The Journal of Systems & Software 202 (2023) 111684

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

Empirical research in software architecture — Perceptions of the community[☆]

Matthias Galster[a,*], Danny Weyns[b,c]

[a] University of Canterbury, Christchurch, New Zealand
[b] Katholieke Universiteit Leuven, Belgium
[c] Linnaeus University, Sweden

ARTICLE INFO

Article history:
Received 2 July 2022
Received in revised form 24 March 2023
Accepted 26 March 2023
Available online 6 April 2023

Keywords:
Software architecture
Empirical research
Perceptions of community
Survey
Literature review

ABSTRACT

Context: Previous research highlighted concerns about empirical research in software engineering (e.g., reproducibility, applicability of findings). It is unclear how these concerns reflect views of those who conduct and evaluate research.
Objective: Focusing on software architecture, one subfield of software engineering, we study perceptions of the research community on (1) how empirical research is applied, (2) human participants, (3) internal and external validity, and (4) replications.
Method: We collected responses from 105 key players in architecture research via a survey; we analyzed data quantitatively and qualitatively.
Results: Although respondents do generally not prefer either quantitative or qualitative research, around 40% express a preference for various reasons. Professionals are the preferred participants; there is no consensus on the value of student participants. Also, there is no consensus on when to focus on internal or external validity. Most respondents value replications, but acknowledge difficulties. A comparison with published research shows differences between how the community thinks research should be done.
Conclusions: We provide evidence that consensus about empirical research is limited. Findings have implications for conducting and reviewing empirical research (e.g., training researchers and reviewers), and call for reflection on empirical research (e.g., to resolve conflicts). We outline actions for the future.

© 2023 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

In this section, we start with describing the problem and context. Then, we introduce the objective of our research and research questions and present the contributions of this paper.

1.1. Problem and context

Empirical research allows us to collect, interpret and use evidence to understand, select, apply, and advance software engineering methods, processes, practices, techniques and tools (Falessi et al., 2010). Unlike anecdotes, rhetoric or folklore, empirical evidence contributes to confidence in scientific discoveries and the usefulness of "solutions". Over the years, the software engineering community has been reflecting on the meaning of science and the role of empiricism (Wohlin and Aurum, 2015), the understanding and characteristics of research methods (Stol and Fitzgerald, 2018), the relevance and worthiness of research (Lo et al., 2015; Tang and Kazman, 2017), the gap between research and practice (Rabiser et al., 2018), and the balance between relevance (i.e., tackling problems that are identified from and solved for substantial problems of practice) and rigor (i.e., applying systematic and scientifically grounded approaches to solving problems) (Ivarsson and Gorschek, 2011). In this paper, we study empirical research in software architecture, an important subfield of software engineering (Garlan, 2014). The software architecture community has distinct high-quality publication venues dedicated to software architecture research and practice, but many members of the community are also active in the broader software engineering community. Targeting empirical research in software architecture allows us to focus the efforts of our analysis and related insights. On the other hand, the perceptions of the software architecture community can also be contrasted with those of the general software engineering community, such as the ones presented by Siegmund et al. (2015). This could help us understand if there are differences between general software engineering and its subfields.

[☆] Editor: Neil Ernst.
[*] Corresponding author.
E-mail addresses: mgalster@ieee.org (M. Galster), danny.weyns@kuleuven.be (D. Weyns).

https://doi.org/10.1016/j.jss.2023.111684
0164-1212/© 2023 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

For software architecture it has been claimed that it is hard to apply empirical methods. This is due to difficulties measuring architecture "goodness" and dependencies between design decisions, social and human factors (Falessi et al., 2007). Also, software architecture relies on diverse concepts related to computer science, modeling, processes, human factors, technologies, etc. and advances are often driven by "talented people and industrial experience" (Falessi et al., 2010). Focusing on the specific subfield of software architecture can lead to insights that general software engineering research may overlook. Furthermore, unlike other studies (e.g., Siegmund et al. (2015)) we analyze answers based on the background of respondents, analyze the understanding of respondents about empirical research and specifically perceptions on human participants, which is a key aspect in software architecture.

Our previous work has shown that empirical research has not been very common in software architecture (Galster and Weyns, 2016), but gained popularity only recently. Researchers may struggle with the different expectations about an empirical paper. Similarly, educators may not be confident about how to train (new generations of) researchers in empirical research. Emerging and less experienced researchers may not have a clear picture of the role of empiricism in software architecture research, in particular on the aspects for which there is consensus and disagreement among experienced researchers. Finally, practitioners may find it difficult to judge the value and purpose of empirical software architecture research. Lacking research standards in software architecture can lead to research without rigorous grounding and therefore can negatively impact the significance and relevance of research. As we have shown in our previous literature review (Galster and Weyns, 2016), empirical studies in software architecture lack details about research and study designs. Similarly, Konersmann et al. found that generalizability and repeatability of evaluations in software architecture could be improved to enhance the maturity of the field (Konersmann et al., 2022). This (a) raises concerns about the reliability of results, and (b) makes it difficult to reproduce and extend them. Reproducibility of empirical studies is an essential part for ensuring a high quality scientific process and improving the credibility of studies (Rodriguez-Pacrez et al., 2018).

1.2. Objective and research questions

We aim at *understanding the perceptions* on empirical research of those who *conduct* and *evaluate* software architecture research. This is to complement previous works on the state-of-practice such as Galster and Weyns (2016) or Konersmann et al. (2022) on *what actually happens* in empirical software architecture research with views on what members of the research community *think is happening* or *should be happening*. We currently lack an understanding of beliefs and perceptions on research of those who conduct and evaluate empirical software architecture research. This can be the reason for why there is a lack of consensus on empirical research and how to build up evidence (Siegmund et al., 2015). Besides hindering the execution of empirical research, this also affects reproducibility and the willingness of researchers to go beyond "one-off" studies and studies that focus on presenting "new and exciting" results (Galster et al., 2018). Furthermore, our study allows us to contrast perceptions in one subfield of software engineering, i.e., perceptions of the software architecture community, with those of the general software engineering community such as the ones presented by Siegmund et al. (2015).

We explore the following research questions:

- **RQ1: What is the understanding of the software architecture research community on how empirical research is applied in software architecture?** In general, empirical research can be quantitative (e.g., experiments to compare the performance of algorithms based on measurements) or qualitative (e.g., case studies that involve observations and interviews of how developers use tools) or a mix of both (Wohlin and Aurum, 2015), typically depending on the research goal. However, there has been an ongoing debate in the broader research community about the usefulness and value of qualitative versus quantitative methods, see for instance Morgan (2018), Sale et al. (2002). In software engineering, qualitative methods are sometimes preferred as they allow researchers to delve into the complexity of a problem and help gain a deep understanding of problems and their context (rather than abstracting complexity away), see Dyba and Dingsoyr (2008). Sometimes, quantitative methods are preferred, for instance because it is easier to interpret and generalize the results (Wohlin et al., 2003), or even because they are considered more "scientific" compared to qualitative research, regardless of the research goal (Tofan et al., 2011). RQ1 wants to get insight in the perception of the community on how empirical research is or should be applied.
- **RQ2: What is the perception of the software architecture research community about human study participants?** Software architecting involves design, creativity, and requires human input and judgment. Also, architecture solutions are usually not "right" or "wrong" per se, but depend on the context and trade-offs (Bass et al., 2012). Therefore, they depend on the experience and expertise of those making architectural decisions (Falessi et al., 2011), see for example the large amount of research on supporting architects to make better decisions (Tofan et al., 2014). Furthermore, in the general software engineering community there have been discussions about human participants (professionals versus students) and whether (or when) students are valid proxies for professionals (Falessi et al., 2018; Salman et al., 2015) and we want to understand the perception of the software architecture community.
- **RQ3: What are the perceptions of the software architecture research community about internal versus external validity?** *Internal validity* is about confounding factors that may impact study outcomes. It is a concern in particular when causal relationships are explored (Runeson and Höst, 2009). Maximizing internal validity means to exclude almost all influencing factors and conduct a study in a controlled setting. This may come at the cost of external validity (Siegmund et al., 2015). *External validity* is about the generalizability of study results and how the studied problem or context is representative and relevant outside the investigated context (Wohlin et al., 2003). There is a trade-off between fully explainable results (but which may not be generalizable) and results that are applicable in different contexts (but for which we cannot explain all underlying factors and relationships), see Siegmund et al. (2015). Increasing internal validity means constraining the context of a study (e.g., to fix or remove confounding factors to understand causality). This, on the other hand, makes results of a study more specific to that study context and more difficult to transfer to other contexts (decreasing external validity). Some have argued for more context-specific research (Basili et al., 2018) and that generalizability is overrated (Briand et al., 2017). RQ3 explores the view of the software architecture research community on internal and external validity.
- **RQ4: What are the perceptions of the software architecture research community about replications?** One way to

increase internal and external validity is to conduct replications. For example, replications help identify and understand confounding factors that may impact the results of a study, hence, contribute to internal validity. Replications aim to obtain the results in a subsequent study by a person or team other than the authors of the original study using, in part, artifacts provided by the author.[1]
In 2015, de Magalhaes et al. investigated replications in software engineering and found that we still lack standardized concepts, guidelines and methodologies to fully support replications (de Magalhaes et al., 2015). Also, previous studies in the field of software architecture found that replications are not common (Galster and Weyns, 2016). Similarly, various other disciplines related to software engineering report problems with replicating studies and results, e.g., Human Computer Interaction (HCI) (Cockburn et al., 2018), mobile computing (Banovic, 2015) and economics (Maniadis et al., 2017), and some researchers raised concerns about a replication crisis in computer science (Cockburn et al., 2020). On the other hand, reproducibility and replications of studies are key characteristics of the scientific method (Baker, 2016). RQ4 investigates opinions and attitudes of software architecture researchers on replications.

1.3. Paper contributions

Our work is meant to stimulate researchers in the software engineering and architecture communities to reflect on their expectations and standards of empirical research. In detail, we put forward the following **contributions:**

1. An overview of perceptions on how empirical research in software architecture *should be conducted*. These perceptions are taken from those who conduct, publish and evaluate empirical research based on a survey of 105 program committee members of the major software architecture conferences. We also compare perceptions to recommendations for empirical research in the literature. This overview also offers insights into those aspects of empirical research in software architecture for which there exists consensus in the community and those aspects for which there is disagreement. Furthermore, it offers insights into the (perceived) maturity of software architecture research.
2. A comparison of the perceptions of the community with *what actually happens*, i.e., the research published at the International Conference on Software Architecture (ICSA), a major venue dedicated to software architecture research.

In summary, we found differences in the perceptions of the members of the research community on empirical research in software architecture. This highlights that consensus on empirical research in software architecture is limited. While similar findings have been reported by others for software engineering in general, e.g., Siegmund et al. (2015), our work confirms that there are different perceptions even *within* one subfield of software engineering. Our contributions support the target audience of this work as follows:

- Researchers who conduct and review empirical research can see that there are different expectations of reviewers about an empirical paper and how to judge a paper.
- Reviewers can benchmark themselves against others, reflect on their views on empirical research compared to others in the software architecture research community, and initiate actions in the community to resolve disagreements.
- Educators can identify issues and potential pitfalls to consider when training (new generations of) researchers in empirical research software architecture.
- Emerging and less experienced researchers get insights about the role of empiricism in software architecture research, in particular on the aspects for which there is consensus among experienced researchers, but also the aspects for which there is disagreement.
- Practitioners can use findings to understand how empirical research is perceived by those who conduct and evaluate research. This can inform practitioners about the purpose and value of empirical software architecture research and help them interpret empirical findings.

Note that we do not argue that more empirical research is always better or that empiricism determines the impact of software architecture research. The type of a chosen research approach or validation depends on the nature of the research questions (Shaw, 2003) and the context in which research is conducted. For example, one may apply formal proofs to demonstrate that a model checker works correctly (rather than collecting and analyzing data).[2] Also, many seminal papers (e.g., Ward Cunningham's paper on technical debt (Cunningham, 1992) and the paper of Dewayne Perry and Alexander Wolf on the foundations of software architecture (Perry and Wolf, 1992)) include no empirical work, but had long-lasting impact.

1.4. Paper outline

In Section 2 we discuss background and related work, including a general discussion of empirical research in software engineering (including some subfields of software engineering) and software architecture. In Section 3 we explain the design of our study. In Section 4 we present the results for each research question. We discuss these results in more detail in Section 5. In this section we also present a review of research published at ICSA and compare it with the perceptions of the software architecture community. Finally, we draw conclusions in Section 6.

2. Background and related work

In the following subsections, we discuss research that explored how empirical research is conducted in software engineering and subfields (Section 2.1). In Section 2.2 we focus on previous research that explored empirical research in the subfield of software architecture.

2.1. Empirical research in software engineering

2.1.1. Literature reviews in general software engineering

Several literature reviews analyzed how empirical research is conducted in software engineering. For example, in 2018 Theisen et al. performed a literature review on papers at the International Conference on Software Engineering 2016 (Theisen et al., 2018). Their study found that reviewers have increased expectations that papers have solid evaluations of the research contribution. Additionally, they found that at least 17% of the papers involve mining software repositories, a category of papers not seen in 2002. Back in 2007, Höfer and Tichy provided a status overview of empirical research in software engineering (Höfer and Tichy, 2007). The authors analyzed articles in Springer's Empirical Software Engineering journal from the first issue in 1996 until 2006. Findings were: (1) The dominant empirical methods are experiments and

[1] https://www.acm.org/publications/policies/artifact-review-badging (ACM Artifact Review and Badging — Version 1.0).

[2] See also Ledo et al. who discuss example evaluation strategies in HCI, such as usage and demonstrations (Ledo et al., 2018).

case studies; longitudinal studies are missing. Around 25% of published experiments are replications. (2) Studies tend to use professionals more often than students as subjects. (3) Studied topics include measurement and metrics, and tools, methods and frameworks. Design and architecture-related topics were less frequently studied. In 2006, Zannier et al. investigated empirical software engineering research at ICSE, the International Conference on Software Engineering (Zannier et al., 2006). The study analyzed 5% of all papers published at ICSE and found that 70% of these papers included some form of evaluation. Experience reports (rather than rigorous empirical studies following a well-established method) were the most frequently type. Furthermore, they found only few papers with a clear hypothesis and noticed an absence of replications.

