Evaluation Methods and Replicability of Software Architecture Research Objects

Marco Konersmann* Angelika Kaplan† Thomas Kühn† Robert Heinrich† Anne Koziolek† Ralf Reussner†
Jan Jürjens*‡ Mahmood al-Doori* Nicolas Boltz† Marco Ehl* Dominik Fuchß† Katharina Großer* Sebastian Hahner†
Jan Keim† Matthias Lohr* Timur Sağlam† Sophie Schulz† Jan-Philipp Töberg†§
*University of Koblenz-Landau, Germany, {lastname,mahmoodaldoori,mehl,matthiaslohr}@uni-koblenz.de
†Karlsruhe Institute of Technology, Germany, {firstname.lastname}@kit.edu, §uexdy@student.kit.edu
‡Fraunhofer Institute for Software and Systems Engineering, ISST, Germany

***Abstract*—Context: Software architecture (SA) as research area experienced an increase in empirical research, as identified by Galster and Weyns in 2016 [1]. Empirical research builds a sound foundation for the validity and comparability of the research. A current overview on the evaluation and replicability of SA research objects could help to discuss our empirical standards as a community. However, no such current overview exists.**

Objective: We aim at assessing the current state of practice of evaluating SA research objects and replication artifact provision in full technical conference papers from 2017 to 2021.

Method: We first create a categorization of papers regarding their evaluation and provision of replication artifacts. In a systematic literature review (SLR) with 153 papers we then investigate how SA research objects are evaluated and how artifacts are made available.

Results: We found that technical experiments (28%) and case studies (29%) are the most frequently used evaluation methods over all research objects. Functional suitability (46% of evaluated properties) and performance (29%) are the most evaluated properties. 17 papers (11%) provide replication packages and 97 papers (63%) explicitly state threats to validity. 17% of papers reference guidelines for evaluations and 14% of papers reference guidelines for threats to validity.

Conclusions: Our results indicate that the generalizability and repeatability of evaluations could be improved to enhance the maturity of the field; although, there are valid reasons for contributions to not publish their data. We derive from our findings a set of four proposals for improving the state of practice in evaluating software architecture research objects. Researchers can use our results to find recommendations on relevant properties to evaluate and evaluation methods to use and to identify reusable evaluation artifacts to compare their novel ideas with other research. Reviewers can use our results to compare the evaluation and replicability of submissions with the state of the practice.

***Index Terms*—software architecture research, meta-research, systematic literature review, evaluation**

I. Introduction

Researchers present many novel approaches and findings in the area of software architecture (SA) in conferences every year. As we are confronted as a community with this large amount of research, it is important to understand how the research is evaluated to understand their validity and how related approaches and their results compare to each other.

Galster and Weyns [1] identified an increase of empirical research in SA-related conference papers in 2016. Empirical research is a sound foundation for the validity and comparability of novel approaches and findings. In addition, the replicability of studies is considered important for trustworthy research. This is reflected in the International Conference on Software Architecture (ICSA) promoting replicability in their calls since 2020[1], introducing an *artifact evaluation track* since 2021[2], and introducing open science principles in the CfP in 2022[3]. That said, the SA community considers the importance to acknowledge contributions that cannot publish their data, e.g., to not publish business secrets. With a current overview on how SA research is evaluated in conference papers and how artifacts are made available, we could have a basis for discussing which standards we would like to set as a community. However, no such current overview exists. Thus, the *goals* of this paper are to:

G1: categorize SA research wrt. evaluation and replicability,
G2: create an overview of the current state of practice in evaluating SA research.

The *European Conference on Software Architecture* (ECSA) and the *International Conference on Software Architecture* (ICSA) series are the major venues to present and discuss novel ideas, approaches, and insights to SA. The papers presented there endure a rigorous peer review, including an assessment of their evaluation sections. In this paper, we systematically review the evaluation of research presented at both conference series. We focus on properties that are evaluated for research objects in full technical papers of these conferences and used evaluation methods. The research object of a paper is what the researchers investigate. Typical research objects in SA research include architecture analysis, design, optimization methods, or architecture decision making. Thus, our *contributions* are:

1) a classification schema for the validation of software architecture research evaluation (cf. goal G1),
2) a systematic literature review (SLR) [2] for an investigation of research objects, evaluation methods, evaluated properties, threats to validity, accessibility of replication

[1]https://icsa-conferences.org/2020/call-for-papers/technical-papers
[2]https://icsa-conferences.org/2021/call-for-papers/artifact-evaluation-track/
[3]https://icsa-conferences.org/2022/conference-tracks/author-instructions/

packages, and their interrelations in software architecture research papers (cf. goal G1, G2), and
3) a discussion of our findings (cf. goal G2) and proposals for improvements.

A characteristic of our SLR is that the reviewed papers are predefined to be ECSA and ICSA papers. We consider papers from 2017 to 2021 to limit the effort of our review to a reasonable size and to create an overview of current practices (goal G2). Our work may support the following stakeholders:

- *Researchers:* Researchers can use our results to get an overview and recommendations about the practice of evaluating software architecture research, e.g., which properties might be relevant to evaluate specific research objects and which method is commonly used to evaluate that. Researchers can also identify reusable case studies based on their research object and property to evaluate.
- *Reviewers:* Reviewers can use our results to compare and assess the evaluation and replicability of submissions based on the state of practice.

Research Questions (RQ)s: We investigate three main research questions. Figure 1 gives an overview of the research questions in the data schema.

RQ 1: What is the distribution of research objects and their evaluation and how did their proportions change over time? With this RQ, we want to provide a basis for categorizing the contributions wrt. their evaluation and replicability (goal G1) and to create an overview of the state of practice (goal G2). We break down RQ 1 into the following sub-questions:

RQ 1.1: What is the proportion of research objects in the body of literature per year?
RQ 1.2: What is the proportion of evaluation methods in the body of literature per year?
RQ 1.3: What is the proportion of papers for which artifacts were provided for replication per year?

RQ 2: How are specific research objects evaluated and how accessible are their evaluation artifacts? With this RQ, we want to identify relationships between research objects and different aspects of evaluations. This helps categorizing the contributions wrt. their evaluation (G1) and creates an overview of these relations (G2) to identify how researchers evaluate specific research objectives and how they report it. We break down RQ 2 into the following sub-questions:

RQ 2.1: What is the relationship between the evaluation method and the research object?
RQ 2.2: What is the relationship between the research object and the evaluated property?
RQ 2.3: What is the relationship between the evaluation method and the evaluated property?
RQ 2.4: What is the relationship between the evaluation method and the threats to validity?
RQ 2.5: What is the relationship between the evaluation method and the provision of replication artifacts?

RQ 3: Which guidelines are used for evaluation? With this RQ, we want to identify which guidelines are stated to be used for evaluations and for discussions of threats to validity. We break down RQ 3 into the following sub-questions:

RQ 3.1: Which guidelines are referenced for applying evaluation methods?
RQ 3.2: Which guidelines are consulted when discussing threats to validity?

We provide a **replication package** [3] with the tabulated and visualized review data, a BibTeX file with all papers considered, scripts for summarizing and visualizing, and a copy of the wiki available to the reviewers with descriptions of all characteristics in the data extraction form. All investigated papers are also listed online[4] for collaboration with raw data.

Henceforth, we describe our research method (Section II) and then present an aggregation of our results (Section III). In Section IV, we discuss our findings and propose improvements. Threats to validity and related work of our study are presented in Section V and Section VI, respectively. Section VII concludes the paper.

II. Research Method

We conducted a systematic literature review (SLR) following the guideline of Kitchenham and Charters [2].

1) Data Sources and Search Strategy: We consider the ECSA and ICSA conference series as the major venues for SA research. Therefore, we use manual search to examine papers of these conferences. We retrieved the papers via DBLP[5].

2) Study Selection: The inclusion criteria were:

IC1: ECSA and ICSA papers of the years 2017 to 2021 to investigate the last 5 years.
IC2: Paper classes *evaluation research*, *validation research*, *proposal of solution*, and *philosophical paper* of the classification of Wieringa et al. [4].
IC3: Primary research, as secondary studies (i.e., literature studies) are not in focus.
IC4: Papers included in the main proceedings of the respective conferences.

The paper classes *personal experience papers* and *opinion papers* were not included in IC2, as they do not require an evaluation. While experience papers may be built upon experiences with case studies, these then do not serve as evaluation of a research object. Multiple classes can be assigned to a paper. Matching one included class sufficed for inclusion.

The exclusion criteria were:

EC1: Teaching papers, as we focus on research objects related to architecting directly, i.e., activities such as making architecture decisions for a given system, documenting them, and so on.
EC2: Short papers, because they do not have enough space for a detailed description of an evaluation and aim at presenting work in progress or new ideas. Consequently, these papers have lower requirements regarding their evaluation. We identified short papers by the number of pages given in the calls for paper.

