e-Informatica Software Engineering Journal
Volume 19, Issue 1, pages: 1–19
DOI: 10.37190/e-Inf250105

BibTeX

Guidelines for Conducting Action Research Studies in Software Engineering

Miroslaw Staron*
*Corresponding author: miroslaw.staron@gu.se

Article info

Keywords:
action research
guidelines
empirical methods

Submitted: 25 Jun. 2024
Revised: 2 Jan. 2025
Accepted: 2 Jan. 2025
Available online: 8 Jan. 2025

Abstract

Context: Action research is popular in software engineering due to its industrial nature and promises of effective technology transfers. Yet, the methodology is still gaining popularity, and guidelines for conducting quality action research studies are needed.
Objective: This paper aims to collect, summarize, and discuss guidelines for conducting action research in academia-industry collaborations. The guidelines are designed for researchers and practitioners alike.
Method: I use existing guidelines for empirical studies and my own experiences to define guidelines for researchers and host organizations for conducting action research.
Results: I identified 22 guidelines for conducting action research studies. They provide actionable recommendations on identifying the relevant context, planning and executing interventions (actions), reporting them, and reasoning around the ethics of action research.
Conclusions: The paper concludes that the best way of engaging with action research is when we can be embedded in the host organization and when the collaboration leads to tangible change in the host organization and the generation of new scientific results.

1. Introduction

After the software engineering research crisis of the 1990s, empirical software engineering gained popularity [1] as one of the remedies to the challenges with the adoption of research in industry. Experimentation was the first to receive a proper treatment with the seminal book by Wohlin et al. [2] with case studies following after [3] and design science research gaining popularity afterward [4]. Now, almost 30 years after the paper by Glass [1] that coined the term software research crisis, the research landscape is much more diverse. The major conferences and journals are significantly more mature in assessing publications, and the need for explicit research methodology for every study is obvious. In 2020, the ACM published guidelines for reviewers of empirical work in software engineering [5], which contained 17 distinct research methodologies. These are just a selection of evidence that indicates that software engineering has matured as a field, although the evolution and development never stops.

Action research gained popularity in the 2000s [6, 7] when its abilities to strengthen industrial collaborations became evident for academics and practitioners alike. Although it

was initially taken from the field of information systems [8], the emphasis on collaborative research and development appealed to software engineering researchers. The action research addresses the challenges and dilemmas many software engineering researchers face – how to introduce and simultaneously study new technology.

Case studies in software engineering are often meant to be observatory or participant – observatory and, therefore, prescribe objectivity in investigations of the studied phenomena. Researchers must be more observers or participants but not executors of actions and interventions at the company. The primary focus in such studies is to understand phenomena in their natural context, but sometimes changes are needed in addition to the observations. Therefore, the guidelines designed for case studies do not always apply to action research.

Experiments are driven by hypotheses and, therefore, prescribe controllability, which favors isolated, small (even toy) problems as the core of experimentation. Design science research focuses on the artifacts rather than improving the practice of the collaborating company. But when a researcher is embedded in a host company, introduces a new technology, and wants to critically and systematically evaluate it, action research is the only methodology that provides the necessary toolkit for this researcher and the required discourse of analysis of the obtained results.

In this paper, I dive into the question of *What are the necessary guidelines for planning, conducting, and reporting action research studies in software engineering?*

Action research is a well-known research methodology in information systems, pedagogy, and nursing. Therefore, to define the guidelines, I started by reviewing the existing guidelines of Davidson [9], Baskerville [8], and Bleijenbergh [10]. These three were selected as they are used as methodological sources of guidelines by being cited and because they include guidelines and recommendations explicitly. The existing guidelines were refined, adopted, and rewritten to fit the context of software engineering, where software development is the focus. Both practitioners and researchers are from within software engineering. They were also aligned with the preliminary recommendations in my previous work on action research's theory and applications [11].

The remainder of the paper is structured as follows. Section 2 presents an overview of a selection of existing studies in action research and existing guidelines for other research methodologies in software engineering (e.g., case studies). Section 3 describes what action research in academia-industry collaboration context is and which elements are essential for a successful collaboration. Section 4 presents the guidelines, and Section 5 concludes the paper.

2. Related work

Before I discuss action research principles and how they shape this research methodology, I overview the state-of-the-art in action research and guidelines for empirical studies. These provide a fundamental overview of what is recommended for researchers and practitioners today in software engineering.

2.1. Studies in action research

Although action research is not a new methodology, it entered software engineering in the 1990s. Dos Santos and Travassos [6] and their subsequent work on using action research in software engineering [7] have found that this research methodology has the

potential to address the challenges of technology transfer. They identified action research as a methodology that can benefit both industry and academia but did not identify concrete guidelines on how to make such a collaboration successful. The existing guidelines for such collaborations often focus on organizational support and management engagement [12, 13].

My book on action research theory and practice in software engineering [11] followed the same principles and focused on the action research methodology to increase the impact of industry-academia collaborations. The action research methodology presented in that book focuses on a five-step cyclic model, which is similar to the view of action research of Baskerville et al. [8]. This paper follows the same model, although the guidelines are applicable for the action research models with an arbitrary number of phases.

Wieringa [4] presented Design Science Research as a research methodology for software engineering. Even there, he mentioned using the so-called "Technical Action Research," which is a kind of action research that primarily focuses on developing artifacts (e.g., programs and tools). The technical action research emphasizes the importance of the collaborative development of artifacts and the need to be applied directly in software development organizations.

Dittrich et al. [14] presented experiences from another type of action research project focused on method development, not technology development. Their experiences show how good the adoption of research results is when conducting action research. The work by Dittrich et al. has been used to guide research on the co-development of new methods in software engineering.