2.1.2. Literature reviews on particular empirical methods

Some reviews on the use of particular empirical methods exist. For example, Sjoberg et al. studied the use of controlled experiments in software engineering (Sjoberg et al., 2005) and found in 2005 that only around 2% of papers in leading software engineering conferences and journals report controlled experiments. Molleri et al. conducted a systematic mapping study to summarize existing research guidelines, assessment instruments, etc. on how to conduct and evaluate empirical software engineering research (Molleri et al., 2019). Neto et al. explored the use of statistical methods in empirical software engineering research (Neto et al., 2019). In 2021, Wohlin analyzed 100 articles found through Scopus when searching for case studies in software engineering (Wohlin, 2021). Close to 50% of the articles investigated were judged as not being case studies according to the definition of a case study.

2.1.3. Literature reviews in specific subfields of software engineering

Several reviews have been presented to investigate empirical research in particular subfields of software engineering (including software architecture and other areas). For example, in 2014, Daneva et al. reviewed empirical work on software requirements engineering as one subfield of software engineering (Daneva et al., 2014). The study reported a significant growth in empirical requirements engineering research over the years with a noticeable jump in 2004 and 2005. Similarly, in 2016, Ambreen et al. conducted a systematic mapping study to analyze the state-of-the-art of requirements engineering and discussed various trends and opportunities of empirical requirements engineering research (Ambreen et al., 2018). Their analysis includes works published up to 2012 and found an increase in empirical works since around the year 2000, that requirements elicitation is the dominating research area, and that case study research dominates. They also point out a need for replication studies and more studies "in the real world" involving practitioners. Dyba and Dingsoyr investigated the status of empirical research on agile software development (Dyba and Dingsoyr, 2008) and concluded that there is a need for more and better empirical studies. In 2013, Weyns and Ahmad reported a systematic literature review on the validation of architecture-based approaches for self-adaptive systems (Weyns and Ahmad, 2013). They showed that only 2.5% of studies in this area applied systematic empirical methods. Recently, Gerostathopoulos et al. showed a significant increase in the use of systematic empirical methods in the area of self-adaptive systems during the past decade with an average of 52% of the studies using empirical methods (Gerostathopoulos et al., 2021). Finally, in 2020 Chacon-Luna et al. investigated empirical research in software product line engineering (Chacon-Luna et al., 2020). In a systematic literature review, the authors summarize studies that have used empirical evidences in software product line engineering, limited to those ones with the intervention of humans and published between 2000 and 2018. A total of 62 primary studies were selected out of which approximately 56% used the empirical case study strategy while the rest used experimental strategies. Around 86% of the case studies were performed in an industrial environment.

2.1.4. Mixed analyses

Mixed analyses explored how empirical research is applied in software engineering by combining research methods, e.g., a survey of researchers and a literature review. For example, in 2015, Siegmund et al. conducted a study on views of internal and external validity in software engineering research (Siegmund et al., 2015). Their work reviewed 405 papers from premier software engineering conferences (ICSE, ESEC/FSE and EMSE). That study reported an increasing trend of empirical research. The study also collected the perception of software engineering researchers on internal and external validity, and on empirical research in general. The study presented in this manuscript also presents a mixed analysis.

2.2. Empirical research in software architecture

In software architecture it has been claimed that it is hard to apply empirical methods (e.g., due to difficulties measuring architecture goodness, dependencies between design decisions, social factors, types of required subjects in studies with human participants) (Falessi et al., 2007). Qureshi et al. (2013) performed a systematic literature review to synthesize empirical research in software architecture and found that case studies an experiments are the most frequently used empirical methods. Our previous work explored the state-of-practice of empirical research in software architecture (Galster and Weyns, 2016). We conducted a survey of all full technical research papers published at major software architecture conferences between 1999 and 2015. Our study showed that 17% of papers report empirical work and that the number of empirical studies in software architecture has started to increase in 2005. Furthermore, we found that empirical studies are equally frequently used (a) to evaluate newly proposed approaches, and (b) to explore and describe phenomena to better understand software architecture practice. We found that case studies and experiments are the most frequently used empirical methods and that almost half of empirical studies involve human participants. Of the studies that involve human participants, the majority involve professionals rather than students. Another key finding from that study was that replications are almost non-existent in empirical software architecture research. Similarly, Konersmann et al. (2022) assessed evaluation support for replications in software architecture research. Their study analyzed full technical conference papers at the International Conference on Software Architecture and the European Conference on Software Architecture from 2017 to 2021. The study confirmed that experiments and case studies are most frequently used. The study also found that most papers explicitly acknowledge validity threats, but few papers include replication packages. The work presented in this paper complements these previous works that looked at the literature by capturing the perceptions of those who conduct and review empirical research in software architecture.

Razavian et al. analyzed empirical research for architecture decision making (Razavian et al., 2019) and identified various research designs and gaps in research focus and goals. The study also proposes a focus matrix and research cycle to position new research. A special issue of the Empirical Software Engineering journal was published in 2011 (Babar et al., 2011) and a workshop on Empirical Assessment in Software Architecture (EASA) was held at the Working Conference on Software Architecture

(WICSA) in 2008 and 2009. The contributions of these initiatives focused on presenting empirical research, rather than on reflecting on research and providing an overview of the perceptions of the research community.

We previously investigated issues specific to empirical software architecture research as a first step towards better understanding empirical research in software engineering in general (Galster et al., 2018). Based on feedback from software architecture researchers and practitioners, we explored why, despite persistent discussions in the research community, there are still disagreements about why and how to conduct empirical research. We observed that there is disagreement about the role and value of research methods. We also observed that study replications are appreciated, yet, they are rarely conducted. Finally, our previous study found that there is frustration towards the "so-called" empirical research due to low quality or misleading studies. This paper is based on collecting data from similar respondents as we used in Galster et al. (2018). However, in this paper we pursue a different goal and provide a deeper systematic analysis involving substantially more data, more questions and a combination of quantitative and qualitative analyses. Furthermore, we relate perceptions of those who conduct and evaluate research in software architecture to findings and recommendations reported in the literature.

3. Research method

In this section we present the details of the research method, including population and sampling, data collection and analysis.

3.1. Population and sampling

Our population includes those who conduct and review empirical software architecture research, i.e., "key players" who shape the future of research by deciding on research papers, guide young researchers and advise funding agencies. For a representative sample (Wagner et al., 2020), we collected the names and e-mail addresses of all program committee (PC) members of research tracks of all major architecture conferences (starting with their first instance up to 2017): (1) European Conference on Software Architecture (ECSA); (2) International Conference on Software Architecture (ICSA); (3) International Conference on the Quality of Software Architecture (QoSA); (4) International Symposium on Component-based Software Engineering (CBSE); (5) Working IFIP/IEEE Conference on Software Architecture (WICSA).[3] This resulted in a list of 454 unique names.[4] We had to exclude 19 names because we could not find their email addresses. This gave us a final list of 435 contacts. We could not control whether respondents were practitioners or academics, since their affiliation might have changed since they served on a PC. We used Microsoft Outlook to generate customized email messages asking for a response within two weeks. Furthermore, since we could not identify participants that had already responded, we sent a reminder to all potential respondents. We received fewer than 20 bounce messages from email addresses that were invalid as well as out-of-office replies. Note that even though respondents were recruited from program committees of software architecture conferences, all respondents also reviewed for other venues, see Section 4.1. Therefore, respondents are more likely to be aware of systemic problems with empirical research in software architecture, because they are also exposed to research in other areas of software engineering.

3.2. Survey instrument

We created parts of the survey instrument by leveraging instruments from other studies as discussed below. We included closed questions, but also allowed respondents to elaborate on their answers to closed questions in textual comments. Furthermore, we included several open questions. All questions were optional. We present all questions and answer options in the appendix.

1. The first part of the questionnaire solicited general demographic information, see Table A.13. Other studies (e.g., Siegmund et al. (2015)) collected fewer demographic information. This additional information allowed us analyze answers based on the background of respondents.
2. The second part of the questionnaire aimed at questions related to RQ1 and the understanding of respondents on how empirical software architecture research is applied, see Table A.14. This aspect was not explored in previous studies like Siegmund et al. (2015).
3. The third part of the questionnaire covered questions related to RQ2 and the perceptions of software architecture researchers about human participants, see Table A.15. This aspect has not been studied in previous works.
4. The fourth part of the questionnaire aimed at investigating RQ3 and the perceptions of researchers on internal and external validity in empirical software architecture research, see Table A.16. This aspect (except a question about the "dilema" between internal and external validity) was not included in the questionnaire of Siegmund et al. (2015).[5]
5. The fifth part of the questionnaire addressed RQ4 and the perceptions of software architecture researchers on replications, see Table A.17. We adjusted questions and answer options from Siegmund et al. (2015) to be specifically about software architecture. For example, Siegmund et al. (2015) included questions like "As a reviewer of a top-ranked conference..." which we adapted but changed to questions like "As a reviewer of a top-ranked software architecture conference...". Also, we removed the specific focus on experiments (we referred to "study" instead).

The questionnaire concluded with an open question: "Upon reflection of answering the questions, is there anything you can add and that you feel is relevant in the context of this questionnaire?"

Note that we intentionally formulated some of the questions in a "provocative" manner with the aim to trigger contrasting comments from participants. These questions may be subject to confirmation bias which we discuss in threats to validity.

We used a self-administered anonymous online questionnaire (LimeSurvey hosted by KU Leuven). The questionnaire was reviewed by two experts on questionnaire design who were not involved in this research. Furthermore, we conducted pilots with four members from the target population. Based on the feedback we optimized the questions, the structure of the questionnaire and the flow of questions.

3.3. Data analysis

To analyze closed questions and for quantitative data, we used descriptive statistics and quantitative data analysis. Therefore, we mostly report percentages relative to the respective

[3] QoSA, CBSE and WICSA merged in 2017 to become ICSA. We excluded the SEI Architecture Technology User Network (SATURN) conference due to its focus on practitioners.

[4] Data for our study are available online: https://doi.org/10.5281/zenodo.7762326

[5] In Q3.3 we ask about a "dilemma" between internal versus external validity. In general, to get generalizability we need large samples of data collected from realistic settings. Internal validity may require controlling variables, etc. making results potentially less realistic (compare for example a case study with a controlled experiment).

number of responses, frequency analyses, and "relationships" between questions based on contingency matrices (mostly based on the background of respondents as researchers, practitioners or both). We only report findings if they led to relevant insights. Note that the group of practitioners who responded is small compared to researchers. When the proportions (or groups) of participants can be compared, we present the results of z-tests with z-values $>$ 1.96 and significance level α = 0.05 as thresholds for all tests (Montgomery and Runger, 2018).[6]

To analyze comments to questions, we used qualitative data analysis. We used inductive reasoning (i.e., moving from specific fragments in comments to general concepts) to construct analytical codes and infer categories from the data by labeling occurrences (Stol et al., 2016). Similar to others (e.g., Prechelt et al. (2018)), we tried to keep coding simple. We did not have a pre-defined coding schema or a pre-defined granularity or semantic style for the codes. However, we interpreted comments in the context of the question for which they were given. We used a simple version of open coding (Strauss and Corbin, 1990). Similar to Mendez Fernandez et al. we used open coding to add codes to small coherent fragments of the comments (Fernández et al., 2016). We then categorized the developed concepts in a hierarchy of categories as an abstraction of the codes. We coded in a team of two coders more than 700 comments. Coding was done while being co-located, so emerging topics and concepts were discussed immediately. We excluded some comments from coding, e.g., if they did not provide any additional insights or if they were too generic, e.g., a respondent answering "Yes" to a closed question and stating "This is just my opinion" in the comments. Also, we did not check whether answers to a closed question was consistent with the comments for that question. For example, a respondent may have answered that she does not prefer one type of empirical method over the other, but in her comments indicate reasons why one "might" prefer one over the other (i.e., one comment may cover several concepts, which may not necessarily match the answer to the closed question). When reporting example quotes from comments in Section 4, we use verbatim excerpts, including spelling and punctuation mistakes.

4. Results

In this section we first provide demographic information of our data before presenting the results for each research question.

4.1. Demographic information

We received 105 valid responses,[4] i.e., a response rate of around 25% (this is higher than that of other studies, e.g., 10% in Siegmund et al. (2015)). We present their demographics below. We include tables with detailed results in the appendix.

- **Roles (Q0.1):** Seventy two respondents (69%) answered as researchers and 13 (12%) answered as practitioners. The remaining 20 respondents (19%) held both roles.
- **Industry and academic experience (Q0.2 and Q0.3):** Even though most respondents were researchers at the time of completing the questionnaire, most had worked more than ten years in industry, see Table B.18. On the other hand, 22% of respondents had never worked in industry. Only few respondents never spent time in academia, while the majority (72%) spent more than ten years in academia.

Table 1
Contingency matrix for Q1.1.

Role	Quantitative	Qualitative	No preference
Researcher	21	11	40
Practitioner	5	1	7
Both	0	4	16

- **Field of expertise (Q0.4):** As expected, the dominating primary field of expertise is software architecture (see Table B.19). "Other" expertises include performance engineering and dependability analysis. Few respondents indicated empirical software engineering as their primary field of expertise (this could be because empirical software engineering is considered cross-cutting different subfields of software engineering).
- **Reviewing activities (Q0.5):** On average (in addition to the conferences we used to identify participants), respondents reviewed for six different venues (*min* = 1, *max* = 15, *mean* = 7, *stdev* = 3.4), with JSS, TSE and Software as top venues. Table B.20 shows the venues that respondents reviewed for.
- **Publishing and reviewing empirical work (Q0.6 and Q0.7):** Most respondents had published and reviewed empirical studies, see Table B.21.

4.2. RQ1: Understanding of empirical research

4.2.1. Preferred type of research in general (Q1.1)

Overview: A majority of 63 respondents (60%) did not prefer any type of research method over the other. However, 26 respondents (25%) generally preferred quantitative methods and 16 (15%) preferred qualitative methods, hence, in total 42 participants stated that they prefer one method. While a significant group of respondents did not prefer a type of method (z-test compared to those that prefer a method: 2.04 $>$ 1.96 with α = 0.05), a relevant group of 40% of the respondents prefer a type of method. Table 1 shows the distribution based on roles.

Analysis of comments: Forty-six respondents (44%) provided additional textual comments. Based on the comments we identified reasons why respondents generally preferred one research method over the other or had no preference. We summarize the findings in Table 2. For each category (gray rows) and code, we include how often it appeared.

