

Large-Scale Evaluation of Android-Related Research Artifacts

Philipp Holzinger 

Fraunhofer SIT | ATHENE – National Research Center for Applied Cybersecurity,
Darmstadt, Germany

Philipp Rajwa 

Fraunhofer SIT | ATHENE – National Research Center for Applied Cybersecurity,
Darmstadt, Germany

Abstract

Background. Many computer science papers rely on software prototypes that implement novel concepts or generate data for empirical evaluation. To support confirmatory research and reuse, more venues now request code and data submissions and conduct artifact evaluations. However, concerns about a reproducibility crisis remain, as studies show that only a small share of published artifacts is actually accessible and working for reuse or reproducing results.

Aims. To shed more light on this in the context of Android-related research, we investigate to what extent research artifacts are available, functional, and whether they allow us to reproduce published experimental results. Moreover, we want to understand the reasons why research artifacts might fail to fulfill these objectives.

Method. We conducted a large-scale empirical study of Android-related research artifacts that were published between 2011 and 2025. For an initial set of 269 papers, we attempted to locate their associated artifacts, assess whether they can be obtained and executed, and evaluate whether they reproduce the expected output.

Results. Starting from 269 papers, only 163 artifacts were still available. Among these, 118 papers published experimental results, yet we were only able to establish 10 experimental setups. Ultimately, only 8 artifacts could fully or partially reproduce the expected output.

Conclusions. Our findings provide further evidence for reproducibility challenges in Android-related research. Based on these sobering results, we propose a set of recommendations to strengthen the perceived value of artifacts and to improve their availability, quality, and visibility.

2012 ACM Subject Classification General and reference → Empirical studies; Software and its engineering → Software post-development issues; Information systems → Digital libraries and archives

Keywords and phrases research artifacts, reproducibility, reuse, Android, empirical study

Digital Object Identifier 10.4230/LIPIcs.ESEM.2026.32

Category Technical Track Paper

Supplementary Material *Dataset:* <https://doi.org/10.6084/m9.figshare.33256998>

Funding This research work was supported by the National Research Center for Applied Cybersecurity ATHENE. ATHENE is funded jointly by the German Federal Ministry of Research, Technology and Space and the Hessian Ministry of Science and Research, Arts and Culture.

1 Introduction

The number of publications in computer science increases steadily. In 2025, more than 400.000 papers have been published at conferences, workshops, and in journals [11]. The need to have scientific output that is independently verifiable has emerged as a critical factor for



© Philipp Holzinger and Philipp Rajwa;

licensed under Creative Commons License CC-BY 4.0

20th International Symposium on Empirical Software Engineering and Measurement (ESEM 2026).

Editors: Robert Feldt, Maria Paasivaara, Daniel Mendez, Stefan Wagner, and Marvin Muñoz Barón;

Article No. 32; pp. 32:1–32:21



Leibniz International Proceedings in Informatics

LIPIcs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

scientific progress. Reusability and reproducibility enable confirmatory research that helps consolidate or falsify scientific findings, and allows researchers to build on top of previously established knowledge [27].

Yet, in the past years, several researchers have declared a *reproducibility crisis* [4, 25, 35, 3], stating that many research publications and their corresponding artifacts cannot be used to reproduce the scientific findings originally presented in the papers.

The research community in the fields of computer science and software engineering has responded to this by establishing independent artifact evaluation processes for many of their conferences. While this is clearly a step forward in terms of scientific rigor, as of today, the reproducibility crisis does not seem to be resolved yet. In a recent survey of 101 software engineering researchers conducted by Azevedo et al. [3], 74% agreed that there is a crisis in software engineering research concerning repeatability, reproducibility, and replicability.

This is supported by an empirical study of Olszewski et al. [28] who evaluated 298 artifacts related to machine learning security published at top-tier security venues. As they show, 56% of artifacts were not runnable, and only 20% were runnable and yielded the expected results.

Further, a recent case study by Muttakin et al. [26] of 100 randomly sampled ICSE artifacts showed that only 40% of them were executable, and only 35% in this subset reproduced the expected results.