All included papers were accessible and written in English.

[4]https://gitlab.com/SoftwareArchitectureResearch/StateOfPractice/-/wikis/Results

[5]https://dblp.uni-trier.de/db/conf/ecsa/index.html and https://dblp.uni-trier.de/db/conf/icsa/index.html

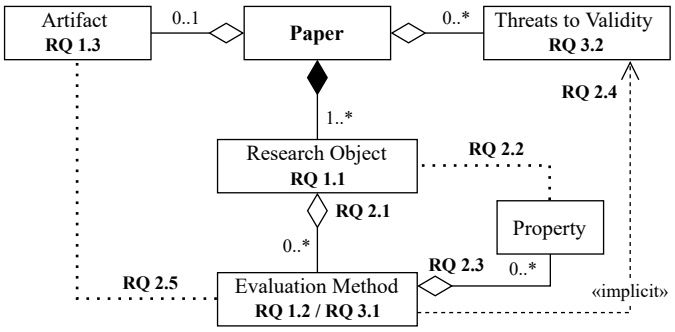


Fig. 1: Data schema of the data extraction and corresponding research questions. Dashed arrows indicate inferred relations.

3) Study Quality Assessment: We did not apply further quality assessment. We survey the current state of practice at the ECSA and ICSA series. Excluding papers via a quality assessment would bias the perception of the current state and would be a threat to validity (cf. Section V).

4) Data Extraction Form and Process: Derived from our research goals and the corresponding RQs, we defined a data extraction form (cf. Table I). Figure 1 shows the interrelations of the extracted data. A paper presents at least one research object. Research objects should be evaluated using an evaluation method for properties. We experienced that threats to validity are usually reported not for specific evaluation methods, but for a paper as a whole. Similarly, tools, input data, and replication artifacts are assigned to a paper, not to specific research objects. Table I gives details of the data extraction form. The complete data extraction form with the definition of its contents can be seen in the copy of the wiki in our replication package [3]. This also includes a description of each data item, e.g., each research object.

For each paper two researchers extracted the data by reviewing the given paper independently. The reviewers discussed pairwise differences, if any occurred. Where no agreement could be created, a third reviewer helped to solve conflicts. The allocation of paper to reviewer was randomized. The distribution was not equal, due to individual time constraints of the participating researchers. We used the SLR Toolkit [5] as tool support to classify papers with our data extraction form. In the following, we describe every data extraction item of the form (cf. Table I).

a) Meta Data: Identifying paper classes required to understand the papers content. This information was therefore extracted manually during the review.

b) Research Object: The research object of a paper is what the researchers investigate. Authors usually explicitly state the research object (e.g., "We present a design method…"). The set of research objects must not be too fine grained, nor too coarse grained to get meaningful results. We extracted a set of 13 research objects in a pilot study using a sample of papers from the ECSA conference that proved to be stable in the main study for ECSA and ICSA. See Figure 2a for the list of research objects.

TABLE I
DATA EXTRACTION FORM

Data Item	Description
	Meta Data
Paper ID	BibTex-Key as unique identifier
Paper Class	Paper class wrt. Wieringa et al. [4]
	Content Data
Research Object	Investigated object(s) of research
Evaluation Method (EM)	Applied evaluation method wrt. ACM Empirical Standards [6]
EM Guidelines	DOI/ISBN of referenced guideline
Property	Property evaluated with a given evaluation method for a given research object
Tool Support	{available, used (but not available), none}
Input Data	{available, used (but not available), none}
Replication Package	{yes, no} for availability
Threats to Validity (TtV)	Categories of threats to validity wrt. [7]
TtV Guidelines	DOI/ISBN of referenced guideline

As our research goals aim at the evaluation of architecture research, and not their paper reports, we investigate how specific *research objects* are evaluated, where a paper may present multiple research objects.

c) Evaluation Method: The evaluation method is the method chosen to evaluate a research object. Evaluation methods are usually explicitly stated by authors (e.g., "We conducted a case study…"). However, these statements often mean different things. For example, reported case studies are sometimes larger example systems and sometimes real cases. We use the ACM Empirical Standards [6] for comparison. Therefore, we map the evaluation methods used in the reviewed papers to the corresponding method description in the ACM Empirical Standards. See Figure 2b for the list of detected evaluation methods. In some cases, methods can be used to construct or to validate a theory. For example, Haselböck et al. [8] use a literature review and an interview to create decision models for microservices. These are not considered evaluation methods in this case. We also collect any cited guidelines for evaluation as stated by the authors.

d) Property: Researchers evaluate specific properties of research objects. The properties are often explicitly stated (e.g., "We evaluated the feasibility of our approach…"). Multiple properties can be evaluated with each evaluation method. In this review, we use a common set of potential properties for comparison: the properties defined by ISO 25010 [9] for *quality of use* and *product qualities* alongside with *quality criteria for analytical methods* introduced by Taverniers et al. [10]. See Figure 4a for the list of detected properties.

e) Tool Support and Input Data: During the evaluation of architecture research, tools or input data are often used. As tool support, we regard tools that are not exchangeable for the evaluation, e.g., tool prototypes for an approach or a specific form for a questionnaire. Input data is the data used as input to the validation, e.g., models, SA descriptions or transcribed answers to questionnaires. We identify whether they are mentioned in the given paper (categorizing it as *used*) and whether it is (still) publicly *available*. Tool support and input data are elicited only on the level of papers, not research objects, because the relation is not always clearly described.

TABLE II
REFERENCED EVALUATION GUIDELINES
(RECURRING ONLY)

Refs.	1st Author	Title of Evaluation Guideline
[11], [12]	Runeson	Guidelines for Conducting and Reporting Case Study Research in Software Engineering Case Study Research in Software Engineering: Guidelines and Examples
[13]	Glaser	Discovery of Grounded Theory: Strategies for Qualitative Research

f) Replication Package: We check whether a replication package is publicly available.

g) Threats to Validity: We collect the authors' statements about threats to validity of their evaluation and the corresponding guidelines, if any were referenced. For comparison, we grouped threats to validity based on Feldt and Magazinius [7]: *Internal Validity*, *External Validity*, *Construct Validity*, *Confirmability*, and *Repeatability* (i.e., dependability in [7]).

5) Data Synthesis: For data synthesis, we collate and summarize the results of the included papers in a quantitative way to present the proportions for RQ 1, the relationship between the extracted meta and content data for RQ 2 (see Section III), and the most-referenced guidelines for RQ 3 (see Table II and Table III). For information visualization, we followed the supplement material of the ACM SIGSOFT Empirical Standards for systematic literature reviews [6].

III. RESULTS

Following our research method, we classified 153 full technical papers. A list of papers with the extracted raw data is in our replication package [3]. We focus on showcasing those results necessary to answer our research questions.

For each paper, we extracted the research object under investigation, identifying a total of 170 research objects investigated, whereas 38 papers investigated two distinct research objects. We identified for each research object of a paper, which evaluation methods are employed and which properties of the research object are evaluated. Figure 2a shows the proportion of research objects over time (RQ 1.1). Analogously, Figure 2b shows the proportion of evaluation methods over time (RQ 1.2). For the latter, we found that most research objects were evaluated with a single method. 16 times research objects were evaluated with two methods in a paper and two times with three. Table II lists the recurring referenced evaluation guidelines (see [3] for the full list).

To elucidate how artifacts are provided for replication (RQ 1.3), we show the proportion of papers each year (see Figure 3) that either provide a replication package (*packaged*), made the employed tool or input data available (*available*), used unavailable tools and input data (*used*), or did not employ a tool or input data (*none*).

For a deeper analysis of the evaluation in the investigated body of literature, we illustrate the relations between (RQ 2.1) *research objects* and *evaluation methods* and, (RQ 2.2) *research objects* and their evaluated *properties* as bubble charts in Figure 4b and Figure 4a, respectively. We further depict the relationships between the *evaluation method* (per *research object*) and the evaluated *property* (RQ 2.3, see Figure 5a), the discussed *threats to validity* (RQ 2.4, see Figure 5b), and the provision of *replication artifacts* (RQ 2.5, see Figure 5c).

TABLE III
REFERENCED THREAT TO VALIDITY GUIDELINES
(RECURRING ONLY)

Refs.	1st Author	Title of Threats to Validity Guideline
[11], [12]	Runeson	Guidelines for Conducting and Reporting Case Study Research in Software Engineering Case Study Research in Software Engineering: Guidelines and Examples
[14], [15]	Wohlin	Experimentation in Software Engineering
[16]	Gasson	Rigor in Grounded Theory Research: An Interpretive Perspective on Generating Theory from Qualitative Field Studies
[17]	Cook	Quasi-Experimentation: Design & Analysis Issues for Field Settings

While properties and evaluation methods were assigned to the corresponding research object(s) of a paper, please note, that the availability of artifacts and the discussion of threats to validity were determined for each paper (see Figure 1). Consequently, the number of instances of evaluation methods exceeds the number of papers and the number of papers providing replication packages per evaluation method is skewed towards the number of evaluation methods per paper. Nevertheless, these relations present current research efforts in the SA community as represented in ECSA and ICSA papers and indicate how different research objects are evaluated.