4.2.2. Preferred type of studies by role (Q1.2, Q1.4)

Overview: As can be seen in Fig. 1, as a reviewer (Q1.2), 52 respondents (50%) did not believe that quantitative studies are easier to review, while 22 respondents (21%) felt that quantitative studies are easier to review than qualitative studies. Similarly, from a researcher's and author's point of view (Q1.4), 44 respondents (42%) did not believe that quantitative studies are easier to get accepted.

Analysis of comments: Thirty-one respondents (30%) provided additional textual comments regarding their view as a reviewer (Q1.2). Based on the comments we identified reasons for why one might prefer reviewing quantitative studies or have no preference, see Table 3.

Similarly, thirty-two respondents (31%) provided additional textual comments regarding their view as an author (Q1.4). Again, we identified reasons for why one might prefer authoring one type of study over the other, see Table 4.

[6] We use the z-test to determine whether the answers of respondents in two groups of respondents are different: $z = (\hat{\mathrm{p}}-p)\ \sqrt{p(1-p) \div n}$ with $\hat{\mathrm{p}}$ as the proportion of the sample choosing one of the options, p as the proportion expected if there is no difference between selected options, and n as the sample size. We use a threshold z=1.96 with the conventional significance cutoff α = 0.05.

Table 2
Comments: Preferred type of research method **in general** (Q1.1).

Categories/codes	#	Example quotes
Why no preference		
Need mix of methods	16	"I am a strong supporter of Mixed-Methods research since neither qualitative nor quantitative research are enough by themselves."; "I strongly believe both sorts are needed and contribute different aspects to the research."
Method is context dependent	15	"The appropriate selection of a research method depends on the research goal."; "It really depends. Sometimes obtaining quantitative data is very difficult or it simply does not make sense for the situation that is being investigated."
Why qualitative		
Offer deeper insights	7	"I find the more interesting results via qualitative methods in general."; "Qualitative studies tend to give more insight."
Rely less on "hard" numbers	8	"There are cases where getting meaningful knowledge cannot be done through "hard" empirical studies."; "[...] quantitative [studies] give the illusion of more precision but are often not so useful."
Why quantitative		
Findings are traceable to data	6	"Evidence produced by quantitative research methods is more compelling."; "I like to see the data to understand how strong the research claims are."
Results are less subjective	4	"[...] qualitative is too subjective and not significant enough to generate a useful result."

Table 3
Comments: Preferred type of study **as reviewer** (Q1.2).

Categories/codes	#	Example quotes
Why no preference		
Problem-dependent	5	"It is important to use an appropriate research method."
Paper-dependent	6	"I like to review interesting studies, starting with interesting research questions. Then the research methodology is secondary."
Why quantitative		
Own knowledge and expertise	4	"I do prefer to review quantitative studies because I am working with them."
Clarity of data and methods	6	"Deriving conclusions from a meaningful measurable data makes it easier to identify threats to validity."

Table 4
Comments: Preferred type of study **as author** (Q1.4).

Categories/codes	#	Example quotes
Why no preference		
Problem-dependent	5	"I choose the research methodology according to the research questions and the opportunities that are available."
Venue-dependent	5	"[...] it really depends on the venue and on the claimed contributions of the submission."
Why quantitative		
Easier to write	3	"I find they are also a bit easier to write — esp for PHD students."
Clarity of data and methods	6	"As a piece of research, they tend to be more precise."

4.2.3. Change of views as reviewer (Q1.3)

Overview: Fifty-seven respondents (54%) stated that they have not changed their views on how they judge empirical research as reviewers over time, while 42 respondents (40%) have changed their views. Six respondents (6%) indicated that they never reviewed an empirical paper. While our data shows that a relevant number of respondents changed their view, the group of these respondents is not significantly different compared to others (z = 0.82, $alpha$ = 0.05).

Analysis of comments: Forty-eight respondents (46%) provided additional textual comments. However, insights from comments are rather anecdotal and we were not able to identify categories or common themes (e.g., regarding how or why views changed). For example, some of the comments indicate that respondents were always skeptical or always positive towards empirical studies, while other comments indicated that appreciation for empirical studies changed over time. For example, one respondent stated: *"I become more favorable to good empirical research (and possibly more harsh to lousy research)"*. Similarly, another one wrote: *"Given the limitations in finding real examples, I tend to accept less formal methods"*. On the other hand, some respondents

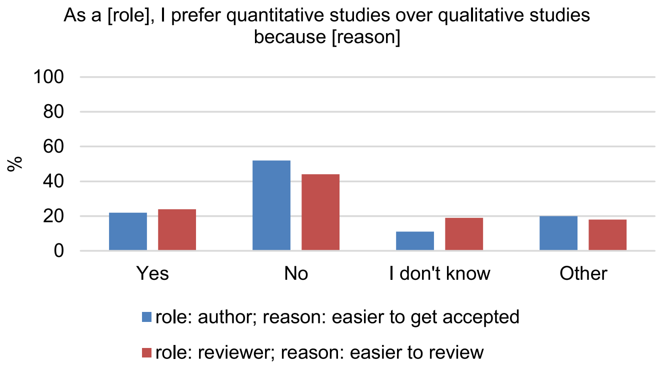

Fig. 1. Preferred type of study as reviewer (Q1.2) and author (Q1.4).

became more sceptical, e.g., *"I am now more biased against such work, because it is usually so contrived. A nice experiment, but a 'who cares?' result"*. Similarly, another respondent commented that *"I've become more critical of methods, sample size, relevance of many experimental setups when they don't reflect what might actual or plausible practice"*. highlighting the lack of practical relevance while another comment pointed out methodological issues: *"Mostly I have become more concerned with threats to validity issues, especially construct validity"*.

4.2.4. Examples of empirical architecture research (Q1.5)

Overview: We asked respondents whether they believe there are currently enough examples of good empirical software architecture research. Twenty-eight respondents (27%) believe that there are enough good examples of empirical software architecture research, while 39 respondents (37%) believe there are no good examples. Twenty-eight respondents (27%) indicated that they did not know, and ten respondents (9%) indicated "Other" while commenting on the meaning of "good" and "enough".

Analysis of comments: Forty-three respondents (41%) provided additional textual comments. Many comments confirmed the answer to the closed question, e.g., *"there are actually very few good ones"* without additional insights. The remaining comments mostly mention problems with current examples, but we were not able to identify re-occurring themes, e.g., types of problems with empirical studies. For example, one responded pointed out the lack of available study material in published empirical studies (*"We lack enough examples and, in particular, we need to have more researchers publishing all of their study materials/data so that others can attempt replication studies"*.) and poor reporting of studies: *"I can recall really few good example. Most of the others report only correlations or statistical tests (often used incorrectly) over subjects (either students or relatively few opensource projects) which are not representative"*. Another problem expressed in comments were unrealistic studies, e.g., *"Most empirical studies I've seen are on toy problems, when the real need is on realistically large scale systems and legacy-driven systems"*.

4.2.5. Summary for RQ1

Key insight(s) from RQ1:

1. The majority of members of the community do not prefer one type of research method over the other (either as reviewers or as authors). On the other hand, a relevant number of respondents (40%) do prefer either quantitative or qualitative research methods in general for various reasons.
2. Some of those who prefer quantitative methods in general argue that findings are traceable to data and results are less subjective. Some of those that prefer qualitative methods argue that they offer deeper insights and rely less on hard numbers.
3. A majority of the members of the community (a proportion that is not statistically significant) did not change their view over time on how they review empirical research. For those that changed their view, there are no common reasons why views changed.
4. Good examples of empirical research seem to be missing in software architecture.

Table 5
Contingency matrix for Q2.1.

Role	Professionals	Students	Mix
Researcher	33	2	32
Practitioner	10	0	2
Both	12	0	7

4.3. RQ2: Human participants

4.3.1. Preferred types of human participants (Q2.1)

Overview: Fifty-five respondents (52%) indicated a preference for professionals as participants in empirical software engineering studies. On the other hand, only two respondents (2%) indicated that they would prefer students. Forty-one respondents (39%) preferred a mixed set of participants, while seven (7%) preferred none of these options. The group of respondents that strictly prefers professionals as participants is not significantly different from the others (z = 0.10, α = 0.05). Table 5 shows responses based on the role of respondents.

Analysis of comments: Fifty respondents (48%) provided textual comments. Almost half of the comments (22 comments) state that in general practitioners are preferred, but the types of participants also depend on the goal of the study and that types of subjects could complement each other. The two respondents who preferred students indicated that students are easier to recruit; one of them stated that *"Professionals are not keen to participate in studies; typically, they are short-term goals oriented"*. and the other one wrote that *" Students are rather similar wrt. their background and skills, i.e., interpretation gets easier. Also professions often are hard to get"*. From the comments we identified reasons for why practitioners are preferred over students, see Table 6.

4.3.2. Value of empirical studies with students (Q2.2)

Overview: Most respondents disagree (29 respondents, 28%) or strongly disagree (15 respondents, 14%) that empirical studies with students have little or no value, see Fig. 2. However, a similar number of respondents agreed (23 respondents, 22%) or strongly agreed (17 respondents, 16%) with this statement. On the other hand, 20 respondents (19%) were neutral about this question and one respondent (1%) had no opinion. Note that in Fig. 2 we show net stacked distributions. These do not include neutral or "I don't know" answers, but results are centered on neutral to show if answers lean to one side (disagreement) or another (agreement). The grey bars in each direction in the stacked distributions represent increments of 20%, i.e., the outer most grey bar in each direction indicates 100%.

Analysis of comments: Fifty-eight respondents (55%) provided additional textual comments. From the comments we identified reasons for including and not including students in empirical studies.[7]

1. **Why include students:** We identified reasons for why students might be used as subjects in empirical software architecture research.
   (a) **Help achieve study goal:** Twenty-nine comments pointed out that students may be suitable to achieve the goal of a study. For example, students may be representative subjects for studies about novice developers. As one respondent wrote, *"They are valid as long as they are considered indicative of outcomes that can be obtained with novice programmers"*., echoed

[7] Unlike previous analyses of comments which were presented as tables, we here present a list to provide more contextual information.

Table 6
Comments: Why prefer professionals as human participants (Q2.1).

Codes	#	Example quotes
More experience	9	“students have no background in software architecture”; “[...] students often lack the sophistication and experience to understand the problems that our treatment addresses.”
Higher external validity	4	“Student-based studies I find to be highly suspect with major threats to external validity.”; “I personally don’t believe on studies that are based on students. Indeed we cannot generalise and in particular cases it might make sense. However, there is a kind of abuse of these studies that just refer to other empirical studies that under many threats to validity ‘demonstrate’ that it makes sense to do studies with students.”
Relevance and realism	4	“Research with [...] professionals has a far higher chance of being realistic and relevant to industry practice.”

Table 7
Contingency matrix for Q2.3.

Role	Possible	Impossible
Researcher	32	29
Practitioner	8	3
Both	9	4

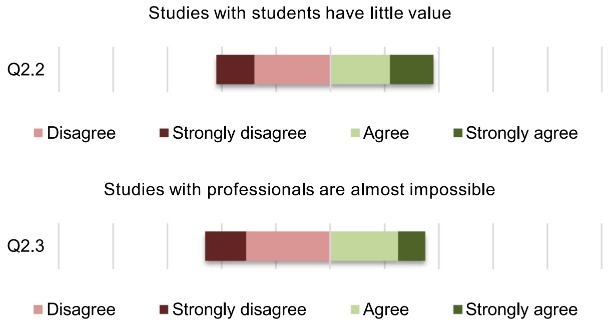

Fig. 2. Perceptions of respondents on the value of studies with students (Q2.2) and the difficulty of studies with practitioners (Q2.3).

in another comment: *“They are somehow novice, and this is valuable for the study”*. Similarly, another one wrote *“Unless the empirical study is about students”*. One respondent gave a concrete example: *“counter example: teaching techniques for architecture design decision-making”*. Finally, one respondent wrote that *“Students are developers/architects too. Additionally, if we don’t use them, we will never be able to pursue/evaluate certain kinds of research.”* and that *“questions that focus on industry practice can be at least partially answered with students”*.

(b) **Represent next generation of practitioners:** Five comments pointed out that students are the next generation of practitioners. As one respondent wrote, *“Especially Master Students are very close to professionals”*. Another one added that *“As they have similar profile and are the future industry workers, results are relevant”*. Another one stated that *“many students are practitioners themselves — in this field we have a tendency to assume age as a predictor of quality and reliability, and this is just plain bullshit”*. In a similar way, one respondent stated that *“One of the nonsense stereotypes. - Many of our students work in companies on the side. - So, they are semi-professionals. Also, the week after graduation they start to develop, in this week they will not become different developers”*.

(c) **Are only available subjects:** Four comments highlighted a more pragmatic view and pointed out that students might be the only subjects researchers have access to. As stated by one respondent, *“Studies with students are better than no studies at all, thus provide valuable results and inputs to validate with ‘harder-to-get’ professionals”*.

(d) **Are less biased:** Three comments emphasized that students, in contrast to practitioners, might be less biased. One respondent wrote that *“I believe there is a lot to be said for the unbiased approach of people that have studied the subject but are not yet clogged down by routine”*. Another one emphasized that *“results of praticioners which refuse to learn and apply new methods (maybe due to their age) can be useless too”*.

2. **Why not include students:** We also identified reasons for why students should not be used in empirical software architecture studies.

(a) **Not representative:** Nine comments highlighted that students are not representative to practitioners, therefore, results are not applicable in industry. As one respondent wrote, *“The main concern of using students only in an empirical study is that they may not represent the real practitioners for whom the results may be of interest”*. while another stated that *“This is about architecture, right? It’s even less likely that students are like professionals in architecture knowledge than in coding”*. Finally, one respondent wrote that *“In the software architecture domain in particular students aren’t representative of the target audience”*.

(b) **Lack experience:** Eight comments highlighted that students lack the experience required for software architecture studies to be meaningful. One respondent wrote that *“Might be of limited value in software architecture since software architecture typically requires years of experience. Students often lack experience with software architecture since student projects are usually small”*. Another one stated that *“Students have limited skills and experience, so results on field could be very different”*. Finally, one respondent wrote that *“The field of architecture is rather abstract and requires experience. It is hard to have students act as architects (or customers of architects)”*.