Although the generalizability of these studies is limited, they nevertheless raise the question of whether the entire computer science community is inadequately addressing the challenges of reusability and reproducibility.

To shed more light on this important topic, in this study, we perform a large-scale evaluation of Android-related research artifacts, to study their availability, functionality, and reproducibility. The reasons for focusing on Android are threefold: first, there is an extensive body of related research published at various conferences over an extended period of time; second, work in this area is more likely to include software-based artifacts with quantitative results; and third, we possess strong technical expertise in this domain, which makes our manual assessments more substantial and meaningful.

As input to our study, we received a comprehensive dataset of the contents of the ACM Digital Library up to the year 2017, and further considered ICSE publications from 2018 to 2025. Filtered by date, publication type and relevance to Android, we ended up with a final dataset of 269 relevant papers that also reference an artifact. This includes 32 papers that had been awarded at least one badge by an artifact evaluation committee.

Our first finding is that we could only retrieve 163 artifacts by following the instructions provided in the papers, and in some cases even artifacts with an *ACM Artifacts Available* badge were no longer available. Among the available artifacts, only a fraction was functional, and we often had to fix fundamental issues in the code provided, such as hard-coded paths, or flawed dependency management. For reproducibility, we focused on a subset of 118 papers with software-based experiments. Despite our best efforts, we fully reproduced the results for only 3 papers; in most cases, we could not even set up the experiments due to non-functional or incomplete artifacts (among other reasons). While none of the artifacts had a reproducibility badge and many were not designed for that purpose, these results are sobering in light of the perceived reproducibility crisis.

In this paper, we make the following original contributions:

- A large-scale evaluation of Android-related artifacts regarding availability, functionality, and reproducibility
- Recommendations to improve the perceived value of artifacts, their availability, quality, and visibility

This paper is structured as follows. Section 2 introduces relevant background, including terminology, and related work. Section 3 presents our methodology, as well as our research questions. Accordingly, Section 4 gives detailed account of the results of our empirical study. In Section 5 we discuss our findings and in Section 6 we present limitations and threats to validity. Finally, Section 7 provides recommendations based on the lessons we learned during our study, and Section 8 concludes.

2 Background and Related Work

In this section, we provide background information on artifacts and evaluation processes, and further present related work.

2.1 Research Artifacts

Software implementations are a core component of many scientific contributions in computer science, serving as the basis for experimental studies. Other researchers can independently verify these studies by reproducing or replicating the experiments. To avoid confusing terminology [37], ACM has adopted the following definitions, which we will also use in this paper [12]:

- *Repeatability*: “Same team, same experimental setup”
- *Reproducibility*: “Different team, same experimental setup”
- *Replicability*: “Different team, different experimental setup”

To support confirmatory research and enable other researchers to build upon existing work, authors can supply their experiments – either in part or in full – as artifacts accompanying their papers. ACM defines an artifact as a “digital object that was either created by the authors to be used as part of the study or generated by the experiment itself. For example, artifacts can be software systems, scripts used to run experiments, input datasets, raw data collected in the experiment, or scripts used to analyze results” [12]. Throughout this paper, we use the term *artifact* to describe the entirety of files and supplementary materials linked to a paper, irrespective of their types, contents, or number.

The number of publications that provide an artifact increases over time [22]. Whether papers with artifacts are also cited more often is not clear. While there is previous work that supports this theory [5, 13], a study performed by Winter et al. [42] could not find a statistically significant difference.

While the fundamental development towards more reproducible research results is welcomed, recent studies lament the missing availability [17, 22] or reusability [7] of many software artifacts and even their failure to reproduce experimental results [7, 28].

2.2 Artifact Evaluation

Recognizing the need for higher availability and quality of research artifacts, different publication venues have introduced artifact evaluation processes. Authors are encouraged to submit their artifacts, an artifact evaluation committee reviews them and awards badges that confirm, e.g., artifact availability, or functionality. Different badging frameworks have been proposed, but in the context of software engineering, programming languages and security, the ACM badging framework [12] is most prevalent.

