others will likely have *more* influence in cultures strong on Uncertainty Avoidance; and so on. But are some of the factors even *qualitatively* different (e.g. absent) in some cultures?
- How can the factors be used systematically during the research design phase? Should they be taught?
- How should the factors be used when reporting on research? For instance, they might provide a compact form for a helpful discussion of advantages and disadvantages of a certain research design to be used in research articles.
- Can we operationalize a rough scale (low, medium, high) for the strength of difficulties in concrete cases? This could be a useful guide not only for research design but also for reviewers.
- Can we work out a catalog of procedures for addressing each of the difficulties: When will they be strongest? How to mitigate them?

Acknowledgments

We thank all our many industrial partners for admitting us to their realm and for working with us. We thank many of them and many of the candidate-partners-that-never-happened for voicing their thoughts and reservations regarding the studies we suggested.

References

[1] *CESI '13: Proceedings of the 1st International Workshop on Conducting Empirical Studies in Industry*, Piscataway, NJ, USA, 2013. IEEE Press.

[2] *CESI 2014: Proceedings of the 2nd International Workshop on Conducting Empirical Studies in Industry*, New York, NY, USA, 2014. ACM.

[3] Robert Davison, Maris G. Martinsons, and Ned Kock. Principles of canonical action research. *Information Systems Journal*, 14(1):65–86, 2004.

[4] Steve Easterbrook, Janice Singer, Margaret-Anne Storey, and Daniela Damian. Selecting empirical methods for software engineering research. In Forrest Shull, Janice Singer, and Dag I.K. Sjøberg, editors, *Guide to Advanced Empirical Software Engineering*, pages 285–311. Springer, 2008.

[5] Floyd J Fowler, Jr. *Survey Research Methods*. SAGE Publications, Inc, 5th edition, 2014.

[6] Sallyann Freudenberg (née Bryant), Pablo Romero, and Benedict du Boulay. "Talking the talk": Is intermediate-level conversation the key to the pair programming success story? In *AGILE 2007*, pages 84–91, Washington, DC, USA, 2007. IEEE Computer Society.

[7] Paul Grünbacher and Rick Rabiser. Success factors for empirical studies in industry-academia collaboration: A reflection. In *Proceedings of the 1st International Workshop on Conducting Empirical Studies in Industry*, pages 27–32, 2013.

[8] Geert Hofstede. *Culture's Consequences: Comparing Values, Behaviors, Institutions, and Organizations Across Nations*. Sage Publications, 2nd edition, 2001.

[9] Shilpi Jain, Muhammad Ali Babar, and Jude Fernandez. Conducting empirical studies in industry: Balancing rigor and relevance. In *Proceedings of the 1st International Workshop on Conducting Empirical Studies in Industry*, pages 9–14, 2013.

[10] Timothy C. Lethbridge, Steve Lyon, and Peter Perry. The management of university-industry collaborations involving empirical studies of software engineering. In Forrest Shull, Janice Singer, and Dag I.K. Sjøberg, editors, *Guide to Advanced Empirical Software Engineering*, pages 257–281. Springer London, 2008.

[11] Ayse Tosun Misirli, Hakan Erdogmus, Natalia Juristo, and Oscar Dieste. Topic selection in industry experiments. In *Proceedings of the 2nd International Workshop on Conducting Empirical Studies in Industry*, pages 25–30, 2014.

[12] Lutz Prechelt. Agile offsharing: Using pair work to overcome nearshoring difficulties. In *Proc. 6th Intl. Workshop on Cooperative and Human Aspects of Software Engineering (CHASE '13)*, pages 157–158, 2013.

[13] Per Runeson and Martin Höst. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14:131–164, 2009.

[14] Stephan Salinger, Christopher Oezbek, Karl Beecher, and Julia Schenk. Saros: An Eclipse plug-in for Distributed Party Programming. In *Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects on Software Engineering*. ACM, 2010.

[15] Stephan Salinger and Lutz Prechelt. What happens during pair programming? In *Proceedings of the 20th Annual Workshop of the Psychology of Programming Interest Group (PPIG '08)*, Lancaster, England, September 2008. www.ppig.org.

[16] Carolyn B. Seaman. Qualitative methods in empirical studies of software engineering. *IEEE Transactions on Software Engineering*, 25(4):557–572, 1999.

[17] Anselm L. Strauss and Juliet M. Corbin. *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*. SAGE, 1990.

[18] Claes Wohlin, Aybüke Aurum, Lefteris Angelis, Laura Phillips, Yvonne Dittrich, Tony Gorschek, Håkan Grahn, Kennet Henningsson, Simon Kågström, Graham Low, Per Rovegård, Piotr Tomaszewski, Christine van Toorn, and Jeff Winter. The success factors powering industry-academia collaboration. *IEEE Software*, 29(9):67–73, 2012.

[19] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in Software Engineering*. Springer, Berlin Heidelberg, 2012.

[20] Valarie A. Zeithaml. Consumer perceptions of price, quality, and value: A means-end model and synthesis of evidence. *Journal of Marketing*, 52:2–22, July 1988.