4.3.3. Empirical studies with professionals (Q2.3)

Overview: Most respondents disagree (33 respondents, 31%) or strongly disagree (16 respondents, 15%) that empirical studies with professionals are almost impossible, see Fig. 2. On the other hand, 26 (25%) agree and 10 (10%) strongly agree with that that statement, while 19 (18%) were neutral and one (1%) indicated “I don’t know”. Table 7 shows the distribution of agreements (“Impossible”) and disagreements (“Possible”) based on role.

Analysis of comments: Fifty-nine respondents (56%) provided additional textual comments. Based on the comments we identified reasons for why it is difficult to involve professionals in empirical software architecture research.

1. **Effort:** Nine respondents indicated that studies with practitioners are time-consuming for both researchers and participating professionals. As a respondent wrote, *" its not impossible, it simply takes much longer, much patience and a careful design which also includes practitioner benefit as a factor"*. Another one wrote that *"They [practitioners] are often too busy with their own commitments"*.
2. **Confidentiality:** Four respondents highlighted that involvement of professionals is constrained by confidentiality requirements. As one respondent wrote, *"I am a professional, and participate in a lot of research. Problem with professional engagement is not the research, but the implications it has on publication of the result"*. One respondent wrote that *"Non-disclosure agreements can limit the excitements for researchers in what they can share"*. Another one stated that *"... studying larger systems in practice often involves a lot of IP handling and other overhead"*.
3. **Missing value proposition:** Fifteen comments highlighted that studies with professionals need to clarify the value for participating professionals and organizations. As a respondent wrote, *"We should learn to offer values to companies, so they see real value in conducting this kind of experiments and studies"*.
4. **Required relationships:** Ten comments pointed out that involving practitioners requires good relationships with industry. As one respondent wrote, *"It is hard, but not impossible. A good business network is required though"*. Another one stated that *"We have good long-term collaborations with many companies, typically in more technical projects, but there is definitely room for empirical research as well"*.

4.3.4. Summary for RQ2

Key insight(s) from RQ2:

1. Professionals are in general the preferred type of human subjects in empirical architecture research. The main argument for including professionals is their experience. Yet, the group of respondents that strictly prefers professionals as participants is not significantly different from those that prefer students or mixed participants in studies.
2. There is no consensus on whether or not empirical studies with students provide value. The main argument in favour of students is that students may be suitable to achieve the study goal. Those not in favour point to their lack of experience and argue that students are not representative.
3. Studies with professionals seem feasible but difficult to conduct. The main arguments are that studies with professionals require a value proposition and good relationships with industry.

4.4. RQ3: Internal versus external validity

4.4.1. Validity in empirical studies (Q3.1, Q3.2)

Overview: As can be seen in Fig. 3, 37 respondents (35%) preferred maximizing external validity in studies *with* human participants, while 33 respondents (32%) preferred maximizing internal validity (Q3.1). On the other hand, 16 respondents (15%) would not prefer either and 19 (18%) indicated that they did not know. The z-test for respondents with a preference for maximizing one type of validity versus those that do not express a preference or "don't know" gives: $3.48 > 1.96$ ($\alpha = 0.05$). This shows that a significant group of participants prefer maximizing one type of validity in studies. Furthermore, as shown in Fig. 3, the distribution of preferences for external or internal validity in studies *without* human participants (Q3.2) was similar to Q3.1: Thirty-seven respondents (35%) indicated a preference for maximizing external validity and 32 respondents (31%) preferred to maximize internal validity. Seventeen respondents (16%) preferred none of the options and 19 (18%) indicated that they did not know. The z-test here gives a similar result: $4.64 > 1.96$ ($\alpha = 0.05$). Sixty-nine respondents (66%) provided the same answer for Q3.1 (validity in studies with human participants) and Q3.2 (validity in studies without human participants). In Table 8 we show the distribution of responses based on the respondents' role.

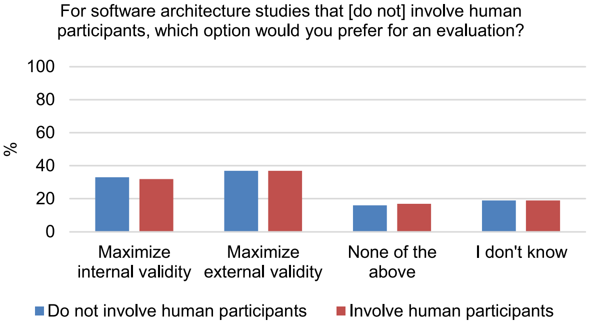

Fig. 3. Internal and external validity with (Q3.1) and without (Q3.2) human participants.

Table 8
Contingency matrix for Q3.1 and Q3.2.

Role	With humans (Q3.1)			Without humans (Q3.2)		
	External	Internal	No	External	Internal	No
Researcher	19	24	15	21	20	15
Practitioner	7	3	1	9	3	1
Both	11	6	3	7	9	1

Analysis of comments: Forty-six respondents (44%) provided additional textual comments regarding internal and external validity in studies with human participants (Q3.1). Eighteen comments indicated that validity types need to be balanced and depend on context. Based on the comments we also identified reasons why external or internal validity may be maximized, see Table 9.

Sixty-six respondents (63%) did not provide comments regarding internal and external validity in studies without human subjects (Q3.2), while 39 respondents (37%) provided additional textual comments. Twenty-eight respondents that provided comments gave the same answer for Q3.1 and Q3.2, confirming their answer to Q3.1 applies also to Q3.2. The remaining 11 respondents provided specific comments for Q3.2. This small number does not allow us to derive any insights in the answers to Q3.2 or the reasons why respondents changed their answer compared to Q3.1.

4.4.2. Dilemma of internal and external validity (Q3.3)

We asked participants whether they have any suggestions on how to resolve the dilemma of internal versus external validity in

Table 9
Comments: Validity in studies with human participants (Q3.1).

Categories/codes	#	Example quotes
Why external		
Reduced bias	3	"In my view [maximizing external validity] is the usually underdone aspect of most human-based SE studies - results simply fail to generalize given limited or biases sampling."
Value to practice	4	"Studies with high external validity yields more direct value to practitioners."; "Software architecture is paramount for practice, external validity and generalizability should have precedence over anything."
Why internal		
More meaningful results	5	"I would put more value on internal validity to have justified results."; "There are often multiple variables in human subjects. It is important to assure the validity of the collected data by aligning these variables. Internal validity is prime [...] without which the results would be meaningless."
Prerequisite for external validity	4	"If I had to choose one, I would strive for internal validity first, then external validity as a next step."; "internal is prerequisite for external validity."

software architecture research. This question was an open question and not mandatory. Fifty-three respondents (51%) provided textual comments. Based on these comments, we identified two categories:

1. **No suggestions:** 31 respondents (i.e., 30% of all respondents and 58% of those that provided textual comments for Q3.3) explicitly stated that they have no suggestions to resolve the dilemma of internal versus external validity in software architecture research. However, out of these, ten respondents explicitly stated that they do not see a dilemma at all. For example, one of the respondents stated: *"I see the two kinds of validity as orthogonal and I don't see any dilemma between them. Internal validity ensures that the results are correctly interpreted. External validity ensures that the results can be generalized"*. Another one wrote, *"To me it doesn't seem to be a dilemma. It depends on what you want to achieve. Internal validity is important when you're getting interesting propositions from a set of data. External becomes interesting if you want to generalize more mature hypotheses"*.
2. **Concrete suggestions:** Twenty-two respondents (i.e., 21% of all respondents and 41% of those that provided textual comments for Q3.3) provided concrete proposals to resolve the dilemma of internal versus external validity in software architecture research (one respondent may have provided more than one suggestion). Suggestions were rather diverse. For example, some respondents suggested to conduct multiple studies or replications, e.g., *"The only way is to perform similar studies in different sections. For example, I like to replicate student experiments with selected practitioners whenever applicable"*. or *"The only solution is extended research programs embodying many studies or experiments (chained together, so to speak)"*. Others suggest in general to apply different methods, e.g., *"Mixed methods research was specifically designed to address that dilemma, use quantity to address internal validity over qualitative results and involve practitioners [...] to strengthen external validity"*. Another one wrote, *"One might consider performing both a case study and a controlled experiment although it is costly"*.. Some respondents indicated that to resolve the dilemma, we may do whatever helps ensure practical relevance of a study, e.g., *"People who do this empirical research should pair up with a person doing *real* work so that they can understand when they are writing useless drivel"*. Finally, four respondents suggested to ensure internal validity first as a basis for any study. One respondent commented that *"Internal validate is a basic requirement — it must be present to analyze the soundness of the findings. The external validity is for the relevance"*.

4.4.3. Summary for RQ3

Key insight(s) from RQ3:

1. A significant group of respondents prefer either external or internal validity in studies. Yet, there is no agreement on whether internal or external validity is perceived as more important, regardless of whether or not a study involves human participants.
2. The main arguments in favour of external validity in studies with human participants are reduced bias and value to practice. The main arguments for internal validity are more meaningful results and that internal validity is a prerequisite for external validity.
3. Respondents did not have any strong opinions about how to resolve the dilemma between internal and external validity.

4.5. RQ4: Role of replications

4.5.1. Reviewing replicated studies (Q4.1 and Q4.2)

- **Number of reviewed replications (Q4.1):** The vast majority of respondents (80 respondents, 76%) had never reviewed a replicated study in a software architecture venue. Note that this number is based on self-reporting, i.e., respondents *may* have reviewed a replicated study, but (at the time of the participating in the survey) may not have remembered it (similarly, those who indicated that they had reviewed replicated studies may have done so for venues which are not about software architecture, but thought to remember that the reviews were for an architecture venue). On the other hand, 18 respondents (17%) had reviewed replicated studies once or twice, six respondents (6%) had reviewed more than twice but less than five times, and only one respondent (1%) had reviewed five or more replicated studies. This shows that the number of respondents who reviewed replicated studies is dropping fast with an increasing number of reviews. Also, these numbers show that reviewing replicated studies for software architecture venues is rather uncommon. No respondent provided additional textual comments.
- **Rating of replications (Q4.2):** Those who had reviewed replications also indicated how they rated those studies. Note that we asked respondents how they rate replications "in general", i.e., they may have not accepted/rejected all the replications they had reviewed.

Table 10
Comments: Replications to advance field (Q4.3).

Categories/codes	#	Example quotes
Why replications		
Increase confidence	16	"Replication is how we improve external validity (and to a lessor extent internal validity)."; "Replication is a way to improve external validity of a treatment."
Follow scientific principles	6	"I think that replication studies are the heart and soul of science, so this kind of work is desperately needed in architecture research."; "Science (and validation) is about replication of experiments."
Difficulties		
Definition of replication	11	"it may be doubtful that any useful question in software architecture can be addressed in a replicated experiment."
Execution of replication	9	"often, a replication exposes serious problems with the original work"; "[there are] too many uncontrolled factors to expect replication."
Publication of replication	6	"I can't imagine getting a replicated study through a reviewing process."; "I've heard (first-hand) that replicated user studies are hard to get accepted."

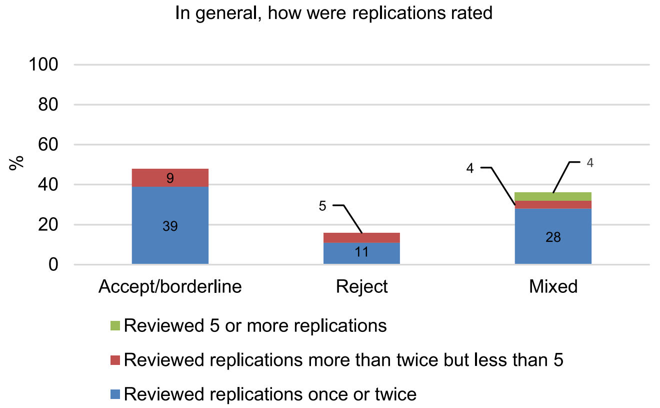

Fig. 4. Number of reviews done by respondents (based on Q4.1) and the rating of reviewed replications (Q4.2).

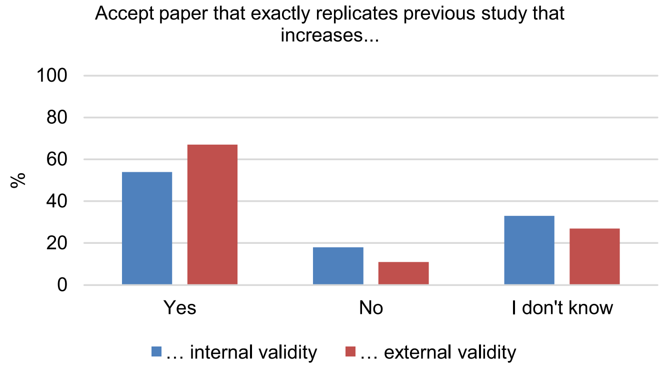

Fig. 5. Agreement of respondents on accepting replications that increase internal validity or external validity of a study (Q4.4 and Q4.5).

Fig. 4 shows our findings for Q4.1 and Q4.2. These responses show that there is no correlation between the number of replications reviewed by the respondents and the ratings they gave them. None of the respondents provided additional textual comments.

4.5.2. Need for replications (Q4.3)

Overview: Regarding the need for the publication of replications, the vast majority of respondents (80 respondents, 76%) confirmed that there is a need to publish more replication studies in software architecture.

Analysis of comments: Forty-eight respondents (45%) provided textual comments that fell into two categories which were somewhat orthogonal to the responses to the closed question: (1) Comments that *motivate* replications, and (2) comments about *difficulties* with replications, see Table 10. The two motivators (increase confidence, follow scientific principles) provide complementary views on why we should conduct replications. While the first point expresses a more pragmatic view on why replications should be done, the second one expresses a more philosophical view. Furthermore, difficulties raised in comments range from fundamental concerns (defining and executing replicated studies) to more pragmatic concerns (publishing studies).

4.5.3. Internal (Q4.4) and external (Q4.5) validity

Overview: Fifty-four respondents (51%) answered "yes" to whether they would accept a paper at a top architecture venue that presents an exact replication of a previous study to increase internal validity (Q4.4). On the other hand, 18 respondents (17%) would not accept such a paper and 33 respondents (31%) answered "I don't know". (see Fig. 5). Similarly, 67 respondents (64%) answered "yes" to whether they would accept a paper at a top architecture venue that presents an exact replication with as contribution an increase of external validity. On the other hand, eleven respondents (10%) would not accept such a paper and 27 (26%) answered "I don't know". When comparing these numbers to the answers about internal validity (Q4.4, see Fig. 5), we notice an increase of "yes" answers for Q4.5 (which asked about an increase of external validity) and a decrease of "no" answers.