It comprises the following three badges commonly awarded by committees:

- *ACM Artifacts Available*: “Author-created artifacts relevant to this paper have been placed on a publically accessible archival repository. A DOI or link to this repository along with a unique identifier for the object is provided.” [12]
- *ACM Artifacts Evaluated Functional*: “The artifacts associated with the research are found to be documented, consistent, complete, exercisable, and include appropriate evidence of verification and validation.” [12]
- *ACM Artifacts EvaluatedReusable*: “The artifacts associated with the paper are of a quality that significantly exceeds minimal functionality.” [12]

In addition to these three badges, the ACM framework further defines an *ACM Artifacts Results Reproduced* and *ACM Artifacts Results Replicated* badge, which require subsequent, independent studies that confirm results presented in previous work.

The ACM badge criteria have found widespread adoption [16, 42], even beyond ACM venues [15]. In 2025, 7 out of 9 CORE2023 A*-ranked conferences in Software Engineering [9] encourage artifact submissions and make explicit references to the ACM badge framework in their call for submissions.

However, even among leading software engineering conferences, there is no consensus on how artifact evaluation should be conducted and what the requirements are.

Out of 25 A-ranked conferences, 23 encourage artifact submission and 16 use a badge framework. However, only 10 award all three ACM artifact evaluation badges, 2 award another subset of the ACM badges, and 4 use their own badge definitions or use another badge framework entirely.

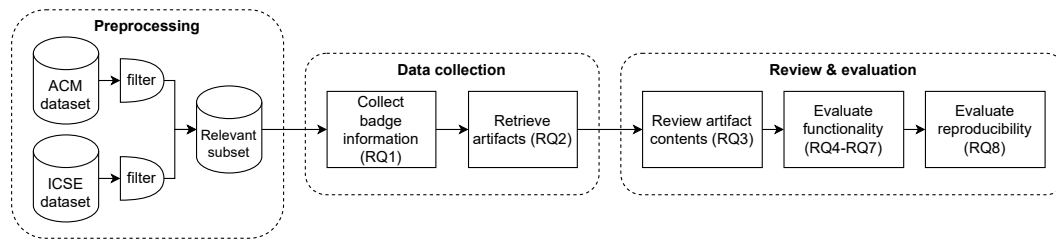
Even fundamental requirements differ between conferences. For example, PLDI 2025 “strongly encourages” submitting artifacts as a Docker container [36], while POPL 2025 does not recommend using Docker due to “cross-platform compatibility issues” [38]. Further, some conferences require authors to publish their artifacts by strictly using one specific artifact archival service [36], while others allow different services [19, 2, 33].

There are even inconsistencies within the same conference when considering back-to-back years. For example, in 2024, ICSE awarded only two badges instead of three, and despite referencing the ACM badging framework, the call used terminology and requirements not in line with the framework [30]. This was fundamentally different in 2025 [31].

In summary, there are efforts to systematize artifact evaluation in the research community, but there is no broad consensus yet.

2.3 Related Work

There is previous work that has studied artifact evaluation processes, expectations, and the availability of artifacts. Liu et al. [22] found that the number of papers with artifacts at top-tier software engineering conferences has increased from 2017 to 2022, but also noted that 30% of artifacts from 2017 were no longer available in 2024. Hermann et al. [16] surveyed members of artifact evaluation committees with respect to their expectations. They found that certain expectations are not adequately communicated in calls and further highlight problematic inconsistencies in the use of terminology. Timperley et al. [41] conducted a survey to identify challenges that influence the quality of artifacts, pointing out insufficient reward for authors and reviewers. Heumüller et al. [17] found that out of 604 ICSE papers from 2007 to 2017 claiming to contain artifacts, only 27% contained working links to these artifacts in 2020. In 2022, Frachtenberg [13] surveyed 2439 papers from system conferences and found that 13% of artifacts were no longer available four years after publication.



■ **Figure 1** Overview of the multistep process we used in this study.