Regarding the declaration of threats to validity (RQ 2.4), we found that 97 of 153 of the papers did not declare threats to validity. 79 of those addressed multiple threats, whereas *External Validity* (83), *Internal Validity* (64), and *Construct Validity* (45) were the most common ones. 26 papers (17%) referenced evaluation guidelines that they applied (RQ 3.1) and 22 papers explicitly reference guidelines for threats to validity (RQ 3.2). Table III shows the recurring referenced guidelines (see [3] for the full list).

IV. FINDINGS AND DISCUSSION

RQ 1: What is the distribution of research objects and their evaluation and how did their proportions change over time?

For each paper, we related the research objects, the research methods, and the provision of artifacts to the publication year for creating an overview and identifying trends and gaps.

RQ 1.1: What is the proportion of research objects in the body of literature per year? The proportion of research objects (see Figure 2a) shows *Architecture Analysis Method*s (31) and *Architecture Design Method*s (26) as major research objects. Behind those, *Architecture Decision Making* (18), *Architecture Optimization Method*s (17), and *Reference Architecture*s (16) are also presented often. Still, many other research objects come up each year, showing that the SA research presented at ECSA and ICSA is diverse.

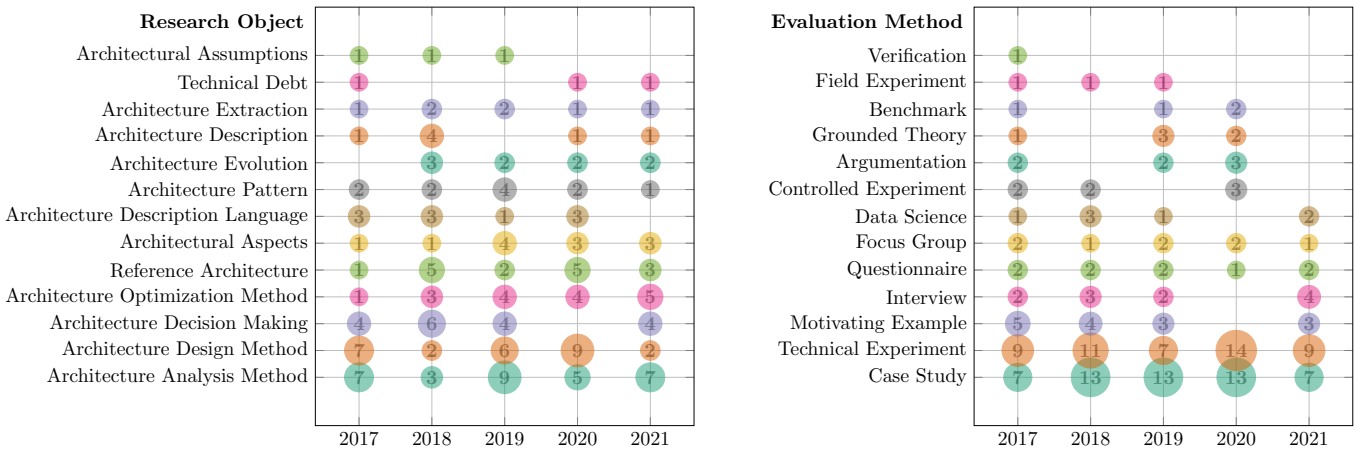


Fig. 2: Number of research objects (a) and evaluation methods (per research object) (b) from 2017 to 2021.

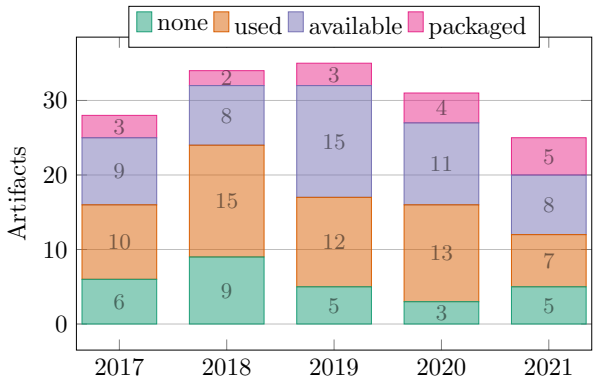


Fig. 3: Number of publications featuring a replication package (packaged), providing links to available tools or input data (available), employed tools or input data (used), or none of the above (none) from 2017 to 2021.

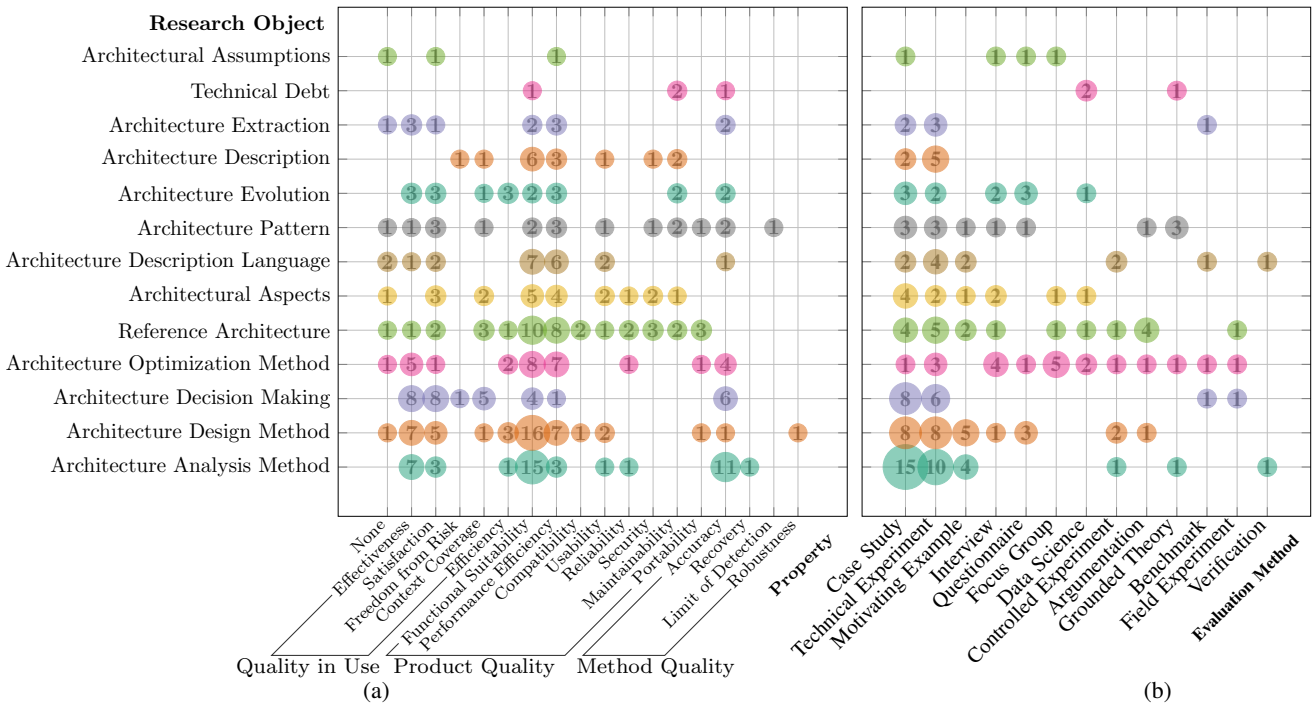
RQ 1.2: What is the proportion of evaluation methods in the body of literature per year? *Case studies* (53) and *Technical Experiments* (50) are the dominating evaluation methods (see Figure 2b) together making up 57% of the employed evaluation methods. Although we cannot conclude whether an evaluation method is better then the other, as each has different characteristics, it is apparent that these dominant methods focus on one specific phenomenon to investigate. *Technical Experiments* are procedures carried out to support or refute a hypothesis regarding a technological artifact. The ACM SIGSOFT Empirical Standards cite Yin [18] to describe *Case Studies*: "An empirical inquiry that investigates a contemporary phenomenon (the "case") in depth and within its real-world context, especially when the boundaries between phenomenon and context [are unclear]". As case studies require a real-world context, they can generate evidence for the tested properties of the research object. However, case studies are not well-suited for generalization, as they deeply investigate one phenomenon. We can observe that many of the applied evaluation methods do not aim at validating the generalizability.