Petersen et al. [15] presented another set of experiences from two action research studies. They identified several positive aspects of action research studies. For example, the deeper impact of the contributions and the more manageable the transfer of the results to industrial practices. However, they also acknowledged such validity threats as context-dependency of the results.

2.2. Existing action research guidelines

Almost any publication that introduces action research as a methodology, whether within software engineering or other fields, includes a reference to the methodology. However, such references do not refer to publications that contain explicit guidelines, are applicable to software engineering, or are directly actionable. The guidelines presented in this paper are based on three other guidelines outlined below, which contain such explicit guidelines.

The most extensive guidelines available are included in the work of Davidson et al. [9]. These guidelines are developed for information systems, where the focus is put on organizational development with the help of an information system. In their work, the authors introduced the concept of *Canonical Action Research* and described five principles, each broken down into several checks:

1. The principle of researcher-client agreement
    a. Did both the researcher and the client agree that CAR was the appropriate approach for the organizational situation?
    b. Was the focus of the research project specified clearly and explicitly?
    c. Did the client make an explicit commitment to the project?
    d. Were the roles and responsibilities of the researcher and client organization members specified explicitly?
    e. Were project objectives and evaluation measures specified explicitly?
    f. Were the data collection and analysis methods specified explicitly?

2. The principle of cyclical process model (CPM)
   a. Did the project follow the CPM or justify any deviation from it?
   b. Did the researcher conduct an independent diagnosis of the organizational situation?
   c. Were the planned actions based explicitly on the results of the diagnosis?
   d. Were the planned actions implemented and evaluated?
   e. Did the researcher reflect on the outcomes of the intervention?
   f. Was this reflection followed by an explicit decision on whether or not to proceed through an additional process cycle?
   g. Were both the exit of the researcher and the conclusion of the project due to either the project objectives being met or some other clearly articulated justification?
3. The principle of theory
   a. Were the project activities guided by a theory or set of theories?
   b. Was the domain of investigation, and the specific problem setting, relevant and significant to the interests of the researcher's community of peers as well as the client?
   c. Was a theoretically based model used to derive the causes of the observed problem?
   d. Did the planned intervention follow from this theoretically-based model?
   e. Was the guiding theory, or any other theory, used to evaluate the outcomes of the intervention?
4. The principle of change through action
   a. Were both the researcher and client motivated to improve the situation?
   b. Were the problem and its hypothesized cause(s) specified as a result of the diagnosis?
   c. Were the planned actions designed to address the hypothesized cause(s)?
   d. Did the client approve the planned actions before they were implemented?
   e. Was the organization situation assessed comprehensively both before and after the intervention?
   f. Were the timing and nature of the actions taken clearly and completely documented?
5. The principle of learning through reflection
   a. Did the researcher provide progress reports to the client and organizational members?
   b. Did both the researcher and the client reflect upon the outcomes of the project?
   c. Were the research activities and outcomes reported clearly and completely?
   d. Were the results considered in terms of implications for further action in this situation?
   e. Were the results considered in terms of implications for action to be taken in related research domains?
   f. Were the results considered in terms of implications for the research community (general knowledge, informing/re-informing theory)?
   g. Were the results considered in terms of the general applicability of CAR?

These guidelines are from the field where software is seen as a tool for organizations. This distinction implies that studies within IS do not study how software is developed but how it impacts organizations. These guidelines also do not consider the necessity for researchers to be embedded in the organizations but often imply that the researchers are external and introduce a change in the organization. This is not how action research is done in software engineering – research must be embedded in the organization, as without it, the change is often abandoned. On the other hand, the change that is adopted leads to technology transfer and wider technology adoption. It also leads to more relevant research in academia, which is often overlooked by the Canonical Action Research and Technical Action Research.

Another explicit set of guidelines is available in the work of Baskerville [8]. These guidelines are conceptually closer to software engineering, but chronologically before Davidson et al.'s guidelines described above. The seven guidelines of Baskerville related to action research projects as a whole:

1. Consider the paradigm shift.
2. Establish a formal research agreement.
3. Provide a theoretical problem statement.
4. Plan data collection methods.
5. Maintain collaboration and subject learning.
6. Promote iterations.
7. Generalize accordingly.

Baskerville's guidelines are at a high abstraction level, which makes them prone to interpretation errors. For example, the third guideline – providing a theoretical problem statement – can be mistaken by conducting a diagnosis phase and identifying a gap in research and theory. Therefore, it must be clarified that action researchers must use theory when designing the entire study, as Davidson prescribes – being guided by a theory.

The third set of guidelines comes from action research applications in human resource management by Bleijenbergh [10]. These guidelines are peripheral to software engineering, but still contain good advice, e.g., the first one.

1. Make sure to involve all relevant organizational stakeholders in the research process.
2. Propose a research strategy that involves empirical observation of real-life settings to get a systemic perspective on the organizational problem.
3. Propose a research strategy that involves participatory research methods, such as workshops, focus groups, participatory modeling, and other decision support systems in developing a plan for action in close collaboration between researchers and (HRM) practitioners.
4. Communicate the need for a relatively large time investment of employees and (HRM) practitioners in the research process.
5. Communicate the need to perform action research over a relatively long period since potentially several cycles of stages are needed to understand the organizational problem and solve it.
6. Make a habit of continuously reflecting on decisions during the research process by making observational, methodological, and theoretical memos.
7. To potentially get high-impact publications out of your action research, consider combining observation of real-life settings with a field experiment.
8. In reporting, create a clear structure of the various stages and cycles of the action research process.
9. To publish action research, authors should put considerable effort into not only describing the rigor of the data collection and analysis, , but also make the contribution to scholarly knowledge explicit.