There is also prior work that had a focus on reusability and reproducibility. Collberg and Proebsting [7] found that out of 402 computer systems papers with artifacts that were published in ACM conferences and journals around 2012, only 56% were available and only 48% could be built by the authors. In 2016, Reaves et al. [40] performed a systematization of papers on Android security research, and finally evaluated 7 selected application analysis tools with respect to reusability. In 2021, Daoudi et al. [10] could only fully reproduce 1 out of 5 Android malware detector approaches despite putting in significant effort and contacting the original authors. In 2024, Mineau and Lalande [23] evaluated the reusability of 22 Android static analysis tools on a novel input dataset, and found that 12 of them were unusable. Olszewski et al. [28] surveyed 750 machine learning papers from 2013 to 2022 that were published at top-tier security conferences. They could only reproduce 20% of the results and found no statistically significant difference regarding the availability of artifacts after the introduction of artifact evaluation committees. Finally, Muttakin et al. [26] performed a case study of 100 randomly sampled ICSE artifacts and showed that only 40% of them were executable, and, out of these, only 35% reproduced the originally published results.

To the best of our knowledge, in this paper we present the first large-scale evaluation of Android-related research artifacts, considering availability, functionality, and reproducibility.

3 Methodology

Figure 1 provides an overview of our approach. In the following, we will define the scope of our study, the datasets we used as input for our evaluation, and present the research questions that we aim to answer.

3.1 Scope

We set the focus of our evaluation on papers centered around the Android operating system, including novel approaches to app development, analysis, or testing. For evaluating the reproducibility of papers and their artifacts, we further limit our scope to software-based experiments, because they deliver quantitative results and can be more practically evaluated on a large-scale than, e.g., user and field studies. One reason for choosing Android specifically is that it was released in 2008 [14], and we can hence look back at a large body of publications in that field. Another reason is that we assume that Android-related papers are more likely to have software-based experiments that they make available as artifacts, as compared to, e.g., papers on formal methods or cryptography. One additional reason is that the Android operating system, as well as app development, is well documented, and there is a large community of developers in this area, which is a benefit in our effort to potentially get a large number of artifacts running.

We further limit the timeframe we consider from 2011 to 2025. The starting year is when the software engineering community introduced artifact evaluation processes, and the end year is the last full calendar year at the time we conducted our study.

3.2 Datasets

Identifying papers that fit our scope and have an artifact associated with them requires title, abstract, and full text. Unfortunately, large publication datasets like the dblp (<https://dblp.org>) often contain bibliographic information only, rendering them unsuitable as a data source for our purposes. Instead, we used two alternative datasets as input to our study.

The first dataset is the contents of the ACM Digital Library for the years 1952–2017, that we received as proprietary, structured XML data. A similar dataset with more recent contents was not available to us upon request. Due to its ~ 440.000 publication entries, we were required to apply filters. We first removed all publications for the years before 2011, which reduced the set to ~ 175.000 entries. We then filtered for entries whose publication type was “regular_article”, “orig-research”, “”, or *none*, by which we removed tutorials, columns, interviews, reviews, introductions, keynotes, and similar contents, which left ~ 123.000 entries. Next, we removed entries that had no fulltext content in our dataset, leaving ~ 107.000 papers. We then processed titles and fulltext contents to filter for those entries that were likely to be related to Android, using search terms like “Android”, “mobile app”, and various combinations – the complete rule set is provided in our artifact. The resulting set was a list of ~ 3.000 papers. We manually reviewed these publications to select those that refer to an artifact, and filtered out duplicates in our dataset, misclassified data entries, and papers that matched our search criteria, but, based on a group discussion, were not in our scope. This primarily concerns papers that use Android-based devices for data collection, e.g., for human-centric behavioral experiments [24], or transportation studies [43], without a focus on development or testing. The final list from this dataset contains 174 papers.

To include more recent publications, as a second dataset, we considered all ICSE publications from the years 2018–2025. This venue has an A* CORE ranking [8], features an artifact evaluation process, and has been held for several years. Based on manual review and a group discussion, we selected the subset of Android-related papers that fit our scope and reference an artifact, which resulted in 95 papers.