We found that 17 papers use a *Motivating Example* and/or *Argumentation* as evaluation method, meaning that they, e.g., describe a problem with a small motivating example, and then show and argue how their approach behaves according to the given example. *Benchmarks* have been used for four individual research objects (see Figure 4b) and for seven evaluated properties (see Figure 5a): *Performance Efficiency* (4), *Accuracy* (2), and *Effectiveness* (1). Overall, we found a large variety of methods used for evaluation and we did not identify a systematic approach on using specific evaluation methods for specific properties.

RQ 1.3: What is the proportion of papers for which artifacts were provided for replication per year? The distribution of evaluation artifact provision shows a trend towards more artifact availability: 2019 and 2021 over 50% of papers made artifacts available or packaged them (cf. Figure 3). A reason for this might be ICSA promoting replicability in the CfPs since 2020 and prior discussions in the community. It should be noted that it is not feasible for all papers to make artifacts available: (e.g., industry participation often requires to not provide details about data or systems).

11% of the papers provide a dedicated, available replication package. 33% of the papers provision employed tools or input data for evaluation, such that they were still available. For 37% of the papers neither tools nor input data employed is made available. Besides that, 18% of the papers did neither feature tools or input data for their evaluation.

Summary RQ 1: SA research at ECSA and ICSA is quite diverse wrt. research objects (inside the research area of SA), with a focus on architecture analysis methods and architecture design methods making up one third of the research objects found. The dominating evaluation methods are case studies and technical experiments, whereas most evaluation methods are used to measure exactly one quality (58% of evaluation methods). The latter is in line with the trend shown in Galster and Weyns [1]. ICSA started calling for artifacts only since 2020. We see that neither research objects nor evaluation methods heavily changed in the past five years, with a trend to more artifact provisioning since 2019. Future work might consider more than 5 years to identify long-term trends.



(a) (b)

Fig. 4: Relations between research objects of publications, properties investigated for evaluation (a) and evaluation methods applied (b), whereas multiple evaluation methods and properties could be considered for one research object.

RQ 2: How are specific research objects evaluated and how accessible are their evaluation artifacts?

We analyze and discuss the relationships between the extracted data items in terms of SA research objects, evaluated properties, evaluation methods, threats to validity, and replication artifacts to create an overview of the current relations.

RQ 2.1: What is the relationship between the evaluation method and the research object? The choice of evaluation methods should be based on the research object and the property to evaluate. For some research objects, like the aforementioned *Analysis Method*s, the number of different evaluation methods is quite small (cf. Figure 4b), while we observe that *Architecture Optimization Method*s and *Reference Architecture*s are evaluated with a multitude of different evaluation methods.

The dominant evaluation methods *Case Study* and *Technical Experiment* are also the most-used for *Architecture Analysis Method*s, *Architecture Design Method*s, *Architecture Optimization Method*s, and *Reference Architecture*s. *Benchmark*s, as a comparative method, are employed for 4 research objects.

RQ 2.2: What is the relationship between the research object and the evaluated property? The most-evaluated property over all research objects is the *Functional Suitability*. This is not surprising, because when presenting a solution for a problem, the first question to ask is often ”Does it work?”, before more elaborate properties are evaluated. The second-most evaluated property is the *Performance Efficiency*, which is also not surprising, considering that SA approaches can be relatively demanding regarding computational resources, when they, e.g., employ pattern matching or model transformations on large graphs like source code. Furthermore, for *Architecture Analysis Method*s and *Architecture Optimization Method*s the *Accuracy* and the *Effectiveness* is often evaluated.

For most research objects, many properties are evaluated (esp. *Reference Architecture*s). Research objects with less evaluated properties are also research objects that have been reported less often. We identified 9 research objects in papers, that do not have any property evaluated, meaning that there is, e.g., a proposal of solution or a conceptual framework presented, but not evaluated.

RQ 2.3: What is the relationship between the evaluation method and the evaluated property? There are no clear correspondences between evaluated properties and methods in the data (see Figure 5a). *Case studies* and *Technical Experiment*s are being used to evaluate a wide range of properties. Properties of the category *Quality in Use* are mainly evaluated using the dominant evaluation methods, but also with evaluation methods that involve humans by definition, like *Questionnaire*s, *Interview*s, and controlled *Experiment*s. Properties of the category *Product Quality* are mainly evaluated using *Technical Experiment*s and *Case Studies*. Additionally, *Motivating Example*s and *Argumentation* are used in this category. In the category of analytical *Method Quality*, *Accuracy* is the most prominent quality considered.

Similar to the relation between evaluation method and properties, we observe that many properties are evaluated with different evaluation methods. We see a minor (and natural) tendency to evaluations that involve human subjects for properties of the category *Quality in Use*. Besides that, we do not see a common agreement on which evaluation methods seem to be suitable for specific properties.

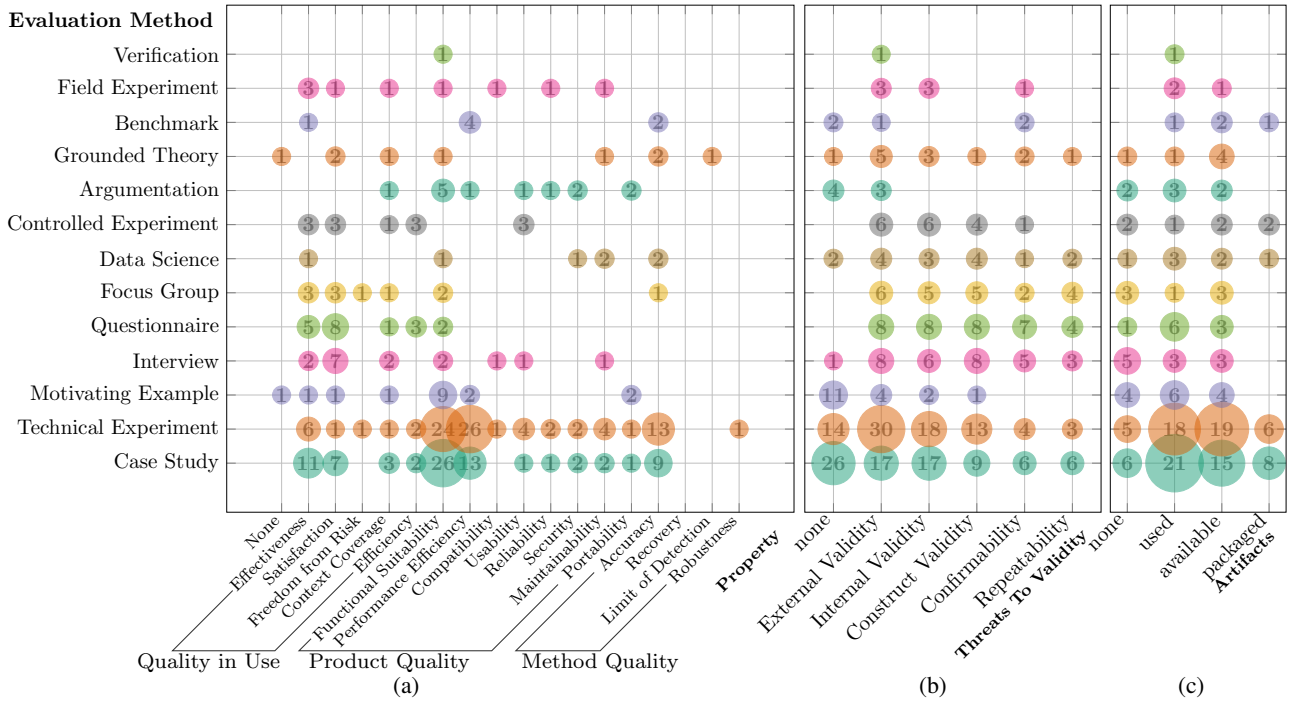


Fig. 5: Relations between evaluation methods (per research object) and evaluated property (a), considered threats to validity (per paper) (b) as well as kind of provided artifact (per paper) (c).

RQ 2.4: What is the relationship between the evaluation method and the threats to validity? Of all 153 papers, 97 (63%) discuss threats to validity. *External* and *Internal Validity* are the most-discussed threats to validity in the papers we investigated. 54% of the papers discuss *External Validity*, i.e., the generalizability of the findings. This is interesting wrt. the observation that the dominating evaluation methods investigate a single phenomenon. The generalizability does not seem to be in focus of the evaluation activities, for it is often neither evaluated nor discussed. 17 papers (34% of the 50 papers that apply case studies) discuss the external validity when applying a *Case Study*. In contrast, for *Technical Experiment*s, 32 papers (66% of 48 papers that employ technical experiments) discuss threats to *External Validity*.

7% of the papers at hand discuss *Repeatability*. 6 of the 50 papers (12%) that apply case studies and 3 of the 48 papers (6%) that apply technical experiments discuss the repeatability. Besides case studies, papers with questionnaires (4 of 10 papers) and focus groups (3 of 7 papers) usually discuss their repeatability. *Confirmability* is discussed in 8% of papers.