The above guidelines focus on projects where organizational change focuses on the human side of processes and methods. They are also the set of guidelines that have been cited the least. Similarly to Baskerville's guidelines, they are very abstract and need contextualization.

The fourth set of guidelines comes from Staron [11], in form of a checklist for making the results valid for more than one project. Staron [11] provided a rudimentary set of guidelines for conducting action research studies, focused on both researchers and practitioners, mostly in the context of method and tool development. These guidelines/checks are as follows:

1. Why do we need to have this research project?
2. What is the perception of the company about the research project?
3. In general terms, what need is the research project addressing?
4. Is the scope of the project's cycle well defined?
5. Is the scope of the research project well delimited?
6. Is the cycle well delimited in time?
7. What is/are the deliverable(s) of the research project?
8. Is there a stakeholder appointed? Does the stakeholder have the mandate to drive and implement the results of the research project?
9. Have security issues been addressed prior to the start of the research project?
10. Are employees identified that are going to support the research project?
11. If additional colleagues are participating, how are they going to be kept informed of the progress and results of the research project?
12. Are there regular meetings taking place between the stakeholder and the researcher?
13. During these regular meetings, is the time plan discussed/followed up on?
14. During these regular meetings, is the scope/deliveries/security discussed/followed up on?
15. Is data collected and stored orderly?
16. Does the storage of the data guarantee easy access and, at the same time, fulfillment of security rules?
17. First, the data is refined, analyses are performed, results are presented, and preliminary conclusions are drawn. So, how are the preliminary analyses and results handled?
18. During the second phase, results are put together to support findings and conclusions. So, are results from the evaluation rigorous enough to support the outcome of the research project?
19. How do you specify the results and knowledge to maximize the impact?
20. How should the company be informed about the outcome and findings of the research project?
21. How can the company learn, implement, and utilize the knowledge gained from research projects performed?

The above four guidelines cover different aspects of action research, as they are developed for different purposes. The subsequent sections describe action research in software engineering and its specifics. However, before I start with the guidelines, I need to explore what action research in software engineering is.

3. Context of action research

Action research is a collaborative research methodology [11], where researchers and practitioners work together in an action research team. On the left-hand side (Fig. 1a), the collaboration in ex vivo means that the researchers are outside of the studied context, mostly by design. Case studies are this research method, and as the researcher's objectivity in understanding the studied phenomena, the unit of analysis is prioritized. On the right-hand side (Fig. 1b), the collaboration is done in vivo, meaning that the researchers are embedded in the context that they study. This means that the researchers report to the same organization as the practitioners. Action research and design science research are examples of the latter.

(a) Ex vivo studies, where the research (and the researcher) are outside of the studied context, e.g., case studies

(b) In vivo studies, where the research is embedded in the context and the organization

Figure 1. Two ways of conducting empirical studies with industry in software engineering

In action research, the fact that the researchers are embedded in the organization means that they are either directly employed at the company or that they have the same access to the company as the employees – for example, they are employed at a research organization (a university), but they have access rights, cards, computers from the company and they are part of a team at the company. In Figure 2, the action research is placed in the context of the inputs – theories and practices, as well as the outputs – new methods, processes, and new knowledge.

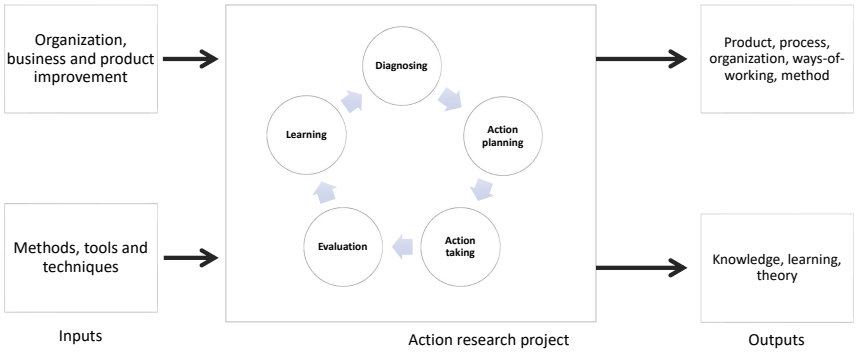

Figure 2. Action research cycles embedded in their context – industry and academia

The in vivo embedding of the research is extremely important as it is the only way to perform interventions (actions) that are the central part of the action research [8]. The fact that the researchers are part of the intervention allows them to understand all aspects of the change they are part of. In other words, they get a first-hand experience of the intervention, not a second-hand perception of it (usually obtained through interviews and observations).

Consequently, this embedding of researchers and practitioners in the host organization (forming an action team, see [11]), the cyclical nature of action research, and explicit focus on interventions and changes to the organization call for different guidelines than the ones for case studies, surveys, or experiments.

On the left-hand side of Figure 2, both the organization and the researchers must come with explicit inputs – the researchers bring in theories, and the organizations (and practitioners) bring concrete challenges to address. Both parties must explicitly describe these inputs, explain them to one another, and agree on them. It is often an iterative process, and theories or challenges can be adjusted during the action research cycles.

However, achieving actionable and scientifically valid results is almost impossible without this understanding. Both Davidson [9] and Baskerville et al. [8] call for theories, but the software engineering context also calls for explicit specification of the initial needs of organization, business, and product improvements from the host organization.

The upfront design and the reliance on theories improve the construct validity of the action research studies. The action team is more confident that the results obtained are based on the correct observations and interventions, not due to chance or confounding factors.

3.1. What constitutes action research

Action research studies build upon three pillars, as shown in Figure 3. We need all of them to make the action research studies work.