In total, we examined 269 papers for our evaluation.

3.3 Research questions

Based on the set of papers that we selected, we aim to answer the following research questions:

- RQ1:** How many artifacts received badges?
- RQ2:** How many artifacts are available?
- RQ3:** How many artifacts contain software?
- RQ4:** How many scripts can be executed?
- RQ5:** How many code artifacts can be built?
- RQ6:** How many binaries can be executed?
- RQ7:** How many artifacts are functional?
- RQ8:** How many artifacts can be used to reproduce paper results?

The first question shall enable us to reason about the potential effects of artifact evaluation on properties of the artifacts that we measure in subsequent questions. RQ2 is dedicated to availability, RQ4–7 are concerned with functionality, while the final research question studies reproducibility. We treat scripts, binaries, and compilable code separately, since they are fundamentally different types of software that may vary in terms of their reusability.

■ **Table 1** Combinations of badges awarded to papers in our dataset.

Badge combination	# of papers
<i>ACM Artifacts Available + ACM Artifacts Evaluated Reusable</i>	15
<i>ACM Artifacts Available</i>	7
<i>ACM Artifacts Available + ACM Artifacts Evaluated Functional</i>	3
<i>ACM Artifacts Available + ACM Artifacts Evaluated Functional + ACM Artifacts Evaluated Reusable</i>	1
<i>ACM Artifacts Evaluated Functional</i>	1
<i>ISSTA Artifact Evaluated</i>	1
<i>PLDI Artifact Evaluated</i>	1
<i>ACM Artifacts Evaluated Reusable</i>	1
<i>OOPSLA Artifact Evaluated</i>	1
<i>Artifact evaluated by FSE</i>	1

When attempting to set up artifacts for evaluation that did not work as is, we set a limit to 3 hours, trying to fix errors. We consider this reasonable, as this exceeds the limits set in prior studies for similar tasks [6, 28]. If within this timeframe we did not succeed, we flagged the respective artifact as “non-functional”. Further, we skipped artifacts whose setup would require complex, hard-to-setup distributed system environments in dedicated network infrastructures, and flagged them as “skipped” accordingly.

4 Results

In the following subsections, we present the results of our empirical evaluation for each of the research questions stated above.

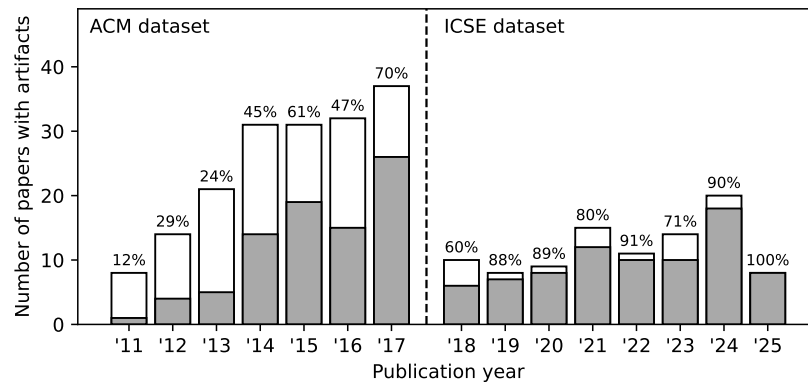
4.1 RQ1: How many artifacts received badges?

In total, 32 out of 269 papers received at least one badge awarded by an artifact evaluation committee. This represents a share of 12%. Table 1 shows the combinations of badges that we found in our dataset. As can be seen, the majority of badges are ACM badges, both in versions 1.0 and 1.1. The most awarded badge in our set is the *ACM Artifacts Available* badge, which was assigned to 26 papers.

Interestingly, although ACM badges were awarded most frequently in our dataset, none of the papers received an *ACM Artifacts Results Reproduced* badge or *ACM Artifacts Results Replicated* badge. We assume that the former was typically not offered by the artifact evaluation committees who reviewed papers in our dataset. The latter badge, as also acknowledged by ACM, may be more suited to awarding it post-publication, which apparently never happened to any of the papers we reviewed.