We experienced that the reporting of threats to validity is quite diverse and no systematic approach seems to be followed by the community to report about threats to validity.

RQ 2.5: What is the relationship between the evaluation method and the provision of replication artifacts? We observe that most *replication artifacts* are made available for *Case studies* and *Technical Experiment*s (cf. Figure 5c), which correlates to the most-employed evaluation methods. Besides that, no focus on specific evaluation methods was found.

Of all 50 papers with case studies, 14 papers (28%) make their tool support and 17 papers (34%) their input data *available*. 8 papers (16%) *package* them in replication packages. However, it is possible that non-disclosure agreements restrict the publication of replication packages. For example, case studies may contain business secrets, that cannot be published. While this hinders the replicability, these papers still provide important insights into SA topics in practice.

Table IV shows those papers with available case studies, related to their research objects and evaluated properties. Genfer and Zdun [19] use their case study for two research objects, therefore this paper occurs in the table twice.

Summary RQ 2: The most prominent way of evaluating SA research is to evaluate the functional suitability and performance using technical experiments and case studies. The human-centered practice *architecture decision making* is mostly evaluated with human-centered evaluation methods interview and focus group. Few comparative evaluation methods, like benchmarks, are used. Overall, we see no clear agreement on which properties should be evaluated for specific research objects or which methods to use for specific properties.

RQ 3: Which guidelines are used for evaluation?

17% of papers reference evaluation guidelines **(RQ 3.1)**. The most-referenced guidelines are two versions from Runeson et al. [11], [12], which we aggregate in Table II. These publications provide guidelines for how to conduct and report case studies. 14% of papers reference guidelines for threats to validity **(RQ 3.2)**. The most-referenced guidelines are the same from Runeson et al. ([11], [12]). Here, guidelines for threats to validity are included in the guidelines for the evaluation method. Overall, we can observe that guidelines are not referenced systematically in the investigated papers.

TABLE IV
PAPERS MAKING INPUT DATA OF CASE STUDIES AVAILABLE

Research Object	Property	Paper ID
Architecture Analysis Method	Accuracy	[20], [21], [22]
	Effectiveness	[23], [19], [22]
	Functional Suitability	[24], [25], [22]
	Performance Efficiency	[26]
	Satisfaction	[24], [25]
Architecture Description Language	Effectiveness	[27]
	Functional Suitability	[27]
	Performance Efficiency	[27]
	Satisfaction	[27]
Architectural Aspects	Functional Suitability	[28]
	None	[19]
Architecture Optimization Method	Functional Suitability	[28]
	Satisfaction	[29]
Architectural Assumptions	Performance Efficiency	[30]
Architecture Design Method	Accuracy	[31]
Architecture Extraction	Effectiveness	[23]
Architecture Pattern	Security	[32]
Reference Architecture	Functional Suitability	[33]

A. Proposals

We observe that diverse research objects are evaluated using diverse evaluation methods. To improve the evaluation quality even further, we derive four proposals based on our discussion:

We identified few papers that discuss the threats to validity wrt. repeatability and confirmability. This is in-line with the observation, that replication artifacts are not provided broadly (see RQ 1.3), which makes it more difficult for other researchers to replicate a study. These observations indicate that the generalizability is not well-studied. We propose **(P1)** to foster the generalizability of evaluation results, e.g., by calling for replication of existing studies and for the application of existing approaches in different contexts. The ICSA conference series introduced an artifact evaluation track in 2021, which motivates the reproduction and replication of studies.

We also propose **(P2)** to develop benchmarks to compare approaches. Benchmarks are rarely employed as evaluation method, although it has been observed that "the creation and widespread use of a benchmark within a research area is frequently accompanied by rapid technical progress and community building" [34]. Comparing approaches with benchmarks can be seen as a maturity indicator (see [35]).

Most notably there are no benchmarks applied for architecture analysis methods, although analysis methods for the same analysis goals could very well be compared against a common benchmark. Our impression from the study is that the reason for this lack of comparison lies not only in the different analysis goals (e.g., security, performance, etc.), but also in the different architecture representations on which these analyses operate. The same applies to architecture optimization methods, which are well-suited comparing their optimization results in benchmarks. This confirms earlier observations by Aleti et al. [36]. However, only one of the investigated papers employed a benchmark for optimizations: Cardoso et al. [37] use a Hadoop benchmark that produces read and write operations on the distributed Hadoop file system HDFS to evaluate their improvement on dynamic architecture reconfiguration.

Besides the papers that employ benchmarks, the input data made available for case studies (see Table IV) should be investigated for their suitability for benchmarks by the respective researchers working with these research objects.

We propose **(P3)** that the SA research community fosters the provision of replication packages, e.g., by highlighting papers with replication packages by using ACM artifact badges [38]. While the provision of tools and input data is very helpful for readers, replication packages, e.g., additionally fix versions of the artifacts or permit easy replication of the results and charts shown in the paper. Ferro and Kelly [39] report on a survey with positive feedback and responses of having replication badges introduced. As a community, we should motivate to increase the confirmability and the repeatability, which includes the provision of replication artifacts where possible. Alternatively, the community could introduce a replication track that explicitly calls for replication studies.

Recently, the community increased the availability of replication artifacts (see RQ 1.3). We think that following this trend would be valuable for the community, to have more papers that provide more empirical evidence and make their artifacts available. Wrt. our observations in this review and the goal to have a better replicability for a higher percentage of papers, we see a benefit of ICSA introducing open science principles and an artifact evaluation track with badges for reproducible artifacts in their CfP. It should be noted that not all papers must follow open science principles or present empirical research to not repel qualitative work and industry participation.

We propose **(P4)** that the community builds guidelines on which properties are important to be evaluated for which research objects, which evaluation methods are most-suitable for them, and which threats to validity should be discussed for these methods. This can serve as a basis for researchers to identify relevant properties, but also for reviewers to identify whether the relevant properties are addressed.

Besides the clusters "Does it work?" and "Is it fast?" we see many different properties evaluated, but we do not observe that a systematic approach is followed. There are multiple potential interpretations for this observation. The community might view the evaluation whether "it works" and whether "it is sufficiently fast" as a good point to report about a research object. Either a thorough evaluation of further properties seems unnecessary or is reported at other venues. Another interpretation is that there is a lack of common agreement of which properties *should be* evaluated for a research object. There seems to be no common agreement on what threats to validity to consider for specific evaluation methods and how to present them, although multiple guidelines for threats to validity exist (for an overview see Peterson et al. [40]).

Especially early career researchers might benefit from having a guidance on which properties are important to evaluate for their research object, which methods can be used for that, and which threats to validity should be considered and

reported. It remains to be discussed whether general guidelines from software engineering should be adapted for SA research or concrete SA guidelines should be developed. The authors propose to use the Empirical Standards [6] as a basis for such a guideline. However, the community should accept when diversion of these guidelines are argued, to stay open for novel types of research, that cannot comply with such guidelines due to their novelty.

V. Threats to Validity

We discuss threats to validity of our review and potential biases based on the guideline of Kitchenham and Charters [2] in the following categories:

1) Selection and Publication Bias: An incomplete data set of included publications is one of the main threats to validity. In our case, we systematically used manual search (i.e., no random selection) of the ECSA and ICSA conference series within a time frame of 5 years (2017–2021). The results and insights are strictly limited to this body of literature. We can overcome this limitation by expanding the time span and by taking further SA research venues into account to verify and consolidate our results or by using a different search strategy (e.g., database search) in future work. We consider ECSA and ICSA as the major venues for SA research. Further venues include more general conferences (e.g., the International Conference on Software Engineering ICSE) or related journals (e.g., the Journal of Software and Systems JSS), but also co-located workshops at ECSA and ICSA.

2) Measurement and Exclusion Bias: To avoid this bias, we used inclusion and exclusion criteria (i.e., full papers in the SA research field are chosen according to our research goals). Additionally, we conducted a briefing in advance with all reviewers to inform about the review process and technical support. Moreover, we manually assigned every ECSA and ICSA paper to two reviewers for further data extraction. The classification proposed in terms of our data extraction items was predefined using a pilot study. Unclarities and ambiguities during the data extraction were resolved in a discussion with the first three authors of this paper.

3) Performance Bias: We documented our method design in a written protocol and used a wiki to document all artifacts of our conducted review process (e.g., inclusion and exclusion criteria, data extraction form including definition of all classes, review process). Additionally, we gave an illustrative example for the data extraction in our briefing, clarified open questions, and gave an introduction into the tool-support. The process conducted in this SLR has some minor deviations from the recommendations of the guideline of Kitchenham and Charters from 2007 [2]: As stated in Section II, we did not apply quality assessment, because excluding more studies would be a threat to validity. Especially, we did not want to further assess the papers regarding their empirical method design but instead analyzed the state of practice regarding evaluation and replicability. Although Kitchenham does not prescribe to measure the inter-rater agreement, it helps to understand the clarity of the data extraction form. We did not measure the inter-rater agreement, because our review process controlled it: the two reviewers for a paper discussed pairwise differences in their review to come to an agreement. Where no agreement could be achieved between the two, a third reviewer was asked for help. After these discussions all reviewers agreed upon a shared review.