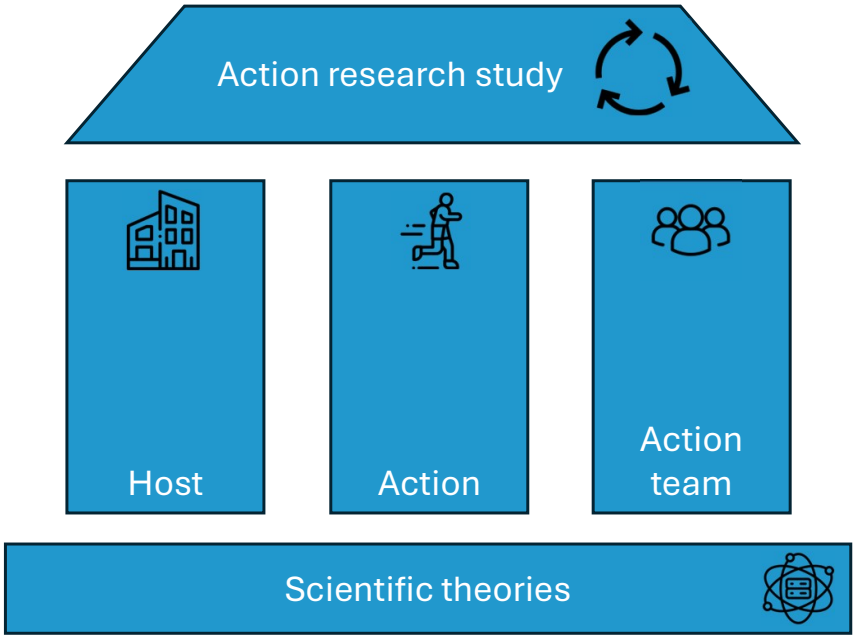

Figure 3. Pillars of action research – host organization, intervention, and the action team

The most important element of any action research study is the **action/intervention**. The action, a synonym for the intervention, is when we change the practice in the host organization. We do that to observe its results and rigorously create new knowledge and insights from that action. The interventions often include improvements to the operations, such as introducing new tools or changing the ways of working in the organization. An example of such an intervention can be changing how the company writes its commit messages by adding the ID of the user story that is implemented; the effect of this action would be to find how many commits are done per user story to understand how to accelerate software development.

The host organization, which directly benefits from the action research project, holds significant responsibilities for implementing the intervention. It must accommodate the action team by granting access to external researchers and ensuring that internal team members have the necessary time and resources to conduct the study. The organization should also allow the action team to obtain time from the reference team and the man-

agement team because they need to help shape and guide high-level goals for the action research project.

We should remember that different organizations and management/governance structures exist. In hierarchical organizations, we must always ensure that the management is informed and involved in the study. They make the decisions, and their permissions are crucial in such organizations. We should engage with our research partners to find the appropriate management level. In more agile, self-organized, or empowered organizations, we must ensure that we have sufficient engagement from the teams and individuals, as they are the ones who are in control of their time and, to some extent, work assignments. In such organizations, we should engage with management after we have engaged the teams.

Finally, the action team must be assembled with researchers and practitioners with the necessary competence. First, at least some research team members should have a research degree because they need skills in planning and conducting quality research studies. The researchers should also be able to identify the novelty of the studied topic – not all topics are equally important for the research community. Researchers need to consider the research value of any action research project. If the topic is not new, the study should consider that by planning for a replication (often with modification) or by expanding on the existing research results by modifying the context to increase the value for the company (do not re-do the work of others) and for the research community (create new findings). Second, at least some action team members should be practitioners from the host company. They have domain competence, which is required for accurate planning, executing, and interpreting the action research project results. Per definition, the ways of working at the company and the company's ability to use the results are in the hands of the host organization and, by extension, by the practitioners of that organization.

3.2. Situations Where Action Research is not Appropriate

One of the main challenges when conducting action research studies is understanding when it is inappropriate and when to pivot and change the research methodology. When engaging in action research, there are a few cues. First, if we do not have access to the company and no company representative is part of the research team, we must choose another research methodology. The lack of access means we cannot conduct necessary interventions for action research.

In the context of industry-academia collaborative action research, we should not engage in action research if we cannot guarantee transparency and the ability to share results (to a degree, we cannot demand that from our industrial partners). Instead, we should focus on conducting offline studies to recreate the company's context, allowing us to conduct experiments.

When researchers cannot commit to the project for the long term, we should also choose another methodology. Irregular meetings, short presentations, and a lack of systematic collaboration lead to poor or no results. We should also not engage when we cannot contribute to the practice and we are after publications, not impacting the industry. The lack of direct industrial impact (more than just publications) indicates that we are not engaging in action research.

4. Guidelines

The guidelines presented in this section have been designed based on guidelines for collaboration with companies from my own research [11, 16], as well as the guidelines from others [3, 8–10]. These guidelines relate to both the Canonical Action Research [9] and Technical Action Research [4].

4.1. Project set-up

I can now introduce the first guideline, which is based on two existing recommendations from Davidson [9] and one from Baskerville et al. [8]. It is about the starting of the action research project.

Guideline 1: Engage in action research only when it can be embedded in the host organization.

Collaboration with an industrial partner or being a part of a software development organization is necessary for an action research project, but it is not a sufficient condition. First, both parties must agree that action research is appropriate and allowed in the organization [9]. We need to be able to apply research results to the organization directly. We must be able to make interventions in the organization and observe their effects.