Analysis of comments: Fifty-four respondents (51%) added textual comments to their answer for Q4.4. We identified codes that grouped comments into comments that argue that replications need to be of good quality and comments that argue that replications need to offer new insights. We were not able to assign codes to 17 of the comments (for example, one respondent stated *"I could be more interested in innovation".*). We also noticed that four respondents commented on the type of publication of a replication (or where and how to publish a replication that only increases internal validity). For example, one respondent wrote that *"I would take it into a journal, but not a conference"*, while another one stated: *"Publish a short article to critique on a published article"*. Furthermore, five respondents commented on the question itself and the question's intent. From these respondents, one answered "yes", two answered "no", and two answered "I don't know". For example, one of the respondents stated: *"I do not think 'exact' replication is possible, especially if humans are involved"*.

Regarding Q4.5, 46 respondents (44%) added textual comments to their answers. Of those, 24 provided exactly the same comment as for Q4.4 and 10 only changed the wording "internal validity" with "external validity" in their comments. Twelve respondents changed their comments for Q4.5 compared to Q4.4. This low number did not allow us to derive categories or codes.

4.5.4. Summary for RQ4

Key insight(s) from RQ4:

1. Those that conduct software architecture empirical research rarely see replication studies to review; there is no real difference between how replication studies are treated in the evaluation.
2. "Increase of confidence" and "at the heart of science" are the main motivations for replications to advance the field.
3. Three difficulties with replications are pointed out: define replication studies, do them, and publish the results.

5. Discussion

Our paper empirically analyzes the perception of those who conduct and evaluate empirical work in software architecture. In particular, we aim at offering empirical evidence to overcome anecdotal opinions for how the community perceives empirical research. Some of our findings may not be surprising, but as argued by Torchiano and Ricca, reporting on the reality may not always offer surprising insights (Torchiano and Ricca, 2013). Yet, empirical evidence offers a scientific basis to better understand reality and advance the aspects under study. In this section, we discuss our results.

5.1. RQ1: Understanding of empirical research

We found that most respondents did not prefer a particular research method and acknowledge that the type of research method should be motivated by the research problem at hand (see Section 4.2). Reasons for or against qualitative and quantitative methods echo those reported in the research method literature (Felderer and Travassos, 2020). Also, some of our findings and comments from participants reflect "bad smells" that Menzies and Shepperd identified in software analytics papers (computational experiments, correlation studies), e.g., findings that are not interesting or simplified assumptions (Menzies and Shepperd, 2019). This indicates a mature understanding of how and why to select research methods by the majority of participants. On the other hand, we also found that a substantial number of respondents (40%) do prefer either quantitative or qualitative research methods in general for various reasons, which may point to a less mature understanding of selecting research methods by this relevant group of respondents.

Interestingly, respondents found that there are not many good examples of empirical architecture research (see Q1.5 in Section 4.2.4). Our respondents represent the community, including the ones who do conduct studies in software architecture. When they say that there are only a few good examples published, it may imply that they are not always satisfied with the empirical papers that they or others publish. Understanding this better could be subject to future work.

One of the implications of these findings could be that those who review empirical research should not only rely on their domain expertise, but also have an understanding and appreciation for the type of research method used in a paper. While this applies to any research domain, this also has implications for journal editors and chairs of conference program committees: Research method expertise adds another dimension to the selection and assignment of reviewers (in addition to domain expertise). This is also reflected in current initiatives to improve the quality of peer reviews in software engineering, such as the ACM SIGSOFT Paper and Peer Review Quality Initiative (Ralph and Robbes, 2020). This initiative aims at providing guidelines to establish empirical standards. Authors on the other hand can support editors, chairs and reviewers by clearly stating research method and approach in the abstract and keywords of manuscripts. As mentioned before, respondents believe that there are not enough papers with good examples of empirical research in software architecture. This may inhibit writing or reviewing such papers, in particular for researchers new to empirical methods. On the other hand, looking on guidelines for systematic reviews, Kitchenham and colleagues found that even though reviewers use common guidelines and common evaluation criteria, assessments can still differ, e.g., if criteria are not understood by reviewers (Kitchenham et al., 2012).

5.2. RQ2: Human participants

In our own previous literature survey (Galster and Weyns, 2016) we found that human participants are used most in empirical software architecture studies, with professionals as the dominating type (mostly in surveys and case studies, but less so in experiments). This is also reflected in the preference of our respondents (see Q2.1 in Section 4.3). We found that students are clearly not preferred in empirical software architecture research. On the other hand, we could not find any conclusive answer to whether or not the community considers empirical studies with students of value (see Q2.2 in Section 4.3).

Falessi et al. analyzed the views of empirical software engineering experts on the use of students (Falessi et al., 2018) and argued that no population (students, professionals, or others) can be deemed better than another in absolute terms. The goal of an empirical study can include different target populations and subjects, ranging from students and novice architects up to experienced practitioners. Hence, using students may not be a "valid simplification of reality", but a valid way to achieve the goal of a study which relies on the skills set of the target population. For example, experience in programming and practicing programming may improve programming skills, but this may not be true for higher level and more abstract software architecting tasks which involve trade-offs and are less concrete.

Also, the focus on professionals seems to imply that empirical software architecture research needs to be relevant and potentially applicable to industry (Garousi et al., 2020) and be based on industry-academia collaborations (Garousi et al., 2019). This seems to ignore (or at least underestimate) the need for fundamental research, even in an applied field such as software architecture. Also, it may imply that students as study participants do not allow us to achieve industry-relevant research. However, software architecture research can also be about studying novices or less skilled practitioners (who would probably benefit from new techniques and solutions most and do not have an established way of working yet) rather than experts, or about teaching and training of software architecture. In fact, only involving high-performing and experienced professionals may not be realistic and hence reduce relevance of a study for practice. Also, the assumption that professionals provide the required skills and background for industry relevance of research might be misleading. For example, Sjoberg and Bergersen in a commentary on Feldt et al. argued that skills of undergraduate students, graduate students, junior, intermediate and senior professionals may overlap for the populations of students and professionals (Feldt et al., 2018). Furthermore, sometimes professionals are not always the most knowledgeable experts (in other words, experience may not be the same as expertise) or more qualified than less experienced ones, as for example shown in a study by Wan et al. about the (wrong) perceptions of professionals on defect prediction (Wan et al., 2018).

We also investigated difficulties with professionals (Q2.3 in Section 4.3). Our finding that around half of our respondents find it impossible to involve professionals in empirical studies partially confirms the perceptions of software engineering experts on the use of professionals (Falessi et al., 2018). However, we cannot compare our results to Falessi et al. since their 14 statements about the use of professionals are much more fine-grained than our analysis. On the other hand, the reasons for why it is difficult to include professionals in empirical studies identified from the comments that respondents provided in our study (time, effort, etc.) are also raised in their study (Falessi et al., 2018) and follow-up commentaries in Feldt et al. (2018). In our study a variety of arguments are given to underpin the difficulty of involving professionals as subjects in empirical studies. The arguments raise questions about (i) the relevance of research with respect to current practice both for researchers and practitioners, and (ii) the relationship between researchers and practitioners.

In the broader context of software engineering, Storey and colleagues investigate how much software engineering research addresses human and social issues, and explore how much software engineering research studies include human and social aspects in research designs (Storey et al., 2020). Their study found that although many papers claim the contributed research should benefit human stakeholders, most focus predominantly on technical contributions. Their results suggest a need for strategies that aim at a deeper understanding of human and social aspects of software development practice to balance the design and evaluation of technical innovations.

5.3. RQ3: Internal versus external validity

Our findings on internal versus external validity partially confirm what others have found related to empirical research in software engineering in general. In summary, we found that there is no agreement on whether to maximize internal or external validity and the impact of human study participants on validity. Previous works, for example Siegmund et al. also identified a mixed degree of awareness of the trade-offs between internal and external validity of empirical studies (Siegmund et al., 2015). In contrast to that study, our work focuses on one subfield of software engineering and shows that even within a field, and a more closed community, different perceptions exist. In the area of programming languages, another software engineering subfield, Stefik and Hanenberg argue that studies rely on weak methodological procedures to a point that it is doubtful whether they could detect fraud, let alone be replicated by other scholars and argue for more rigorous evidence standards (Stefik and Hanenberg, 2017). This may also be the case for software architecture as our findings indicate that in software architecture no expectations exist what makes good evidence subject to validity threats.

To systematically address validity issues, different proposals have been made in other communities, e.g., pre-registered reports which encourage researchers to rigorously describe and review their study design before actually conducting a study (Cockburn et al., 2018) and guidelines for the evaluation of empirical work.[8] These could also apply to software architecture research. As found in a review by Konersmann et al. (2022), such guidelines are not yet systematically used when discussing validity threats in software architecture papers.

Similarly, internal and external validity could be improved through replications. Regarding increasing validity through reproducability, Kitchenham et al. (2020) raised concerns about the analysis and meta-analysis of experiments and, in a literature review, identified families of experiments that used meta-analysis, to investigate their methods for effect size construction and aggregation, and to assess the reproducibility and validity of their results. They were able to only reproduce five out of 13 studies. Kitchenham et al. conclude that meta-analysis is not well understood by software engineering researchers and present recommendations for reporting and meta-analyzing families of experiments. We discuss replications further in Section 5.4.

5.4. RQ4: Role of replications

As mentioned above, our work focuses on one subfield of software engineering and shows that even within a subfield (and a more closed community), different perceptions on the usefulness of replications and related difficulties exist.[9] Siegmund et al. identified mixed perceptions on the value of replications in software engineering in general (Siegmund et al., 2015). Konersmann et al. found that around 11% of papers at ICSA and ECSA 2017 to 2021 provide a dedicated and available replication package (Konersmann et al., 2022). Comparing the perceptions of members of the architecture community to what we have previously found in a study of the literature (Galster and Weyns, 2016), we note that replications are almost non-existent at software architecture venues. There could be several reasons for this. First, replications could be published at other (non-software architecture) venues and therefore not appear in typical architecture conferences such as ICSA or ECSA. Second, replications might be published in journals since the space required to properly report a replication would exceed the typical page limit of conferences. Third, the community simply does not publish replications (for several reasons). This is also indicated by the fact that 76% of respondents never reviewed a replication (see Q4.1). On the other hand, those who had reviewed replications were willing to accept them. Therefore, researchers should not worry too much about getting potentially rejected “just because” their study is a replication (see Q4.2). Our study shows that reviewers consider other factors as well when reviewing replications (quality of study, etc.). Finally, members of the community see the value of replications (Q4.3, Q4.4 and Q4.5).

The state of replications in software engineering research has been explored in various previous literature reviews. For example, Bezerra et al. in 2015 found that the number of replications increased significantly between 2011 and 2012 compared to the time before 2010 (Bezerra et al., 2015), mostly as external replications conducted by researchers not involved in the original study. More recently, Cruz et al. observed that this trend continued and is reflected in an increasing trend in the number of published replications in 2013–2018 (Cruz et al., 2020). Cruz et al. also pro-

[8] See for example https://www.sigarch.org/a-checklist-manifesto-for-empirical-evaluation-a-preemptive-strike-against-a-replication-crisis-in-computer-science/

[9] Note that we do not include a discussion about repeatability, reproducability and replicability; the definition of these concepts are subject of discussion, e.g., the evolution of terms from Version 1 to Version 1.1 of ACM's badging system which swapped the definitions of reproducibility and replicability https://www.acm.org/publications/policies/artifact-review-badging

vide an overview of related reviews on replications in software engineering.

In the context of software engineering in general, others (e.g., Shepperd (2018)) have argued that replications are usually confirmatory (Q4.4 and Q4.5 in our study indicate that replications are more accepted if they help increase generalizability), but that contributions to knowledge are negligible. Therefore, researchers should apply meta-analysis which combines all relevant/related studies to increase sample sizes and to estimate the magnitude of differences between groups (e.g., based on effect size in experiments (Shepperd, 2018)). Recently, Santos et al. proposed guidelines and an analysis procedure specifically tailored to the analysis of groups of software engineering replications (Santos et al., 2021).

Note that we do not argue that replications should dominate publications. However, if replications were a common practice, more than 24% of respondents would have reviewed a replication. Also, we would like to highlight that replications are just a means to achieve reliability and confidence of research findings, i.e., replications should serve a purpose and not be conducted just for the sake of a replication (or "another paper").

Authors of original studies can support replications by following guidelines for reproducible research, e.g., (ROpenSci, 2020). Also, explicit reviewer training (as proposed for conferences like the International Conference on Computing Education Research, ICER (Ko, 2019)) might help.

5.5. Types of reviewers

Based on our data we could also identify "types of reviewers" or profiles of respondents and how they approach empirical work in software architecture. These types of reviewers could be used as a starting point to define representative reviewer behavior, including the values, goals and limitations of reviewers, in order to guide decisions about the design of a review process. We could draw from the idea of reviewer profiles in consumer reviews and how organizations recognize and influence reviewer behavior. For example, Mathwick and Mosteller identified factors that motive online reviewer engagement considering altruistic and egoistic motives and classified reviewers (Mathwick and Mosteller, 2018).

To identify reviewers' types, we "clustered" respondents with similar answers based on questions Q0.1, Q0.2, Q0.3, Q0.6, Q0.7, Q1.1, Q2.1, Q2.2, Q2.3 and Q3.1. This allowed us to analyze responses beyond two-dimensional contingency matrices. We grouped respondents if we found at least seven respondents with similar characteristics.[10] Numbers in brackets indicate how often we found that type of reviewer amongst our respondents.