VI. Related Work

In the following, we identify and present the following main areas that are related to our work. Namely, literature studies on SA research, literature studies on evaluation methods as well as validity and replication issues in software engineering (SE) research. All of theses papers use the same research method (i.e., secondary studies). Furthermore, we regard guidelines for conducting and evaluating research methods as another related area. Finally, we also present works that use other research methods (e.g., online questionnaires or experience report), but also focus on meta-research (i.e., research on research) in SA.

a) Secondary Studies on Software Architecture Research: We regard literature studies on SA research as related that do not focus on a single sub-discipline of SA research.

Qureshi et al. [41] performed a literature review to synthesize empirical work (i.e., evidence) in SA and report trends, patterns and knowledge gaps in this area. They first analyze SA areas (e.g., SQ design, SA documentation/description, etc.) and identify trends in this point. In addition, they investigate the strength of empirical evidence (e.g., action research) in terms of source of evidence and methods used. They did not analyze any interdependencies or interrelations in their work.

Galster and Weyns [1] aim to assess the state of practice of empirical SA research in a literature study. They argue how important empirical research in SA is to gain insights about phenomena, to increase confidence into a new solution approach, and to create evidence for the validity of gained research results. Therefore, they conduct an analysis of which empirical methods are most prominent, what the role of human participants in SA papers is, and which validity concerns are addressed. In contrast, we also include non-empirical methods as evaluation method and, moreover, focused on replication issues associated with empirical work to investigate trustworthiness beyond validity threats.

b) Secondary Studies on Evaluation Methods, Validity and Replication Issues in Software Engineering Research: We regard literature studies as related, that address evaluation methods, corresponding threats to validity and replication issues in SE (i.e., papers of evaluation types not specific to SA research and that do not investigate evaluation methods according to the corresponding research object or its property).

First, the proposed *validation types* (i.e., analysis, evaluation, experience, example, and no validation or blatant assertion) of Shaw [42] should categorize strategies to investigate a research problem in SE. A literature study of ICSE papers form 2002 is conducted to validate these categories of all submitted (i.e., including rejected papers) and only accepted papers. Follow-up studies by Theisen et al. [43] (literature review of ICSE 2016 submitted and accepted papers) and

Bertolino et al. [44] (literature review of ICSE, ESEC/FSE and ASE accepted papers from 2012-2016) validated and consolidated the categorization scheme in this point based on the body of considered literature. Thus, they provide, in contrast to our review, specific validation types where concrete evaluation methods (e.g., case study, grounded theory, etc.) are clustered and not named directly, but described due to their characteristics listed above.

Second, Glass et al. [45] examine *research methods* (e.g., action research, conceptual analysis, data analysis) in six leading research journals in the SE research field. They regard a more detailed level of research technique like we do in our study. The paper concludes that methods and approaches in SE research are narrow.

Third, Zelkowitz and Wallace [46] also proposed similar validation categories (i.e., twelve-model classification scheme for performing experimentation within the software development domain) to examine the extent to which SE papers validate the claims made in those papers. Theses are specific to methods for experimental or empirical studies. In contrast to their study, we also regard non-empirical evaluation methods for SA research objects.

Regarding *Threats to Validity* Feldt and Magazinius [7] outline that for the quality of a research study the consideration of threats to validity regarding the results and the study itself is very important. Based on existing advice and guidelines, they provide an overview of possible threats in empirical SE research. With the performance of a literature review of 43 papers published in the ESEM conference, they analyzed validity issues. Based on this analysis, they present, among others, recommendations on how to better support validity analysis in the future. Furthermore, Wright et al. [47] discuss challenges to internal, external, and construct validity in SE research and investigate via literature study (i.e., papers of ICSE and FSE conference using real world software systems) whether issues of external validity are properly addressed. Regarding *replication issues* we see the mapping study of Cruz et al. [48] as related, as the authors conducted a literature study by including papers published form 2013 to 2018 that reported at least one replication of an empirical study in SE.

c) Guidelines for Venues and Research Methods: As this paper aims to support researchers on how to conduct and assess SA research validation, we consider guidelines of conferences or journals (e.g., [49], [50]) and of SE paper quality initiatives (cf. The ACM SIGSOFT Empirical Standards [6]) as another related area to our work.

d) Meta-Research in Software Architecture: Galster et al. [51] discuss issues in empirical SA research, such as poorly designed and communicated studies, lack of replications, and little practical relevance. They consulted 455 programme committee members of all editions of major SA conferences (i.e., CBSE, ECSA, ICSA, QoSA, and WICSA) in a questionnaire to get a deeper understanding of perceptions of SA researchers. They look at SA as a research sub-field of SE, noting that these issues can also be observed in the larger context of software engineering research.

Also Falessi et al. [52] (previous work in [53]) map the empirical paradigm with respect to its applicability to SA research. They state that the major efforts in SA research by developing new methods, techniques, and tools should be aligned with a rigorous empirical assessment. In an experience report, they describe thirteen challenges and ten lessons learned that they derived from the use of, e.g., controlled and replicated experiments. In conclusion, they support and suggest a greater synergy between the empirical SE and SA communities.

VII. Conclusion

In this paper, we categorized software architecture research wrt. their evaluation and replicability (goal G1) and created an overview of the current state of practice in evaluating software architecture research (goal G2). Therefore, we created a classification schema for the validation of software architecture research evaluations, conducted a systematic literature review of full technical papers published at ECSA and ICSA from 2017 to 2021. We analyzed 153 full papers, discussed our findings, and presented proposals for improvement. Our research questions address (1) the distribution of research objects and how they are evaluated, the provision of replication artifacts, and trends over time, (2) the relations between research objects, evaluation methods, evaluated properties, replication artifacts, and (3) guidelines applied for discussing evaluations and threats to validity.

We observe that architecture analysis and design methods are the most reported research objects and diverse research objects are reported about in the investigated papers. Technical experiments and case studies as the dominant evaluation methods focus on the validation over a single phenomenon.

The major evaluation methods applied in the body of literature do not focus on generalizability and there is a lack of discussion about the external validity of studies. There are also few comparative studies, such as benchmarks. Most papers (89%) do not provide replication packages and only about 33% make input data or tools available that were used in the evaluation. The provision of artifacts increased since 2019, with ICSA introducing replicability and open science aspects in their CfPs since 2020. There are, however, evaluation artifacts available in the investigated papers, that can be used by the community, e.g., to build benchmarks for different research objects and properties to check. We listed in Table IV potential candidates from the list of papers with case studies that make their input data available. In our discussion, we present four proposals to improve the evaluation (see Section IV-A).

Researchers can use our results, e.g., to get an overview about the practice of evaluating software architecture research objects and to find examples for evaluations of comparable research. Reviewers can use our results to compare the evaluation and replicability of submissions with the state of the practice. In future work, we will investigate the development of guidelines about which properties should be evaluated for specific research objects and which evaluation methods are feasible for them.

References

[1] M. Galster and D. Weyns, "Empirical research in software architecture: How far have we come?" in *13th Working IEEE/IFIP Conference on Software Architecture, WICSA 2016, Venice, Italy, April 5-8, 2016*. IEEE Computer Society, 2016, pp. 11–20. [Online]. Available: https://doi.org/10.1109/WICSA.2016.10

[2] B. Kitchenham and S. Charters, "Guidelines for Performing Systematic Literature Reviews in Software Engineering," 2007. [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.117.471&rep=rep1&type=pdf

[3] M. Konersmann, A. Kaplan, T. Kühn, R. Heinrich, A. Koziolek, R. Reussner, J. Jürjens, M. al Doori, N. Bolz, M. Ehl, D. Fuchß, K. Großer, S. Hahner, J. Keim, M. Lohr, T. Sağlam, S. Schulz, and J.-P. Töberg, "Dataset for 'Evaluation Methods and Replicability of Software Architecture Research Objects'," Jan. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6044059

[4] R. Wieringa, N. Maiden, N. Mead, and C. Rolland, "Requirements engineering paper classification and evaluation criteria: a proposal and a discussion," *Requirements Engineering*, vol. 11, no. 1, pp. 102–107, nov 2005. [Online]. Available: https://doi.org/10.1007/s00766-005-0021-6

[5] S. Götz, "Supporting systematic literature reviews in computer science: The systematic literature review toolkit," in *Proceedings of the 21st ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, ser. MODELS '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 22–26. [Online]. Available: https://doi.org/10.1145/3270112.3270117