From the organization's perspective, the researchers must be embedded in it because they must understand the details of the company's operations. To understand it, they must have access to the documents, processes, and products in the same way as the company's employees. This means there is a paradigm shift in the organization – research is done in parallel to the normal operations [8]. If we cannot embed actions/interventions in the host organization, I recommend choosing other types of research methodologies, in particular, design science research or a case study. We should choose the case study based on whether the goal is to observe the organization and keep the distance from its operations and whether our goal is to understand these operations without influencing them. We should choose design science research when our goal is to create and develop a new artifact and to study its use in an industrial context. In that case, we can and should influence the company's operations, but we should focus on the qualities of this artifact.

From the researcher's perspective, the actions should be guided by theories in software engineering. We must be able to find existing ones or formulate new ones; otherwise, scientific investigation will not be guided by theories. As software engineering theories are formulated as strictly as in other fields, e.g., mathematics or physics, we can substitute them with existing empirical results. We can use existing empirical evidence to guide us when planning actions. We can also use existing research results when we explicitly want to validate them in our action research project. The next guideline makes it explicit:

Guideline 2: The objectives of the research project must be specified upfront.

When conducting action research projects, we must be clear about the project's aims, goals, and objectives. The researchers should specify their research results, publications, and engagement expectations. The practitioners should be transparent about their expectations regarding resources, the ability to conduct interventions, and collaboration with academic partners. When discussing these expectations, one or both parties may realize that action research is not the best methodology to study the phenomena in question or deliver the

desired results. Therefore, we need to be aware that there is always the possibility to pivot, hence the next guideline.

Guideline 3: Be ready to pivot on your research methodology if the situation is not optimal for action research.

Every action research cycle has the perfect activity where we decide upon the continuation of the project – the learning phase. Davidson [9] explicitly calls for evaluating whether it is appropriate to continue. However, discontinuing action research can mean pivoting on the research methodology. We should be prepared for it because a non-optimal environment for action research can lead to failed projects and a lack of tangible results. In most cases, to get more familiar with the organization, it is better to start with a case study and follow the guidelines by Runeson et al. [3] or follow the guidelines by Wohlin et al. [17] to find even a better fit.

4.2. Host environment

The host environment must be organized to ensure both parties a positive action research experience. The academic side should be able to test theories and ideas. In contrast, the industrial side should perceive value in increased knowledge, enhanced competence, improved products, new features, or operational improvements. This balance is essential for successful collaboration.

Action research projects must be embedded within the host organization. Company-employee researchers often conduct these projects internally, frequently leading to applied results. External members of the action team must ensure they are treated equally. This consideration leads to the following recommendation.

Guideline 4: Action team should comprise both practitioners and researchers alike.

Just as the input values are both theories and the organization's challenges, the action team must reflect that. However, it is strongly advised to use the host organization's computers and infrastructure to ensure information security. This ensures that any information intended for publication is thoroughly scrutinized before being taken outside the organization. Therefore, here is my next guideline:

Guideline 5: Understand the host organization and secure appropriate engagement in the correct order: approach management first in hierarchical organizations, and start with the team in empowered organizations.

All previous recommendations have similar guidelines, although they are presented more fragmentedly. They either call for the involvement of all relevant parties (Bleijenbergh [10]), explicit commitment (Davidson [9]), and formal research agreement (Baskerville et al. [8]). Modern software development organizations are not as hierarchical as they used to be (or as non-software organizations are). Therefore, the action research collaboration agreement stretches over several levels of these organizations.

Therefore, the following guideline, which is for researchers from an external organization, is equally important.

Guideline 6: The researchers must respect the rules, principles, and obligations of the host organization as if they were employed there.

This guideline means that researchers must read, understand, and adhere to the host organization's rules, principles, and obligations as if they were employees. These obligations include maintaining confidentiality, upholding the organization's standards and practices throughout the research project, and being transparent and loyal to the organization.

Although it may be considered challenging for academic freedom to pursue any research problem or publish the results based on the findings, it is not. Software development companies' intellectual property is often soft and needs protection. Revealing parts of the source code, tests, or requirements (to name a few examples) can lead to leaking intellectual property to competitors. Therefore, researchers must be able to generalize their results to such a level that does not harm the company and is transferable to other contexts.

4.3. Interventions

One of my long-term collaborators once aptly summarized the essence of interventions in action research by saying, "Somebody has to do something." This captures the fundamental principle that action research must involve meaningful, tangible actions within the host company. To make these tangible observations and improvements, we must plan the actions and plan how to observe their results. Therefore, we need to follow the next guideline.

Guideline 7: Every intervention should be planned, lead to observable effects, and properly evaluated.

The observable effects must be both quantitative and qualitative because, without any observable effect, it's impossible to understand what the action/intervention changed. Quantitative effects can be collected by measuring products, processes, or designs, while qualitative effects can be observed through interviews, workshops, and post-mortem analyses. The quantitative aspect allows measuring the intervention's effects, while the qualitative component captures the broader context.

Consequently, we must prepare (plan) for the intervention together in the action team. We must measure the baselines and conduct qualitative data collection before the intervention. After the intervention, we must evaluate the effect and collect the exact measurements and qualitative data to understand the scope, size, and magnitude of the impact. Davidson [9] has a similar recommendation but without the emphasis on the observability and measurability of the effect, which I find very important.

Implementing concrete changes is the starting point and a prerequisite for its success. The boldface accent indicates that the emphasis is on the implementation.

Guideline 8: Interventions in each cycle should be atomic.

Long interventions are prone to interruptions; the longer the intervention, the higher the risk that other priorities will take precedence over the research study. Common disruptors, such as implementing new features or investigating new defects, can cause the host organization to shift focus. These disruptions are confounding factors, making it difficult to determine whether the observed results are due to the intervention or these other changes. Davidson [9] recommends specifying the focus up-front, but we can go even further and ensure that the focus is as atomic as possible.

If the intervention's nature requires longevity, the action team should include measures to capture the effects of potential confounding factors. The action team must also be prepared to adjust the data collection methods amidst changes to the context.