The share of papers with badges in our dataset is too small to allow for statistical evaluation, however, in the following research questions, we will highlight these papers.

Note that for two papers we found it hard to clearly identify the type of badge that was awarded to the paper, based on the visual indication of the badges on the title pages of the two respective papers. We indicated this in the full list of papers that we provide as an artifact to this publication.



■ **Figure 2** Number of publications with artifacts from 2011 to 2025. White: total number of publications that reference artifacts. Gray: Share of publications whose artifacts are still available.

4.2 RQ2: How many artifacts are available?

For further investigation, we first tried to retrieve the artifacts of the 269 papers in our dataset in the way that is indicated in the respective paper. Typically, papers provided a URL that was supposed to either lead directly to downloadable content, to a repository, or to a project website with further instructions on how to retrieve a copy of the artifact. In 7 cases, artifacts were only available upon request, e.g., by sending an email or submitting an online form.

Figure 2 shows an overview of how many artifacts we were able to retrieve at least partially. Partial availability occurs when only a subset of the specified URLs is accessible and leads to the intended content. As can be seen, in total, we were able to retrieve the artifacts of 163 papers, which represents a share of 61%. As indicated in Figure 2, the share of available artifacts declines over time, i.e., papers published between 2011 and 2013 have a much lower availability rate than papers published more recently. It is still noteworthy, though, that even recent years do not have an availability rate of 100%.

We further found indication that the way artifacts are provided has an impact on availability over time. If we consider only the subset of papers whose artifacts were originally provided via well-known, third-party hosters¹, the availability rate was significantly higher: 92 out of 108 artifacts were available, a share of 85%. This indicates that artifacts hosted on university websites, private blogs, etc. are less likely to remain available.

We found the lowest availability rate for artifacts that were supposed to be available upon request, with no downloadable content whatsoever. All our requests to retrieve the 7 artifacts resulted in no response and hence complete unavailability of these artifacts.

Interestingly, 5 out of the 32 papers with badges were not available anymore, two of which received an *ACM Artifacts Available* badge, one *ISSTA Artifact Evaluated* badge, one *PLDI Artifact Evaluated* badge, and one *Artifact evaluated by FSE* badge.

Note that we have only relied on the instructions presented in the original papers. We have not conducted additional online research to find alternative ways for retrieving any artifacts in cases where we did not succeed.

¹ github.com, github.io, gitlab.com, bitbucket.org, sourceforge.net, zenodo.org, softwareheritage.org, figshare.com

■ **Table 2** Artifacts that contain any type of software.

Combination of types of software	# of papers
Code-only	40
Scripts-only	40
Scripts + code	27
Scripts + code + binaries	15
Code + binaries	12
Scripts + binaries	7
Binaries-only	4

■ **Table 3** Number of script artifacts that we were able to execute. In parentheses: subset of these artifacts that required modifications to make them executable.

Executable	# of artifacts
No	43 (n/a)
Yes	34 (11 32%)
Partially	10 (7 70%)
Skipped*	2 (n/a)

*due to time & effort constraints

4.3 RQ3: How many artifacts contain software?

We divided the types of software that may be included in an artifact into three categories: scripts, code, and binaries. We classified artifacts from the perspective of a user who wants to reuse the artifact and hence applied the following guidelines.

If an artifact contained substantial logic in any script language that represents a part of the scientific contribution of the original authors, we flagged it as an artifact that contains scripts. This does not include simple make/build scripts, setup shell scripts, etc.

Similarly, we flagged all artifacts to contain code if there is logic implemented in a compiled language, like Java or C++, that needs to be compiled and executed for reusing the artifact. If an artifact provided source code, e.g., as input files to an analysis implemented in a script, this did not qualify.