- **Type 1 — The mature researcher (15):** This reviewer has more than ten years of experience in total, has published empirical research studies and reviewed them. Furthermore, this type of reviewer does not prefer any research method over the other, and, when human participants are involved, prefers a mix of student and industry participants and believes that results of empirical studies with students have value.
- **Type 2 — The mature practitioner (8):** This type of reviewer has more than 10 years of experience in industry and has reviewed empirical papers. This reviewer prefers professionals as study participants and sees little or no value in studies with students. This reviewer believes that empirical studies with professionals are feasible.
- **Type 3 — The quantitative-oriented reviewer (8):** This type of reviewer has more than 10 years of experience in total, prefers quantitative over qualitative research methods. This type mostly prefers studies with professional participants, and prefers to maximize external over internal validity.
- **Type 4 — The qualitative-oriented reviewer (8):** This type of reviewer has more than 10 years of experience in total. This reviewer prefers qualitative over quantitative methods in studies with practitioners or a mix of practitioners and students. Furthermore, this reviewer prefers maximizing internal validity.
- **Type 5 — The practice-oriented reviewer (7):** This reviewer has more than ten years of experience in total and has reviewed empirical studies. This reviewer also prefers empirical studies with professionals and even agrees that results of empirical studies with students have little or no value. Also, this reviewer prefers external over internal validity. This may indicate an interest in novelty rather than scientific rigour. Furthermore, this may indicate a lack of appreciation for internally valid studies and that external validity or practical relevance of a study is seen as most important to make software architecture research "relevant".
- **Type 6 — The balanced reviewer (12):** This reviewer has at least 10 years of experience in total, and has published and reviewed empirical papers. This reviewer sees value in studies with students and believes that studies with professionals are feasible. This reviewer is not inclined to prefer internal or external validity in studies.

We summarize these types of reviewers in Table 11. We split Table 11 into two subtables for readability ('–' in a row indicates that the answer to the respective question is not used to characterize the type of reviewer). The above types of reviewers were identified in 58 respondents (four respondents map to two types). The remaining 47 respondents could not be clustered into any of these categories due to their diversity. These reviewer profiles could help utilize engagement theory in the academic peer review process by linking motives with levels of behavioral engagement in the review process (rather than only characterizing reviewers based on the content of their reviews, such as reviewers who are overcritical, unrealistic or focused on minor issues such as typos in manuscripts). Thus, profiles may help "review managers" (journal editors, PC chairs, etc.) tailor reviewing environments to attract and retain a diverse and engaged reviewing community. As has been shown in consumer reviews, not all reviewers share the same external and internal motivators for their behavior. Therefore, it has become of crucial importance to understand why and how reviewers contribute (Mathwick and Mosteller, 2018).

5.6. Comparison with published works

As discussed in Section 2, in our previous work published in 2016 we reviewed publications at major software architecture conferences (Galster and Weyns, 2016). That review was rather broad and when applicable we already reflected on findings from that review in the context of this survey in previous sections. Also, other existing reviews, e.g., Konersmann et al. (2022) explore a similar topic, but have different research questions and results that are not easily comparable to the research questions of our study and the survey questions. Therefore, in this section we provide a more targeted comparison of the research questions and the goal of our survey with the published software architecture research. This is to provide further insights about what the community thinks is happening (based on the survey reported in this paper) and the actual architecture research. We therefore

[10] The six types of reviewers emerged during the analysis and forming groups with at least seven respondents provided a good balance between descriptive types and granularity of profiles.

Table 11
Types of reviewers (R = Researcher; P = Practitioner; Exp. = Experience; I = Industry; A = Academia)

(a) Types of reviewers — part a

Type	Role (Q0.1)	Exp. I (Q0.2)	Exp. A (Q0.3)	Exp. Total	Published (Q0.6)	Reviewed (Q0.7)	Preferred methods (Q1.1)
Type 1	R	–	–	>10	Yes	Yes	No preference
Type 2	P	>10	–	–	Yes	–	–
Type 3	–	–	–	>10	–	–	Quantitative
Type 4	–	–	–	>10	–	–	Qualitative
Type 5	–	–	–	>10	–	Yes	–
Type 6	–	–	–	>10	Yes	Yes	–

(b) Types of reviewers — part b

Type	Preferred human subjects (Q2.1)	Students (Q2.2)	Professionals (Q2.3)	Maximum validity (Q3.1)
Type 1	Mix	Value	–	–
Type 2	Professionals	No value	Feasible	–
Type 3	Professionals	–	–	External
Type 4	Mix	–	–	Internal
Type 5	Professionals	No value	–	External
Type 6	–	Value	Feasible	No

analyzed more recent research published at the International Conference on Software Architecture (ICSA).

5.6.1. Review process

We reviewed papers published from the first instance of the ICSA (2017) until 2021. This time period also aligns with the period of our survey. We followed a procedure similar to our previous work (Galster and Weyns, 2016) and analyzed all empirical papers published at ICSA. Therefore, we manually analyzed each full paper in the research track to decide whether or not a paper presents empirical work. We analyzed the whole paper, rather than only the title, abstract or keywords. Also, we did not only rely on methods claimed in papers. For example, a paper may claim to present a "case study" even though it may present an illustrative example (Wohlin, 2021). On the other hand, some papers do not use common terminology of empirical research or strictly follow empirical guidelines, but still present findings based on systematically collected and analyzed data. Therefore, we used a broader definition of empirical research as suggested by Sjoberg et al. (2007): Empirical research includes any research based on systematically collected evidence. It "seeks to explore, describe, predict, and explain natural, social, or cognitive phenomena by using evidence based on observation or experience. It involves obtaining and interpreting evidence, by, e.g., experimentation, systematic observation, interviews or surveys, or by the careful examination of documents or artifacts". As a consequence, we not only checked whether there was a well-established empirical method, but used as inclusion criteria (a) whether there was a clear study objective, (b) whether data was collected and analyzed systematically, (c) if findings were linked back to the objective, and (d) if there were insights beyond the studied case or example and offer insights about the claim in a paper. All of these criteria had to be met by a study. Furthermore, if the authors claimed a method, we looked for clues that the claim about the method was true (e.g., references to literature related to the claimed method). Similarly, exclusion criteria were (a) papers that discuss how to do empirical research, and (b) papers that provide infrastructures for empirical research and use examples of empirical studies for illustration purposes.[11] Two researchers reviewed papers.

5.6.2. Analyzed data

For each "empirical" paper we extracted information that was directly related to the research questions of our survey.

Regarding survey RQ1 (understanding of empirical research) we collected the following data:

- Empirical method used: case study, experiment, quasi-experiment, interview study, survey, systematic literature study (systematic literature review or mapping study, but not informal literature reviews), multi-method study, systematic empirical enquiry. This list is based on our previous work (Galster and Weyns, 2016). We included "systematic empirical enquiry" for work that does not follow the structure or design of well-established methods, but still formulates objectives, collects and analyzes data, and presents insights in a systematic manner (including linking findings back to the objective).
- Purpose of empirical study: focus, evaluation (an empirical study could be used to explore a problem, e.g., in a descriptive survey, or to evaluate a new approach (e.g., an experiment to compare a new technique with an alternative technique).
- Type of collected/raw data: qualitative, quantitative, mixed. Note that type of collected (raw) data is about the data used as a starting point for the analysis and not about data that are generated as part of the data analysis, such as numbers obtained as part of analyzing qualitative data, e.g., frequencies of codes/labels obtained during open coding).

Related to survey RQ2 (human participants) we recorded whether a study involved human participants or not. A study may involve no human participants, professionals, students, or students and professionals (mixed). Furthermore, in some studies the type of participants may be unclear (e.g., a study that analyzes Stack Overflow posts that could be written by either students or professionals).

Regarding survey RQ3 (internal versus external validity) we extracted the following data:

- Types of validity threats discussed (as reported by authors rather than our own interpretation): We differentiated internal, external, construct, and conclusion validity and other types of validity. Furthermore, we indicated whether the discussion of validity threats was generic and did not refer to any of the usual types. Finally, we recorded if there was no discussion of validity threats at all.
- How validity threats are reported: To get an idea how explicitly validity threats are in papers, we recorded whether there is a dedicated section for validity threats or if they are reported as part of a general discussion.

Finally, related to survey RQ4 (role of replications) we looked at the following data:

[11] An example is "Constructing a Shared Infrastructure for Software Architecture Analysis and Maintenance" at ICSA 2021.

- Does the paper report a replication: yes, no
- How is the replication reported: If a replication is reported it may be the focus of the paper (e.g., a dedicated replication study) or part of another study (e.g., a replication as part of a comparative study).
- How does the paper motivate the replication: If a replication was reported, we recorded if there was a motivation for the replication and what the motivation was.
- Does the paper in any way acknowledge the need for a replication of the work presented in the paper (this goes beyond general statements such as future work that is needed): For any paper we checked whether there was any mention of the need for a replication as part of planned/future work or by including a replication package (we did not apply specific criteria for the replication package and we also did not assess the quality of a replication package).

5.6.3. Limitations of literature review

Limitations related to the review are that we relied on the reporting in papers. Also, researchers may have been biased when judging whether a paper is empirical or not, in particular if papers did not report established method and study designs. To increase confidence, two researchers reviewed and discussed papers.

5.6.4. Analyzed papers of literature review

We analyzed a total of 95 papers (2017: 17; 2018: 22; 2019: 21; 2020: 19; 2021: 16). Out of these, 55 (58%) were classified as empirical. This increasing number compared to our previous study (Galster and Weyns, 2016) (17% of papers at CBSE, ECSA, QoSA and QoSA from 2004 to 2015) could indicate that more good examples of empirical works become available, one issue raised by the participants of our survey. The list of 95 papers and the extracted data is available online.[4]

5.6.5. RQ1: Understanding of empirical research

We found that the dominating research method (29 of 55) in ICSA papers was "systematical empirical enquiry" (recall that we used that category for empirical work that does not follow a well-established empirical method, but still formulates objectives, collects and analyzes data, and presents insights in a systematic manner). These papers may range from some pseudo-case studies to systematic measurements. On the other hand, we found only one case study (note that the term "case study" might be used in other papers, but they do not comply with guidelines for empirical case studies, a finding similar to what other have reported (Wohlin, 2021)). We found three experiments and four quasi-experiments, one survey (excluding literature surveys), six interview studies (they differ in our review from surveys in that they use interviews as data collection and use a relatively small sample), and four literature studies (these included systematic mapping studies and systematic literature reviews or grey literature reviews, but not informal literature reviews). Finally, we found seven multi-method studies (combine two or more research methods in one paper).

In most papers, an empirical study was used to evaluate a newly proposed approach (32 out of the 55 empirical papers), while the rest of the papers presented empirical work as the focus, i.e., the empirical work was the core of the paper with the empirical results as the main contribution. Finally, most papers (28 of 55) utilize quantitative data, while 19 consider qualitative data and eight papers consider both. Note that this is about the raw data collected and not the analysis of data.

The above findings reflect the overall trend of perceptions expressed in the survey that research methods should be balanced. Further, the analysis of published empirical work indicates that different empirical methods are used for different goals and purposes, again something that is reflected in the perceptions of survey respondents.

5.6.6. RQ2: Human participants

Regarding the human participants (RQ2 of our survey) we found that 33 of the 55 empirical papers do not include human participants. Of the remaining 22 paper, 15 involve practitioners while five involve students and two papers both.

Interestingly, respondents indicated that they prefer professionals. Also, even though participants of the survey feel that it is difficult to conduct studies with practitioners, many published studies do invest the effort to conduct such studies.

5.6.7. RQ3: Internal versus external validity

Regarding validity (RQ3 of our survey), we found that only six papers do not discuss validity threats. External validity (30 of the 55 papers), construct validity (24 papers) and internal validity (23 papers) are discussed most. This finding is similar to Konersmann et al. who found that most papers of ICSA and ECSA 2017 to 2021 that do acknowledge validity threats acknowledge external validity (Konersmann et al., 2022). Furthermore, papers discuss conclusion validity (10) and reliability (4). Nineteen papers do not refer to any validity type but discuss validity in a more generic sense. Two papers discuss statistical and content validity. Interestingly, most papers (46) include a dedicated section to discuss validity threats while three papers "weave" the discussion of validity threats into general discussions about limitations. While internal validity is mentioned less frequently than external validity, this could be due to the fact that internal validity may not be a concern for all types of research and the dominating research method in the set of analyzed studies was systematic empirical enquiry. For example, according to the ACM Empirical Standard, "Internal validity only applies to explanatory case studies that seek to establish causality". Therefore, some studies may not discuss internal validity. We only found a few experiments for which internal validity was reported as a primary concern.

The above findings show that published research acknowledges validity threats. Reflecting on our findings from the survey that there is no agreement whether internal or external validity should be maximized, it appears that in practice published research acknowledges external validity more frequently than internal validity.

5.6.8. RQ4: Role of replications

Finally, regarding replications (RQ4 of our survey), we found no replication. Few papers even acknowledge replications: Seven papers provide a replication package, three papers explicitly refer to replications when discussing validity threats, and two papers mention replications as future work. This is again similar to Konersmann et al. (2022) who found few replication packages in recent ICSA and ECSA papers.

Interestingly, most respondents consider replications essential for scientific progress. However, this is not reflected in the practice of software architecture research.

5.6.9. Summary of comparison

In Table 12 we compare the high-level findings of the survey (i.e., the perceptions of the community) with our literature review (i.e., actual research practices).

5.7. Threats to validity

In this section we discuss threats to the validity of the survey. We discussed limitations of the light-weight literature review separately in Section 5.6.3.

Table 12
Comparison between survey and literature review of high-level findings regarding the survey research questions.

RQ	Survey	Published research
RQ1	The use of different research methods *should* be balanced	The use of different research methods *is* balanced
RQ2	Professionals are the preferred type of subjects	Professionals are frequently involved as subjects
RQ3	No preference of internal or external validity	External validity is reported more
RQ4	Replications are important	Replications are not published

5.7.1. Sampling bias and response bias

We applied purposive sampling to identify "key players" in the field of software architecture who conduct, publish and review empirical works. However, there might be others in the communities who also publish and review empirical works. Furthermore, those who have strong opinions on the topic of the study may have more likely responded than those who are neutral or do not have particular views. However, such responses would probably have led to more neutral answers to our questions.

5.7.2. Internal validity

In our study we do not make any claims about causality, but report the perception of those who conduct and review empirical software architecture research. We analyzed correlation (see "profiles" of reviewers above), but not to show causality, but to characterize potential types of reviewers. Also, note however, that the group of practitioners is small compared to researchers, so we cannot draw statistically significant conclusions. Regarding the years of experience in industry and academia, we could not find any interesting results. However, other third-variable explanations cannot be discounted. Other confounding variables might include individual differences (e.g., personality, research philosophy, academic rank), research culture at institution or country of respondents, etc. Some of the questions in the questionnaire may have been rather abstract. Examples are Q4.4 (As a reviewer of a top-ranked software architecture conference, would you accept a paper that, as the main contribution exactly replicates a previously published study that increases internal validity?) and Q4.5 (As a reviewer of a top-ranked software architecture conference, would you accept a paper that, as the main contribution exactly replicates a previously published study that increases external validity?). Our purpose was to understand the perceptions around replications to improve validity. Any refinement of such questions may either result in a bias towards particular topics or lead to more questions. We acknowledge that the responses to these types of questions should be considered rather general.