The host organization must be prepared for this, which means allocating additional resources, adjusting the project schedule, and accounting for the risk of failure in the current plan, such as in the ongoing sprint. Conducting a mock-up study or creating a pilot product to demonstrate a tool outside of current practices is not action research, as it does not introduce observable changes; it is merely an offline study. The next guideline is therefore:

Guideline 9: Practitioners in the action team must be able to conduct the intervention.

Practitioners should intervene within the scope of their work rather than attempting to change or advocate for the practices of other teams or organizations. The intervention should focus on their methods, tools, and infrastructure, ensuring that the context of the intervention is specific to their environment and not external to it. If other teams are affected, they should be included in the action team. Davidson [9] recommends that researchers and practitioners are motivated, but only motivation is insufficient; they must be able to conduct it, given their daily work, the scope of their industrial project, or competence. When it is impossible to perform an intervention, we can pivot to design science research and focus on the artifact instead, as we did in one of our research studies [18].

4.4. Planning, conducting and analyzing

Since the host organization engages in various activities, including organizational changes, it is essential to distinguish between interventions and routine operations.

Guideline 10: The scope of the intervention must be specified up-front based on theories.

Although interventions are atomic, their scope must be thoroughly planned and based on existing theories. Otherwise, we cannot contribute to the existing body of scientific knowledge. We must set the stop criteria and ensure we reach them. The same is true for the deliverables – they also need to be defined up-front, and their quality goals must be specified beforehand. We need to know when the intervention was successful and when it was not. Davidson and Baskerville et al. recommend having a solid theoretical backing of actions, but in software engineering, we can also identify new theories based on empirical observations. If so, our initial theory must be formulated upfront, validated, refined, and changed during the action research study.

Changes in ways of working that lack such planning cannot be considered interventions in action research. Properly distinguishing these activities ensures the validity and reliability of the research outcomes – if we do not have a baseline, we do not know if we improved. Additionally, this approach helps maintain clarity and focus on the research objectives, preventing confounding planned interventions with everyday business activities, hence our next guideline:

Guideline 11: Every intervention must be compared to a baseline, so ensure that there is one.

Inspired by the recommendation to plan data collection methods by Baskerville et al. [8], we can go one step further. We must establish a proper baseline for comparison. For instance, when introducing a new method, ensure that data from the old method is available for comparison. This may require collecting data before initiating the intervention. We should focus on quantitative and qualitative data in this data collection. The first provides us with facts, whereas the second allows us to get a deeper understanding of the reasons behind the phenomena observed in the quantitative data.

If pre-intervention data collection is necessary, we should be prepared to mitigate the Hawthorne effect [19]. We collect the data based on the theoretical models we bring to the project (see Guideline 2). For example, when we want to validate the effect of introducing modern code reviews, we should measure the performance of code reviews before this introduction in terms of speed, duration, and number of reviews. We also need to be prepared to adjust if the data that is needed cannot be collected:

Guideline 12: Explicitly plan for the intervention and data collection based on the theoretical model.

Since organizations are constantly changing, ensure that when the intervention is done, there are no confounders like reorganizations, new team onboarding, product changes, or other events that could affect the result. If they are, be prepared to document these events and study their effects or pivot (if the situation for action research is no longer optimal).

Our data collection methods must originate from the theories we apply in the action research. We must follow a theoretical model, as Davidson [9] points out. Measurements and data collection are essential parts of such a model.

Nevertheless, we must use the knowledge we gain in the diagnosing phase and pre-defined theories.

Guideline 13: Design and plan interventions based on causes identified in the diagnosing phase.

We can do it in two ways: either select the theory that aligns with the needs identified in the diagnosing phase or complement the existing theory with concepts and relations identified in the diagnosing phase. Davidson [9] recommends basing actions on the diagnosis, but we should try to reconcile both theories and empirical findings in the diagnosing phase.

Unlike other research methodologies, adjusting the study's setup during each cycle allows for significant flexibility. The action team can introduce new theories (e.g., from processes to products), change the scope of the cycle (e.g., from methods to tools), adjust the team (e.g., introduce new roles), or re-design the intervention so that it fits both the organization and the theory.

Sometimes, it is difficult to conduct action research studies for practical reasons. We may introduce harm to the organization by revealing some information or making a change that will cause problems for the company's operations. Therefore, I recommend also the following:

Guideline 14: Be ready to stop the intervention if it causes harm to the organization

Since the outcome of research activities is always unknown, we take the risk that the activity does not go according to plan. Therefore, we must, at all costs, prevent damage or harm to the organization, its employees, and its business. Otherwise, we do not follow the Nuremberg convention that all scientists should comply with – "Do no harm." This also means that we must have a contingency plan in place.

The contingency plan means we must continuously keep track of our collected data and how to address the research questions using it. We can also prepare alternative research questions to address using the data we have to maximize the chance of reporting results. We should remember that even negative or inconclusive results are of certain value to the research community. Additionally, publishing negative experiences, if they are generalizable, helps inform others of the risks encountered, contributing to a broader understanding and awareness within the field. This approach enhances the learning process and builds

a repository of knowledge that can guide future interventions. Furthermore, documenting these experiences fosters a culture of transparency and continuous improvement within the organization.

When we engage in action research, it is easy to forget that sometimes we must change the topic or even stop the study. Therefore, my next guideline comes directly from my experiences in measurement system development, which was conducted as an action research study.

Guideline 15: The management and the action team should continuously assess the need for further studies.

We must remember that action research is a collaborative endeavor, requiring continuous involvement from the host organization's management to define and potentially finalize the studies. Depending on the specific context and goals of the research, the process might conclude after two or three cycles or extend to ten or more cycles.