[6] P. Ralph, S. Baltes, D. Bianculli, Y. Dittrich, M. Felderer, R. Feldt, A. Filieri, C. A. Furia, D. Graziotin, P. He, R. Hoda, N. Juristo, B. A. Kitchenham, R. Robbes, D. Méndez, J. Molleri, D. Spinellis, M. Staron, K. Stol, D. A. Tamburri, M. Torchiano, C. Treude, B. Turhan, and S. Vegas, "ACM SIGSOFT Empirical Standards," *CoRR*, vol. abs/2010.03525, 2020. [Online]. Available: https://arxiv.org/abs/2010.03525

[7] R. Feldt@inproceedingsFeldtM10, author = Robert Feldt and Ana Magazinius, title = Validity Threats in Empirical Software Engineering Research - An Initial Survey, booktitle = Proceedings of the 22nd International Conference on Software Engineering & Knowledge Engineering (SEKE'2010), Redwood City, San Francisco Bay, CA, USA, July 1 - July 3, 2010, pages = 374–379, publisher = Knowledge Systems Institute Graduate School, year = 2010, timestamp = Thu, 12 Mar 2020 11:30:50 +0100, biburl = https://dblp.org/rec/conf/seke/FeldtM10.bib, bibsource = dblp computer science bibliography, https://dblp.org and A. Magazinius, "Validity threats in empirical software engineering research - an initial survey," in *Proceedings of the 22nd International Conference on Software Engineering & Knowledge Engineering (SEKE'2010), Redwood City, San Francisco Bay, CA, USA, July 1 - July 3, 2010*. Knowledge Systems Institute Graduate School, 2010, pp. 374–379.

[8] S. Haselböck, R. Weinreich, and G. Buchgeher, "Decision models for microservices: Design areas, stakeholders, use cases, and requirements," in *Software Architecture - 11th European Conference, ECSA 2017, Canterbury, UK, September 11-15, 2017, Proceedings*, ser. Lecture Notes in Computer Science, A. Lopes and R. de Lemos, Eds., vol. 10475. Springer, 2017, pp. 155–170. [Online]. Available: https://doi.org/10.1007/978-3-319-65831-5_11

[9] "Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models," International Organization for Standardization, Standard, Mar. 2011.

[10] I. Taverniers, M. De Loose, and E. Van Bockstaele, "Trends in quality in the analytical laboratory. ii. analytical method validation and quality assurance," *TrAC Trends in Analytical Chemistry*, vol. 23, no. 8, pp. 535–552, 2004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165993604030031

[11] P. Runeson and M. Höst, "Guidelines for Conducting and Reporting Case Study Research in Software Engineering," *Empirical software engineering*, vol. 14, no. 2, pp. 131–164, 2009. [Online]. Available: https://doi.org/10.1007/s10664-008-9102-8

[12] P. Runeson, M. Host, A. Rainer, and B. Regnell, *Case Study Research in Software Engineering: Guidelines and Examples*, 1st ed. Wiley Publishing, 2012.

[13] B. G. Glaser and A. L. Strauss, *Discovery of Grounded Theory: Strategies for Qualitative Research*. Aldine de Gruyter, 1967.

[14] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering*. Springer Berlin Heidelberg, 2012.

[15] C. Wohlin, *Experimentation in software engineering : an introduction*. Boston: Kluwer Academic, 2000.

[16] S. Gasson, "Rigor in Grounded Theory Research: An Interpretive Perspective on Generating Theory from Qualitative Field Studies," in *The handbook of information systems research*. IGI Global, 2004, pp. 79–102.

[17] T. Cook, *Quasi-experimentation : design & analysis issues for field settings*. Boston: Houghton Mifflin, 1979.

[18] R. K. Yin, *Case Study Research and Applications*. Sage Publications Ltd., Jan. 2017.

[19] P. Genfer and U. Zdun, "Identifying Domain-Based Cyclic Dependencies in Microservice APIs Using Source Code Detectors," in *Software Architecture - 15th European Conference, ECSA 2021*, ser. Lecture Notes in Computer Science, S. Biffl, E. Navarro, W. Löwe, M. Sirjani, R. Mirandola, and D. Weyns, Eds. Cham: Springer International Publishing, 2021, pp. 207–222.

[20] M. Bhat, K. Shumaiev, A. Biesdorf, U. Hohenstein, and F. Matthes, "Automatic extraction of design decisions from issue management systems: A machine learning based approach," in *Software Architecture - 11th European Conference, ECSA 2017, Canterbury, UK, September 11-15, 2017, Proceedings*, ser. Lecture Notes in Computer Science, A. Lopes and R. de Lemos, Eds., vol. 10475. Springer, 2017, pp. 138–154. [Online]. Available: https://doi.org/10.1007/978-3-319-65831-5_10

[21] C. Paterson and R. Calinescu, "Accurate analysis of quality properties of software with observation-based markov chain refinement," in *2017 IEEE International Conference on Software Architecture, ICSA 2017, Gothenburg, Sweden, April 3-7, 2017*. IEEE Computer Society, 2017, pp. 121–130. [Online]. Available: https://doi.org/10.1109/ICSA.2017.14

[22] J. Keim, S. Schulz, D. Fuchß, C. Kocher, J. Speit, and A. Koziolek, "Trace Link Recovery for Software Architecture Documentation," in *Software Architecture - 15th European Conference, ECSA 2021*, ser. Lecture Notes in Computer Science, S. Biffl, E. Navarro, W. Löwe, M. Sirjani, R. Mirandola, and D. Weyns, Eds. Cham: Springer International Publishing, 2021, pp. 101–116.

[23] I. Pigazzini, F. A. Fontana, and A. Maggioni, "Tool support for the migration to microservice architecture: An industrial case study," in *Software Architecture - 13th European Conference, ECSA 2019, Paris, France, September 9-13, 2019, Proceedings*, ser. Lecture Notes in Computer Science, T. Bures, L. Duchien, and P. Inverardi, Eds., vol. 11681. Springer, 2019, pp. 247–263. [Online]. Available: https://doi.org/10.1007/978-3-030-29983-5_17

[24] J. L. M. Filho, L. S. Rocha, R. M. C. Andrade, and R. Britto, "Preventing erosion in exception handling design using static-architecture conformance checking," in *Software Architecture - 11th European Conference, ECSA 2017, Canterbury, UK, September 11-15, 2017, Proceedings*, ser. Lecture Notes in Computer Science, A. Lopes and R. de Lemos, Eds., vol. 10475. Springer, 2017, pp. 67–83. [Online]. Available: https://doi.org/10.1007/978-3-319-65831-5_5

[25] A. Musil, J. Musil, D. Weyns, and S. Biffl, "Continuous adaptation management in collective intelligence systems," in *Software Architecture - 13th European Conference, ECSA 2019, Paris, France, September 9-13, 2019, Proceedings*, ser. Lecture Notes in Computer Science, T. Bures, L. Duchien, and P. Inverardi, Eds., vol. 11681. Springer, 2019, pp. 109–125. [Online]. Available: https://doi.org/10.1007/978-3-030-29983-5_8

[26] A. Avritzer, V. Ferme, A. Janes, B. Russo, H. Schulz, and A. van Hoorn, "A quantitative approach for the assessment of microservice architecture deployment alternatives by automated performance testing," in *Software Architecture - 12th European Conference on Software Architecture, ECSA 2018, Madrid, Spain, September 24-28, 2018, Proceedings*, ser. Lecture Notes in Computer Science, C. E. Cuesta, D. Garlan, and J. Pérez, Eds., vol. 11048. Springer, 2018, pp. 159–174. [Online]. Available: https://doi.org/10.1007/978-3-030-00761-4_11

[27] M. Artac, T. Borovsak, E. D. Nitto, M. Guerriero, D. Perez-Palacin, and D. A. Tamburri, "Infrastructure-as-code for data-intensive architectures: A model-driven development approach," in *IEEE International Conference on Software Architecture, ICSA 2018, Seattle, WA, USA, April 30 - May 4, 2018*. IEEE Computer Society, 2018, pp. 156–165. [Online]. Available: https://doi.org/10.1109/ICSA.2018.00025

[28] L. Nunes, N. Santos, and A. R. Silva, "From a monolith to a microservices architecture: An approach based on transactional

contexts," in *Software Architecture - 13th European Conference, ECSA 2019, Paris, France, September 9-13, 2019, Proceedings*, ser. Lecture Notes in Computer Science, T. Bures, L. Duchien, and P. Inverardi, Eds., vol. 11681. Springer, 2019, pp. 37–52. [Online]. Available: https://doi.org/10.1007/978-3-030-29983-5_3

[29] J. F. Almeida and A. R. Silva, "Monolith migration complexity tuning through the application of microservices patterns," in *Software Architecture - 14th European Conference, ECSA 2020, L'Aquila, Italy, September 14-18, 2020, Proceedings*, ser. Lecture Notes in Computer Science, A. Jansen, I. Malavolta, H. Muccini, I. Ozkaya, and O. Zimmermann, Eds., vol. 12292. Springer, 2020, pp. 39–54. [Online]. Available: https://doi.org/10.1007/978-3-030-58923-3_3