5.7.3. External validity

We invited program committee members only, who are responsible for reviews. We acknowledge that some may not be very active in conducting research themselves. Also, our sample was drawn from members of the software architecture community and may therefore not be representative of the population of software engineering researchers or researchers in other fields of software engineering, e.g., requirements engineering. Therefore, our study may be somewhat imprecise regarding the populations it represents. However, as can be seen in the demographics of our sample, respondents were not only active in the field of software architecture and reviewed software architecture research, but also worked in other fields of software engineering. Furthermore, we included a control question (Q0.8). Responses to this question indicate that most respondents do not see a difference in empirical research in architecture and other field. Only twenty-one respondents (20%) consider empirical research in software architecture as different to empirical research in other fields of software engineering. For example, one respondent stated that *"Our contextual issues may be broader, but I don't see any essential difference"*. Another one asked *"Why should it be different?"* and another one pointed out that *"If people think that there is sth special about empirical research and architecture, they are researchers on empirical studies (but not on software architecture) from my point of view"*.

5.7.4. Construct validity

Questions on our questionnaire might have been too simple (or sometimes even "provocative" or "leading") and may have encouraged respondents into a simplified answer (e.g., Q1.2, Q1.4, Q3.3). Also, such questions may be subject to confirmation bias. However, this was (to a certain degree) on purpose since we hoped to (a) identify whether respondents notice this (and comment on it), and (b) trigger more detailed comments from participants.

Furthermore, respondents may have misinterpreted questions. Most questions required basic knowledge of empirical research terminology (e.g., internal validity, replications). Practitioners without research experience may have mis- or not understood some questions and therefore provided misleading answers. Similarly, despite piloting the questions, some wording of questions may have been difficult to interpret. E.g., in Q4.4 and Q4.5 we ask "would you accept a paper that [...] exactly replicates a previously published study that increases external validity?" It may be unclear (without considering context) which of both studies increases the validity.

Finally, respondents may have thought about software engineering in general, while our questions were about software architecture. Therefore, some responses may reflect the opinions (or practices) in software engineering rather than software architecture. On the other hand, we included a final question to identify any issues respondents had with the questionnaire. However, no issues were raised.

5.7.5. Reliability of qualitative analysis

Since we analyzed and coded data from anonymous respondents, we could not validate concepts with respondents. We did not perform member checking (Creswell and Miller, 2000) to check the categories for *resonance* to make sure that the findings from the qualitative analysis make sense to respondents since the responses were anonymous and we could not relate individual statements to respondents.

Also, as argued by others (e.g., Fernández et al. (2016)), coding is a creative task and influenced by subjectivity of coders (based on experience, expertise, expectations and maybe the researchers' opinions). To mitigate these problems, data analysis was performed while researchers were co-located in one room to have frequent and detailed discussions about the codes, the concepts that emerged, the level of abstraction of codes/concepts, to explicitly state (and discuss) biases, as well as discussions about how to code particular comments.

The qualitative analysis of textual comments from respondents also raises questions about *credibility* (i.e., whether there is sufficient data to merit claims). We do not claim credibility beyond the sample, since our sample was rather small (compared to the population of software engineering researchers) and not all respondents did provide textual comments. Some "categories" that we identified were only found in the responses from a small number of respondents.

Finally, all quotes are opinions and express the perception of individuals. Therefore, we refrain from generalizations based on individual comments. For example, one quote included in Section 4.2 above states that *"I can recall really few good example. Most of the others report only correlations or statistical tests (often used incorrectly) over subjects (either students or relatively few opensource projects) which are not representative"*. This may be an opinionated and superficial generalization (some respondents may not have taken the time to really consider whether the methods used were sufficient to answer the RQs posed by those papers).

6. Conclusions

In this paper we analyzed the perception of the software architecture research community on empirical research. We conducted a questionnaire-based survey and based on the responses from 105 participants we found that the perceptions differ quite a bit, in particular regarding the role of replications. Referring to the contributions outlined in the introduction (Section 1) we found the following:

1. Overview of the perception on how empirical research in software architecture should be conducted:
    (a) Most respondents do not prefer quantitative or quality research methods, yet, a relevant group does prefer one method over the other in general.
    (b) Professionals are the preferred type of human participants.
    (c) There is no agreement on what type of validity (internal or external) to optimize.
    (d) The role of replications and how they are valued by the community is unclear.
2. For many aspects related to conducting empirical research in software architecture there is no agreement amongst members of the community.
3. Overall the views of members of the community indicate a good understanding of trade-offs involved in empirical research.
4. Comparing the view of the community with published research at ICSA, the premier conference for software architecture research, showed that the views of the community partially differ to practices applied in published research.

To summarize (and referring to how contributions support the target audience as discussed in the introduction), the actionable insights provided by our study are:

- Researchers who conduct and review empirical research can see that reviewers may have different expectations about an empirical paper and how to judge it. The types of reviewers described in Section 5.5 generalize different expectations.
- Reviewers can benchmark themselves against others, reflect on their views on empirical research compared to others in the research community, and initiate actions in the community to resolve disagreements. We discuss such actions and initiatives in Section 5.1 (general review initiatives), Section 5.3 (guidelines for empirical work) and Section 5.4 (initiatives around reproducible research).
- Educators can identify issues and potential pitfalls to consider when training (new generations of) researchers. For example, to help students learn about empirical research (by looking at existing studies in detail, guided by their supervisors) and at the same time to establish replications as valid and necessary research, the students could conduct a replication study as a first milestone.
- Emerging and less experienced researchers get insights about the role of empiricism in software architecture research, in particular on the aspects for which there is consensus among experienced researchers, but also the aspects for which there is disagreement. Our work can guide reviewing activities as well as expectations of researchers on empirical software architecture research (Section 5). We also showed that what researchers think should happen does not always align with what does happen (see also Table 12). Emerging and less experienced researchers should not feel obliged to always present "original" research, but also pursue replication studies to contribute to building a strong body of knowledge in software engineering and software architecture in particular.
- Practitioners can use findings to inform their view about the value and purpose of empirical software architecture research. E.g., as we show in Table 11, reviewers may have a practitioner's view when reviewing research papers. Yet, when interpreting empirical results, practitioners can use findings that point to limited consensus on aspects of empirical research in software architecture.

Future work could obtain insight from other software engineering research communities about empirical practices. This could help address potential confirmation bias and whether there are systemic problems with empirical research in software architecture. Findings from other communities could then be compared with findings from software architecture. Furthermore, we can conduct literature reviews like the one done in this paper to understand what type of research is published in other fields of software engineering. This would help us identify different "practices of doing research" in different fields of software engineering. Last but not least, the outcome of the study may be used as a starting point for a reflection on empirical research aiming to resolve conflicting opinions and outline paths for improvement for the future, e.g., through a dedicated workshop or a seminar.

CRediT authorship contribution statement

Matthias Galster: Design, execution and analysis of the reported research, Writing – original draft, Revising the manuscript. **Danny Weyns:** Design, execution and analysis of the reported research, Writing – original draft, Revising the manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data are available in an online repository.

Acknowledgments

We thank the respondents. Also, we thank the anonymous reviewers for their feedback that helped us to significantly improve the manuscript.

Table A.13
Questionnaire: Demographic information.

ID	Question	Response options
Q0.1	When answering this questionnaire, do you consider yourself as a software engineering researcher or practitioner?	Researcher; Practitioner; Both
Q0.2	How many years have you worked in the software industry?	0; <1; 1–3; 4–5; 6–10; >10
Q0.3	How many years have you worked in academia?	0; <1; 1–3; 4–5; 6–10; >10
Q0.4	What is your primary software engineering field of expertise?	Requirements engineering; Architecture/design; Programming languages; Software testing; Evolution and maintenance; Software processes; Empirical software engineering; Other (free text)
Q0.5	Have you ever served as a reviewer for the following software engineering journals or conferences?	TOSEM; EMSE; Software; TSE; IST; JSS; ASE; EASE; ECOOP; FSE; ESEM; GPCE, ICPC, ICSE; ICSME; OOPSLA/SPLASH; RE
Q0.6	Have you ever published an empirical study?	Yes; No
Q0.7	Have you ever reviewed an empirical study?	Yes; No
Q0.8	In your opinion, is empirical research in software architecture different to empirical research in other fields of software engineering?	Yes; No; I don't know

Table A.14
Questionnaire: Understanding of empirical architecture research (RQ1).

ID	Question	Response options
Q1.1	In general, I prefer...	...quantitative research methods over qualitative methods; ...qualitative research methods over quantitative methods; ...no type of research method over the other
Q1.2	As a reviewer, I prefer quantitative studies over qualitative studies because they are easier to review.	Yes; No; I don't know; Other (free text)
Q1.3	As a reviewer, over time, have you changed your views on how you judged empirical research in software architecture?	Yes; No; I have never reviewed an empirical paper
Q1.4	As a researcher/author, I prefer quantitative studies over qualitative studies because it is easier to get them accepted.	Yes; No; I don't know; Other
Q1.5	I believe there are currently enough good examples of good empirical software architecture research.	Yes; No; I don't know

All questions were single choice.

Table A.15
Questionnaire: Human participants in empirical studies (RQ2).

ID	Question	Response options
Q2.1	In general, what kind of human participants would you prefer in empirical software architecture studies?	Professionals; Students; Mixed; None of the above
Q2.2	Please indicate your agreement with the following statement: Results of empirical studies with students have little or no value.	Strongly agree; Agree; Neutral; Disagree; Strongly disagree; I don't know
Q2.3	Please indicate your agreement with the following statement: Performing empirical studies with professionals is almost impossible.	Strongly agree; Agree; Neutral; Disagree; Strongly disagree; I don't know

All questions were single choice.

Table A.16
Questionnaire: Internal and external validity (RQ3).

ID	Question	Response options
Q3.1	For software architecture studies that involve human participants, which option would you prefer for an evaluation?	Maximize internal validity; Maximize external validity; None of the above; I don't know
Q3.2	For software architecture studies that do not involve human participants, which option would you prefer for an evaluation?	Maximize internal validity; Maximize external validity; None of the above; I don't know
Q3.3	Do you have any suggestions on how to resolve the dilemma of internal versus external validity in software architecture research?	Free text

All questions were single choice.

Appendix A. Questionnaire

Regarding roles (Q0.1), respondents who were no longer active would choose their most recent role(s). Regarding work experience in industry (Q0.2), we acknowledge that not all experience in the software industry is relevant to this research (e.g., experience as a first line support technician or salesperson may not inform a view about our research questions). On the other hand, this is mitigated by selecting respondents who were all affiliated with the field of software architecture research. The venues in Q0.5 were not used to identify potential participants, but to see whether participants are used to reviewing high quality research beyond software architecture venues. We chose them based on their high ranking according to the Computing Research and Education Association of Australasia (CORE). All questions but Q0.5 were single choice.

Appendix B. Demographic information

See Tables B.18–B.21.

Table A.17
Questionnaire: Role of replications (RQ4).

ID	Question	Response options
Q4.1	How often have you reviewed a replicated study in software architecture venues?	Never; Once or twice; More than twice but less than 5; 5 or more
Q4.2	In general, how were the replications rated by you?	Accept; Borderline; Reject; Mixed; I have never reviewed a replication study
Q4.3	Do you think we need to publish more replications in software architecture?	Yes; No
Q4.4	As a reviewer of a top-ranked software architecture conference, would you accept a paper that, as the main contribution exactly replicates a previously published study that increases internal validity?	Yes; No; I don't know
Q4.5	As a reviewer of a top-ranked software architecture conference, would you accept a paper that, as the main contribution exactly replicates a previously published study that increases external validity?	Yes; No; I don't know

All questions were single choice.

Table B.18
Respondents' years in industry and academia (Q0.2 and Q0.3).

Years	Industry (Q0.2)	Academia (Q0.3)
0	23 (22%)	7 (7%)
<1	0 (n/a)	0 (n/a)
1–3	20 (19%)	2 (2%)
4–5	9 (9%)	6 (6%)
6–10	11 (10%)	14 (13%)
>10	42 (40%)	76 (72%)

Table B.19
Primary fields of expertise (Q0.4).

Field of expertise	%
Architecture/design	75
Other	9
Software testing	6
Evolution and maintenance	4
Programming languages	2
Software processes	2
Empirical software engineering	2
Requirements engineering	1

Table B.20
Conferences and journals participants reviewed for (Q0.5).

Venue	%	Venue	%
1. JSS	81	10. ECOOP	15
2. TSE	67	11. ESEM	15
3. Software	61	12. OOPSLA/SPLASH	13
4. IST	53	13. ICSME	12
5. ICSE	49	14. GPCE	10
6. TOSEM	48	15. RE	8
7. FSE	38	16. ICPC	7
8. ASE	37	17. EASE	6
9. EMSE	34		

Table B.21
Publishing and reviewing empirical work (Q0.6 and Q0.7).

	Yes	No
Published empirical study (Q0.6)	72 (69%)	33 (31%)
Reviewed empirical study (Q0.7)	92 (88%)	13 (12%)

References

Ambreen, T., Ikram, N., Usman, M., Niazi, M., 2018. Empirical research in requirements engineering: Trends and opportunities. Requir. Eng. 23, 63—95.

Babar, M.A., Lago, P., v. Deursen, A., 2011. Empirical research in software architecture: Opportunities challenges and approaches. Empir. Softw. Eng. 16, 539–543.

Baker, M., 2016. Is there a reproducibility crisis? Nature 33, 452–454.

Banovic, N., 2015. To replicate or not to replicate? GetMobile: Mob. Comput. Commun. 19, 23–27.

Basili, V., Briand, L., Bianculli, D., Nejati, S., Pastore, F., Sabetzadeh, M., 2018. Software engineering research and industry: A symbiotic relationship to foster impact. IEEE Softw. 35, 44–49.

Bass, L., Clements, P., Kazman, R., 2012. Software Architecture in Practice. Addison-Wesley Professional.

Bezerra, R., Silva, F., Santana, A., Magalhaes, C., Santos, R., 2015. Replication of empirical studies in software engineering: An update of a systematic mapping study. In: ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. ESEM.