Long-lived action research projects can address multiple challenges and test several theories. A good action research team does not happen often, but when it does, it usually lasts for many iterations. However, there is a danger with long-lived, static action research teams; they can get stuck in solving no longer important problems. Conducting a post-mortem workshop after the intervention is always a good idea to understand what to do next and how. If the change is not possible or needed by the host organization, the action team can find other problems to address, change the team's setup, or even change organizations [11].

4.5. Reporting

We often think that research must be conducted from start to finish to generate knowledge. In action research studies, however, even intermediate results are important. Since action research is done in cycles, we should be able to present the results from each cycle independently from each other as well as in the context of each other. Hence, the next guideline:

Guideline 16: Package intermediate results in a reusable way.

We must ensure that our results, tools, and methods are reusable, making them accessible to the reference team, other teams within the host organization, or even other organizations (see Davidson's fifth recommendation [9] about the impact of the action research projects). Researchers must remember that the value for the company is not found in a published paper, confusion matrix, or statistics alone. The real value for the host organization lies in tangible and actionable improvements, new products, features, increased operational efficiency, or better architectures. This focus on practical outcomes ensures that the research has a meaningful and lasting impact on the organization's success. It is often these kinds of tangible results that the research community values the most.

For the organization, the results of the action research cycles must be documented and disseminated even more frequently. Immediate publication on internal forums, web pages, podcasts, and webinars helps to spread good practices. Therefore, the action team should follow the next guideline to make communication efficient.

Guideline 17: The results should be continuously documented for academia and industry.

This internal documentation should include hands-on guides on how to apply the new knowledge, contact points for internal expertise, and locations of stored information

from the action research. Action research projects often develop tools and instruments specifically configured for the host organization's infrastructure, which must be readily accessible internally. This approach ensures that the practical benefits of the research are fully realized and integrated within the organization. Davidson [9] recommends timely documentation of the research results, although it does not distinguish between the industrial and scientific audience.

When conducting action research, it's crucial to communicate the actions and results to management and the reference group. They must know if the actions provide value, and we need to know if the actions are going according to the plan. We must remember that our work should be about generating new knowledge, not conducting activities—the activities are a means to achieve the results.

Defining the context of the research is almost as important as the results. In action research, describing the organization by its name is tempting, but most often, it should be complemented with more details. Describing the organization's characteristics is more important, e.g., [20]. If the action team has to choose whether to reveal the organization's name or its characteristics, the characteristics provide more value to other researchers.

Guideline 18: Characterize the context of the study, including undertaken actions/interventions.

We should include as much relevant information about the context as possible when generalizing empirical and action research studies. In particular, such information as:

- Type of the organization and its process – e.g., web development organization working according to agile principles.
- Size and other characteristics of the product/service – e.g., 1,000,000 LOC written in C#.
- Context of the project – e.g., part of an internal reorganization or a larger research project.
- Host organization – e.g., the software development team of 20 persons.

This way of conveying characteristics helps others identify whether their context is similar or different from the reported study.

4.6. Ethical considerations for action research studies

Our ethical stance should be to "do no harm" when researching. In action research, this includes two aspects: direct harm to people (and, by extension, the organization) and indirect harm through new insights (and, by extension, products). This is included in Guideline 14: Be ready to stop the intervention if it causes harm to the organization.

We have access to the data of the host organization, which means that we can also access information that was not meant for the purpose of this study or even information that was part of the study but is not meant to be spread. This can include private conversations, e-mails, or commit messages containing sensitive information. Revealing such sensitive information can harm individuals, as they can lose their roles or even jobs. This can also be detrimental to the organization's business as the customers may lose trust in the company's products and services. Therefore, we should always consult the company's management and legal department before spreading information that could be sensitive – both internally and externally.

At the same time, we must recognize that our products and knowledge can potentially harm organizations and individuals. Our aim should always be to foster improvement rather

than cause problems. When our findings pertain to individual improvements, we should direct our results to those individuals, inviting them to knowledge-sharing sessions and seminars. When presenting such results in a larger forum, it is essential to anonymize the data by removing names and identifying details of individuals or teams. We can emphasize best practices and effective working methods, but we must avoid singling out specific individuals or organizations.

Guideline 19: Reflect on the results objectively and avoid overinterpretation.

One of the main problems that I observed in action research collaborations is the tendency to overinterpret the results. Both researchers and practitioners tend to generalize the results too quickly. The fact that an improvement worked for one organization does not guarantee it will work for another. Overinterpreting the results is ethically problematic because we are in danger of speculation – we do not know if the same methods would work elsewhere. Both Davidson [9] and Baskerville et al. [8] warn about this too.

When reflecting on the results, it is better to follow the next guideline:

Guideline 20: Interpret the results based on theory

The theoretical foundation of action research cycles helps us ensure objective results. When we specify the theory upfront, we use it to interpret the results, which helps avoid overinterpretation, speculation, or even the Hawthorne effect.

5. Conclusions

Action research has been identified as one of the upcoming research methods in the last two decades. An increasing number of software engineering papers have been published in which this research methodology has been used, which indicates that there is a growing interest in it. The rise of industrial software engineering research, pioneered by large software development companies, nourishes this methodology.

This paper identified 20 guidelines for conducting action research studies in the context of academia-industry collaboration. Although these guidelines are probably relevant to industrial action research (in vivo action research), they have not yet been validated in that context. The guidelines should be seen as the basics and either a reference for action research or a starting point for designing such studies. They can be complemented by studying existing articles about action research, e.g., Kantola et al. [21] or Natarajan et al. [22] and Kemell et al. [23].