[30] R. Yasaweerasinghelage, M. Staples, I. Weber, and H. Paik, "Predicting the performance of privacy-preserving data analytics using architecture modelling and simulation," in *IEEE International Conference on Software Architecture, ICSA 2018, Seattle, WA, USA, April 30 - May 4, 2018*. IEEE Computer Society, 2018, pp. 166–175. [Online]. Available: https://doi.org/10.1109/ICSA.2018.00026

[31] Y. Zhang, B. Liu, L. Dai, K. Chen, and X. Cao, "Automated microservice identification in legacy systems with functional and non-functional metrics," in *2020 IEEE International Conference on Software Architecture, ICSA 2020, Salvador, Brazil, March 16-20, 2020*. IEEE, 2020, pp. 135–145. [Online]. Available: https://doi.org/10.1109/ICSA47634.2020.00021

[32] J. C. S. Santos, A. Peruma, M. Mirakhorli, M. Galster, J. V. Vidal, and A. Sejfia, "Understanding software vulnerabilities related to architectural security tactics: An empirical investigation of chromium, PHP and thunderbird," in *2017 IEEE International Conference on Software Architecture, ICSA 2017, Gothenburg, Sweden, April 3-7, 2017*. IEEE Computer Society, 2017, pp. 69–78. [Online]. Available: https://doi.org/10.1109/ICSA.2017.39

[33] C. Islam, M. A. Babar, and S. Nepal, "Architecture-centric support for integrating security tools in a security orchestration platform," in *Software Architecture - 14th European Conference, ECSA 2020, L'Aquila, Italy, September 14-18, 2020, Proceedings*, ser. Lecture Notes in Computer Science, A. Jansen, I. Malavolta, H. Muccini, I. Ozkaya, and O. Zimmermann, Eds., vol. 12292. Springer, 2020, pp. 165–181. [Online]. Available: https://doi.org/10.1007/978-3-030-58923-3_11

[34] S. E. Sim, S. M. Easterbrook, and R. C. Holt, "Using Benchmarking to Advance Research: A Challenge to Software Engineering," in *Proceedings of the 25th International Conference on Software Engineering, May 3-10, 2003, Portland, Oregon, USA*, L. A. Clarke, L. Dillon, and W. F. Tichy, Eds. IEEE Computer Society, 2003, pp. 74–83. [Online]. Available: https://doi.org/10.1109/ICSE.2003.1201189

[35] W. Hasselbring, "Benchmarking as empirical standard in software engineering research," in *EASE 2021: Evaluation and Assessment in Software Engineering, Trondheim, Norway, June 21-24, 2021*, R. Chitchyan, J. Li, B. Weber, and T. Yue, Eds. ACM, 2021, pp. 365–372. [Online]. Available: https://doi.org/10.1145/3463274.3463361

[36] A. Aleti, B. Buhnova, L. Grunske, A. Koziolek, and I. Meedeniya, "Software architecture optimization methods: A systematic literature review," *IEEE Transactions on Software Engineering*, vol. 39, no. 5, pp. 658–683, 2013. [Online]. Available: https://doi.org/10.1109/TSE.2012.64

[37] P. V. Cardoso, R. W. A. Fazul, and P. P. Barcelos, "Employment of optimal approximations on apache hadoop checkpoint technique for performance improvements," in *2020 IEEE International Conference on Software Architecture, ICSA 2020, Salvador, Brazil, March 16-20, 2020*. IEEE, 2020, pp. 1–10. [Online]. Available: https://doi.org/10.1109/ICSA47634.2020.00009

[38] ACM, "Artifact Review and Badging Version 1.1," 2020. [Online]. Available: https://www.acm.org/publications/policies/artifact-review-and-badging-current

[39] N. Ferro and D. Kelly, "Sigir initiative to implement acm artifact review and badging," *SIGIR Forum*, vol. 52, no. 1, p. 4–10, Aug. 2018. [Online]. Available: https://doi.org/10.1145/3274784.3274786

[40] K. Petersen and Ç. Gencel, "Worldviews, Research Methods, and their Relationship to Validity in Empirical Software Engineering Research," in *2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement, Ankara, Turkey, October 23-26, 2013*. IEEE Computer Society, 2013, pp. 81–89. [Online]. Available: https://doi.org/10.1109/IWSM-Mensura.2013.22

[41] N. Qureshi, M. Usman, and N. Ikram, "Evidence in Software Architecture, A Systematic Literature Review," in *17th International Conference on Evaluation and Assessment in Software Engineering, EASE '13, Porto de Galinhas, Brazil, April 14-16, 2013*, F. Q. B. da Silva, N. J. Juzgado, and G. H. Travassos, Eds. ACM, 2013, pp. 97–106. [Online]. Available: https://doi.org/10.1145/2460999.2461014

[42] M. Shaw, "Writing Good Software Engineering Research Papers: Mini-tutorial." USA: IEEE Computer Society, 2003.

[43] C. Theisen, M. Dunaiski, L. Williams, and W. Visser, "Software Engineering Research at the International Conference on Software Engineering in 2016," *ACM SIGSOFT Software Engineering Notes*, vol. 42, no. 4, pp. 1–7, 2018. [Online]. Available: https://doi.org/10.1145/3149485.3149496

[44] A. Bertolino, A. Calabrò, F. Lonetti, E. Marchetti, and B. Miranda, "A Categorization Scheme for Software Engineering Conference Papers and its Application," *Journal of Systems and Software*, vol. 137, pp. 114–129, 2018. [Online]. Available: https://doi.org/10.1016/j.jss.2017.11.048

[45] R. Glass, I. Vessey, and V. Ramesh, "Research in Software Engineering: An Analysis of the Literature," *Information and Software Technology*, vol. 44, no. 8, pp. 491–506, 2002. [Online]. Available: https://doi.org/10.1016/S0950-5849(02)00049-6

[46] M. V. Zelkowitz and D. Wallace, "Experimental Validation in Software Engineering," *Information and Software Technology*, vol. 39, no. 11, pp. 735–743, 1997. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950584997000256

[47] H. K. Wright, M. Kim, and D. E. Perry, "Validity concerns in software engineering research," in *Proceedings of the Workshop on Future of Software Engineering Research, FoSER 2010, at the 18th ACM SIGSOFT International Symposium on Foundations of Software Engineering, 2010, Santa Fe, NM, USA, November 7-11, 2010*, G. Roman and K. J. Sullivan, Eds. ACM, 2010, pp. 411–414. [Online]. Available: https://doi.org/10.1145/1882362.1882446

[48] M. Cruz, B. Bernárdez, A. Durán, J. A. Galindo, and A. Ruiz-Cortés, "Replication of studies in empirical software engineering: A systematic mapping study, from 2013 to 2018," *IEEE Access*, vol. 8, pp. 26 773–26 791, 2020. [Online]. Available: https://doi.org/10.1109/ACCESS.2019.2952191

[49] A. Snyder, "How to Get Your Paper Accepted at OOPSLA," *SIGPLAN Not.*, vol. 26, no. 11, p. 359–363, Nov. 1991. [Online]. Available: https://doi.org/10.1145/118014.117983

[50] J. Gray and B. Rumpe, "How to Write a Successful SoSyM Submission," *Softw. Syst. Model.*, vol. 15, no. 4, pp. 929–931, 2016. [Online]. Available: https://doi.org/10.1007/s10270-016-0558-5

[51] M. Galster, D. Weyns, A. Tang, R. Kazman, and M. Mirakhorli, "From craft to science: the road ahead for empirical software engineering research," in *Proceedings of the 40th International Conference on Software Engineering: New Ideas and Emerging Results, ICSE (NIER) 2018, Gothenburg, Sweden, May 27 - June 03, 2018*, A. Zisman and S. Apel, Eds. ACM, 2018, pp. 77–80. [Online]. Available: https://doi.org/10.1145/3183399.3183421

[52] D. Falessi, M. A. Babar, G. Cantone, and P. Kruchten, "Applying empirical software engineering to software architecture: challenges and lessons learned," *Empir. Softw. Eng.*, vol. 15, no. 3, pp. 250–276, 2010. [Online]. Available: https://doi.org/10.1007/s10664-009-9121-0

[53] D. Falessi, P. Kruchten, and G. Cantone, "Issues in applying empirical software engineering to software architecture," in *Software Architecture, First European Conference, ECSA 2007, Aranjuez, Spain, September 24-26, 2007, Proceedings*, ser. Lecture Notes in Computer Science, F. Oquendo, Ed., vol. 4758. Springer, 2007, pp. 257–262. [Online]. Available: https://doi.org/10.1007/978-3-540-75132-8_20