Briand, L., Bianculli, D., Nejati, S., Pastore, F., Sabetzadeh, M., 2017. The case for context-driven software engineering research: Generalizability is overrated. IEEE Softw. 34, 72–75.

Chacon-Luna, A., Gutierrez, A., Galindo, J., Benavides, D., 2020. Empirical software product line engineering: A systematic literature review. Inf. Softw. Technol. 128.

Cockburn, A., Dragicevic, P., Besancon, L., Gutwin, C., 2020. Threats of a replication crisis in empirical computer science. Commun. ACM 63, 70–79.

Cockburn, A., Gutwin, C., Dix, A., 2018. HARK no more: On the preregistration of CHI experiments. In: Conference on Human Factors in Computing System. CHI, pp. 1–12.

Creswell, J.W., Miller, D.L., 2000. Determining validity in qualitative inquiry. Theory Into Prac. 39, 124–130.

Cruz, M., Bernardez, B., Duran, A., Galindo, J., Ruiz-Cortes, A., 2020. Replication of studies in empirical software engineering: A systematic mapping study, from 2013 to 2018. IEEE Access 8, 26773–26791.

Cunningham, W., 1992. The WyCash portfolio management system. In: Object-Oriented Programming Systems, Languages, and Applications. OOPSLA, pp. 29–30.

Daneva, M., Damian, D., Marchetto, A., Pastor, O., 2014. Empirical research methodologies and studies in requirements engineering: How far did we come? J. Syst. Softw. 95, 1–9.

de Magalhaes, C., Silva, F.D., Santos, R., Suassuna, M., 2015. Investigations about replication of empirical studies in software engineering: A systematic mapping study. Inf. Softw. Technol. 64, 76–101.

Dyba, T., Dingsoyr, T., 2008. Empirical studies of agile software development: A systematic review. Inf. Softw. Technol. 50, 833–859.

Falessi, D., Babar, M.A., Cantone, G., Kruchten, P., 2010. Applying empirical software engineering to software architecture: Challenges and lessons learned. Empir. Softw. Eng. 15, 250–276.

Falessi, D., Cantone, G., Kazman, R., Kruchten, P., 2011. Decision-making techniques for software architecture design: A comparative survey. ACM Comput. Surv. 43, 1–30.

Falessi, D., Juristo, N., Wohlin, C., Turhan, B., Munch, J., Jedlitschka, A., Oivo, M., 2018. Empirical software engineering experts on the use of students and professionals in experiments. Empir. Softw. Eng. 23, 452–489.

Falessi, D., Kruchten, P., Cantone, G., 2007. Issues in applying empirical software engineering to software architectures. In: 1st European Conference on Software Architecture. ECSA, pp. 257–262.

Felderer, M., Travassos, G., 2020. Contemporary Empirical Methods in Software Engineering. Springer.

Feldt, R., Zimmermann, T., Bergersen, G., Falessi, D., Jedlitschka, A., Juristo, N., Munch, J., Oivo, M., Runeson, P., Shepperd, M., Sjoberg, D., Turhan, B., 2018. Four commentaries on the use of students and professionals in empirical software engineering experiments. Empir. Softw. Eng. 23, 3801—3820.

Fernández, D.M., Wagner, S., Kalinowski, M., Felderer, M., Mafra, P., Vetrò, A., Conte, T., Christiansson, M., Greer, D., Lassenius, C., Männistö, T., Nayabi, M., Oivo, M., Penzenstadler, B., Pfahl, D., 2016. Naming the pain in requirements engineering. Empir. Softw. Eng. 22, 2298–2338.

Galster, M., Weyns, D., 2016. Empirical research in software architecture – how far have we come? In: 13th Working IEEE/IFIP Conference on Software Architecture. WICSA, pp. 11–20.

Galster, M., Weyns, D., Tang, A., Kazman, R., Mirakhorli, M., 2018. From craft to science: The road ahead for empirical software engineering research. In: 40th International Conference on Software Engineering: New Ideas and Emerging Results. ICSE-NIER, pp. 77–80.

Garlan, D., 2014. Software architecture: A travelogue. In: Future of Software Engineering. FOSE, pp. 29—39.

Garousi, V., Borg, M., Oivo, M., 2020. Practical relevance of software engineering research: Synthesizing the community's voice. Empir. Softw. Eng. 25, 1687—1754.

Garousi, V., Pfahl, D., Fernandes, J., Felderer, M., Mäntylä, M., Shepherd, D., Arcuri, A., Coskuncay, A., Tekinerdogan, B., 2019. Characterizing industry-academia collaborations in software engineering: Evidence from 101 projects. Empir. Softw. Eng. 24, 2540—2602.

Gerostathopoulos, I., Vogel, T., Weyns, D., Lago, P., 2021. How do we evaluate self-adaptive software systems? In: International Symposium on Software Engineering of Adaptive and Self-Managing Systems. SEAMS, pp. 59–70.

Höfer, A., Tichy, W., 2007. Status of empirical research in software engineering. In: Basili, V., Rombach, D., Schneider, K., Kitchenham, B., Pfahl, D., Selby, R. (Eds.), Empirical Software Engineering Issues – Critical Assessment and Future Directions. Springer, pp. 10–19.

Ivarsson, M., Gorschek, T., 2011. A method for evaluating rigor and industrial relevance of technology evaluations. Empir. Softw. Eng. 16, 365–395.

Kitchenham, B., Madeyski, L., Brereton, P., 2020. Meta-analysis for families of experiments in software engineering: A systematic review and reproducibility and validity assessment. Empir. Softw. Eng. 25, 353–401.

Kitchenham, B., Sjoberg, D., Dyba, T., Pfahl, D., Brereton, P., Budgen, D., Höst, M., Runeson, P., 2012. Three empirical studies on the agreement of reviewers about the quality of software engineering experiments. Inf. Softw. Technol. volume 54, 804–819.

Ko, A., 2019. Changes coming to the ACM ICER conference. https://medium.com/bits-and-behavior/changes-coming-to-the-acm-icer-conference-494c808766dd (Accessed: 07 Oct 2020).

Konersmann, M., Kaplan, A., Kühn, T., Heinrich, R., Koziolek, A., Reussner, R., Jürjens, J., al Doori, M., Boltz, N., Ehl, M., Fuchs, D., Groser, K., Hahner, S., Keim, J., Lohr, M., Saglam, T., Schulz, S., Toberg, J.-P., 2022. Evaluation methods and replicability of software architecture research objects. In: 19th International Conference on Software Architecture. ICSA, pp. 157–168.

Ledo, D., Houben, S., Vermeulen, J., Marquardt, N., Oehlberg, L., Greenberg, S., 2018. Evaluation strategies for HCI toolkit research. In: 2018 CHI Conference on Human Factors in Computing Systems. CHI, pp. 1–17.

Lo, D., Nagappan, N., Zimmermann, T., 2015. How practitioners perceive the relevance of software engineering. In: 10th Joint Meeting on Foundations of Software Engineering. FSE, pp. 9–19.

Maniadis, Z., Tufano, F., List, J., 2017. To replicate or not to replicate? Exploring reproducibility in economics through the lens of a model and a pilot study. Econ. J. 127, F209–F235.

Mathwick, C., Mosteller, J., 2018. Online reviewer engagement: A typology based on reviewer motivations. J. Serv. Res. 20, 204–218.

Menzies, T., Shepperd, M., 2019. Bad smells in software analytics papers. Inf. Softw. Technol. 112, 35–47.

Molleri, J., Petersen, K., Mendes, E., 2019. CERSE - catalog for empirical research in software engineering: A systematic mapping study. Inf. Softw. Technol. 105, 117–149.

Montgomery, D.C., Runger, G.C., 2018. Applied Statistics and Probability for Engineers. Wiley.

Morgan, D., 2018. Living within blurry boundaries: The value of distinguishing between qualitative and quantitative research. J. Mixed Methods Res. 12, 268–279.

Neto, F.O., Torkar, R., Feldt, R., Gren, L., Furia, C., Huang, Z., 2019. Evolution of statistical analysis in empirical software engineering research: Current state and steps forward. J. Syst. Softw. 156, 246–267.

Perry, D., Wolf, A., 1992. Foundations for the study of software architecture. ACM SIGSOFT Softw. Eng. Notes 17, 40–52.

Prechelt, L., Graziotin, D., Fernandez, D.M., 2018. A community's perspective on the status and future of peer review in software engineering. Inf. Softw. Technol. 95, 75–85.

Qureshi, N., Usman, M., Ikram, N., 2013. Evidence in software architecture, a systematic literature review. In: 17th International Conference on Evaluation and Assessment in Software Engineering. EASE, pp. 97–106.

Rabiser, R., Schmid, K., Becker, M., Botterweck, G., Galster, M., Groher, I., Weyns, D., 2018. A study and comparison of industrial vs. Academic software product line research published at SPLC. In: 22nd International Conference on Systems and Software Product Line Engineering. SPLC, pp. 14–24.

Ralph, P., Robbes, R., 2020. The ACM SIGSOFT paper and peer review quality initiative: Status report. ACM SIGSOFT Softw. Eng. Notes 45, 17–18.

Razavian, M., Paech, B., Tang, A., 2019. Empirical research for software architecture decision making: An analysis. J. Syst. Softw. 149, 360–381.

Rodriguez-Pacrez, G., Robles, G., Gonzalez-Barahona, J.M., 2018. Reproducibility and credibility in empirical software engineering: A case study based on a systematic literature review of the use of the SZZ algorithm. Inf. Softw. Technol. 99, 164–176.

ROpenSci, 2020. Reproducibility guide. https://ropensci.github.io/reproducibility-guide/ (Accessed: 07 Oct 2020).

Runeson, P., Höst, M., 2009. Guidelines for conducting and reporting case study research in software engineering. Empir. Softw. Eng. 14, 131–164.

Sale, J., Lohfeld, L., Brazil, K., 2002. Revisiting the quantitative-qualitative debate: Implications for mixed-methods research. Qual. Quant. 36, 43–53.

Salman, I., Misirli, A.T., Juristo, N., 2015. Are students representatives of professionals in software engineering experiments? In: 37th International Conference on Software Engineering. ICSE, pp. 666–676.

Santos, A., Vegas, S., Oivo, M., Juristo, N., 2021. A procedure and guidelines for analyzing groups of software engineering replications. IEEE Trans. Softw. Eng. 47, 1742–1763.

Shaw, M., 2003. Writing good software engineering research papers. In: 25th International Conference on Software Engineering. ICSE, pp. 726–736.

Shepperd, M., 2018. Replication studies considered harmful. In: 40th International Conference on Software Engineering: New Ideas and Emerging Results. ICSE-NIER, pp. 73–76.

Siegmund, J., Siegmund, N., Apel, S., 2015. Views on internal and external validity in empirical software engineering. In: 37th International Conference on Software Engineering. ICSE, pp. 9–19.

Sjoberg, D., Dyba, T., Jorgensen, M., 2007. The future of empirical methods in software engineering research. In: Future of Software Engineering. FOSE, pp. 358–378.

Sjoberg, D., Hannay, J., Hansen, O., Kampenes, V., Karahasanovic, A., Liborg, N., Rekdal, A., 2005. A survey of controlled experiments in software engineering. IEEE Trans. Softw. Eng. 31, 733–753.

Stefik, A., Hanenberg, S., 2017. Methodological irregularities in programming-language research. IEEE Comput. 50, 60–63.

Stol, K., Fitzgerald, B., 2018. The ABC of software engineering research. ACM Trans. Softw. Eng. Methodol. 27.

Stol, K., Ralph, P., Fitzgerald, B., 2016. Grounded theory in software engineering research: A critical review and guidelines. In: 38th International Conference on Software Engineering. ICSE, pp. 120–131.

Storey, M., Ernst, N., Williams, C., Kalliamvakou, E., 2020. The who, what, how of software engineering research: A socio-technical framework. Empir. Softw. Eng. 25, 4097–4129.

Strauss, A., Corbin, J., 1990. Basics of Qualitative Research: Grounded Theory Procedures and Techniques. SAGE.

Tang, A., Kazman, R., 2017. On the worthiness of software engineering research.

Theisen, C., Dunaiski, M., Williams, L., Visser, W., 2018. Software engineering research at the international conference on software engineering in 2016. ACM SIGSOFT Softw. Eng. Notes 42, 1–7.

Tofan, D., Galster, M., Avgeriou, P., Schuitema, W., 2014. Past and future of software architectural decisions – a systematic mapping study. Inf. Softw. Technol. 56, 850–872.

Tofan, T., Galster, M., Avgeriou, P., Weyns, D., 2011. Software engineering researchers' attitudes on case studies and experiments: An exploratory survey. In: 15th International Conference on Evaluation and Assessment in Software Engineering. EASE, pp. 1–5.

Torchiano, M., Ricca, F., 2013. Six reasons for rejecting an industrial survey paper. In: 1st International Workshop on Conducting Empirical Studies in Industry. CESI, IEEE, pp. 21–26.

Wagner, S., Mendez, D., Felderer, M., Graziotin, D., Kalinowski, M., 2020. Challenges in survey research. In: Travassos, G., Felderer, M. (Eds.), Contemporary Empirical Methods in Software Engineering. Springer, pp. 93–125.

Wan, Z., Xia, X., Hassan, A., Lo, D., Yin, J., Yang, X., 2018. Perceptions, expectations, and challenges in defect prediction. IEEE Trans. Softw. Eng. 1–26.

Weyns, D., Ahmad, T., 2013. Claims and evidence for architecture-based self adaptation - a systematic literature review. In: 7th European Conference on Software Architecture. ECSA, pp. 249–265.

Wohlin, C., 2021. Case study research in software engineering—It is a case, and it is a study, but is it a case study? Inf. Softw. Technol. 133.

Wohlin, C., Aurum, A., 2015. Towards a decision-making structure for selecting a research design in empirical software engineering. Empir. Softw. Eng. 20, 1427–1455.

Wohlin, C., Höst, M., Henningsson, K., 2003. Empirical research methods in software engineering. In: Conradi, R., Wang, A. (Eds.), Empirical Methods and Studies in Software Engineering. Springer, pp. 7–23.

Zannier, C., Melnik, G., Maurer, F., 2006. On the success of empirical studies in the international conference on software engineering. In: 28th International Conference on Software Engineering. ICSE, pp. 341–350.

Matthias Galster is a professor at the University of Canterbury in Christchurch, New Zealand.

Danny Weyns is a professor at KU Leuven, Belgium and Linnaeus University, Sweden.