The guidelines presented in this paper are specifically targeted towards software engineering, i.e., the discipline of development of software and the development of the processes for developing software. What makes software engineering specific in this context is the fact that software development is the main context, not the use of IT technology or software in other domains, which is often the case of studies in information systems; it is the information systems research where action research historically has been very popular.

However, despite the growing popularity of action research, it is not for every type of research. We still need to use experimentation, case studies, and surveys to understand the fundamental phenomena of software engineering, to study industrial cases in software engineering, or to understand the spread of phenomena in the entire community. It's important to use the best methods and tools suited for the task and not revive to a specific one all the time. Action research works best when it is done in collaboration between

academia and industry and when the entire action team can be embedded in the host organization.

Acknowledgments

I would like to thank Software Center, a collaboration between five universities and 17 companies, for providing and financing the context of this work. I also thank the anonymous reviewers for constructive, actionable feedback.

CRediT authorship contribution statement

Miroslaw Staron: Conceptualization, data collection, analysis, writing.

Declaration of competing interest

I declare that I have no competing interests.

Funding

This paper has been partially sponsored by Software Center, www.software-center.se

References

[1] R.L. Glass, "The software-research crisis," *IEEE Software*, Vol. 11, No. 6, 1994, pp. 42–47.
[2] C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell et al., "Experimentation in software engineering: An introduction," *The Kluwer International Series In Software Engineering*, 2000.
[3] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, Vol. 14, 2009, pp. 131–164.
[4] R.J. Wieringa, *Design science methodology for information systems and software engineering*. Springer, 2014.
[5] P. Ralph, N. bin Ali, S. Baltes, D. Bianculli, J. Diaz et al., "Empirical standards for software engineering research," *arXiv preprint arXiv:2010.03525*, 2020.
[6] P.S.M. dos Santos and G.H. Travassos, "Action research use in software engineering: An initial survey," in *2009 3rd international Symposium on empirical software Engineering and measurement*. IEEE, 2009, pp. 414–417.
[7] P.S.M. Dos Santos and G.H. Travassos, "Action research can swing the balance in experimental software engineering," in *Advances in computers*. Elsevier, 2011, Vol. 83, pp. 205–276.
[8] R.L. Baskerville, "Investigating information systems with action research," *Communications of the Association for Information Systems*, Vol. 2, No. 1, 1999, p. 19.
[9] R. Davison, M.G. Martinsons, and N. Kock, "Principles of canonical action research," *Information Systems Journal*, Vol. 14, No. 1, 2004, pp. 65–86.
[10] I. Bleijenbergh, J. van Mierlo, and T. Bondarouk, "Closing the gap between scholarly knowledge and practice: Guidelines for HRM action research," *Human Resource Management Review*, Vol. 31, No. 2, 2021, p. 100764.
[11] M. Staron, *Action research in software engineering*. Springer, 2020.

[12] Q. Song and P. Runeson, "Industry-academia collaboration for realism in software engineering research: Insights and recommendations," *Information and Software Technology*, Vol. 156, 2023, p. 107135.
[13] P. Runeson, "It takes two to tango – an experience report on industry–academia collaboration," in *Fifth International Conference on Software Testing, Verification and Validation*. IEEE, 2012, pp. 872–877.
[14] Y. Dittrich, K. Rönkkö, J. Eriksson, C. Hansson, and O. Lindeberg, "Cooperative method development: Combining qualitative empirical research with method, technique and process improvement," *Empirical Software Engineering*, Vol. 13, 2008, pp. 231–260.
[15] K. Petersen, C. Gencel, N. Asghari, D. Baca, and S. Betz, "Action research as a model for industry-academia collaboration in the software engineering context," in *Proceedings of the 2014 international workshop on Long-term industrial collaboration on software engineering*, 2014, pp. 55–62.
[16] M. Staron and W. Meding, *Software development measurement programs*, Vol. 10. Springer, 2018.
[17] C. Wohlin and P. Runeson, "Guiding the selection of research methodology in industry – Academia collaboration in software engineering," *Information and Software Technology*, Vol. 140, 2021, p. 106678.
[18] M. Staron, J. Strom, A. Karlsson, and W. Meding, "Using generative AI to support standardization work – The case of 3GPP," *International Conference on Product-Oriented Software Process Improvement, PROFES*, 2024.
[19] P. Sedgwick and N. Greenwood, "Understanding the Hawthorne effect," *The BMJ*, Vol. 351, 2015.
[20] M. Staron and W. Meding, "Mesram – A method for assessing robustness of measurement programs in large software development organizations and its industrial evaluation," *Journal of Systems and Software*, Vol. 113, 2016, pp. 76–100.
[21] K. Kantola, J. Vanhanen, and J. Tolvanen, "Mind the product owner: An action research project into agile release planning," *Information and Software Technology*, Vol. 147, 2022, p. 106900. [Online]. https://www.sciencedirect.com/science/article/pii/S0950584922000611
[22] T. Natarajan and S. Pichai, "Behaviour-driven development and metrics framework for enhanced agile practices in scrum teams," *Information and Software Technology*, Vol. 170, 2024, p. 107435. [Online]. https://www.sciencedirect.com/science/article/pii/S0950584924000405
[23] K.K. Kemell, A. Nguyen-Duc, M. Suoranta, and P. Abrahamsson, "Startcards – a method for early-stage software startups," *Information and Software Technology*, Vol. 160, 2023, p. 107224. [Online]. https://www.sciencedirect.com/science/article/pii/S0950584923000782

Authors and affiliations

Miroslaw Staron
e-mail: miroslaw.staron@gu.se
ORCID: https://orcid.org/0000-0002-9052-0864
Computer Science and Engineering,
University of Gothenburg and Chalmers University of
Technology, Sweden