Lastly, we flagged all artifacts to contain binaries if the artifact contains a pre-compiled executable that needs to be run for reusing the artifact. If, for example, binary executables are provided merely as input to an analysis implemented in a script, we consider them input data, and did not flag this as an artifact that contains binaries.

While this is not a formal classification model, we consider it adequate to roughly describe the contents of artifacts in such a way that it prepares users who want to reuse an artifact for potential challenges.

The results of our classification are shown in Table 2. In total, 145 out of the available 163 artifacts contain software, representing a share of 89%. Code and/or scripts are contained in all artifacts but 4, while 107 artifacts do not contain binaries. A total of 80 artifacts are either code-only or scripts-only artifacts.

4.4 RQ4: How many scripts can be executed?

For each of the 89 artifacts that contain scripts we attempted to execute them. In cases where execution of a script as-is did not succeed, we tried to resolve any error messages, syntax errors, etc. with the goal of making it executable in the way originally intended by the authors.

Table 3 shows an overview of the results of this effort. As can be seen, we were able to execute the scripts successfully for only 34 of the 89 artifacts containing scripts, representing 38%. Only 23 of these 34 worked out of the box, the remaining 11 required modifications from us, e.g., due to syntax errors, hard-coded paths, etc. An additional 10 artifacts contained scripts that we were able to execute partially, 7 of which required modifications from our side.

■ **Table 4** Number of code artifacts that we were able to build. In parentheses: subset of these artifacts that required modifications to build them.

Buildable	# of artifacts
Yes	47 (28 60%)
No	44 (n/a)
Partially	2 (1 50%)
Skipped*	1 (n/a)

*due to time & effort constraints

■ **Table 5** Number of binary artifacts that we were able to execute.

Executable	# of artifacts
Yes	30
No	6
Partially	1
Skipped*	1

*due to time & effort constraints

Full execution of these scripts was not possible, e.g., because they relied on unavailable input data/third-party services, or other software in the respective artifact that did not work.

We restricted our efforts in terms of time and computational/infrastructural resources which resulted in 2 artifacts being skipped for further evaluation.

The remaining 43 artifacts contained scripts that, despite our best efforts, we were unable to execute successfully, representing 48% of the 89 artifacts that contained scripts in our set.

Out of these 43 artifacts, 8 belong to papers that received an *ACM Artifacts Evaluated Reusable* badge, as well as 1 that belongs to a paper that was awarded an *ACM Artifacts Evaluated Functional* badge. Our results do not comply with these badges. An additional 2 papers whose script artifacts we were unable to execute received an *ACM Artifacts Available* badge only, which is in line with our results, as we were able to retrieve the artifacts.

4.5 RQ5: How many code artifacts can be built?

In our set of 163 available artifacts, 94 contain code that needs to be compiled for reuse. We attempted this for each one of them and Table 4 shows an overview of our results.

As can be seen, we were able to build 47 code artifacts fully, and 2 additional artifacts partially. More than half of these artifacts required modifications from our side. Reasons for this include minor code errors, various issues concerning project/directory structures, hard-coded paths, dependency version mismatches, etc., as well as changes to make code artifacts compatible with more modern build tool chains.

We again skipped 1 artifact due to effort constraints, which is also one of the two artifacts we skipped in RQ4.

Out of the 44 code artifacts that we were unable to build, 5 received badges, including 3 papers that received *ACM Artifacts Evaluated Reusable* badges, and 2 papers that received an *ACM Artifacts Available* badge only.

4.6 RQ6: How many binaries can be executed?

A total of 38 artifacts contain binaries whose execution is required for artifact reuse. This includes native binaries, as well as pre-compiled Java code, etc. The results of our efforts to run them is shown in Table 5.

As can be seen, 30 of these binary artifacts are runnable as intended. We were able to execute 1 binary artifact only partially, because its functionality depends on input data not provided along with the artifact. One artifact that we skipped already in the two previous research questions due to effort constraints is the one artifact that we also skipped here. The

remaining 6 binary artifacts did not run without errors. They either crashed with unhandled errors, included hard-coded paths, or, in one case, relied on a third-party service no longer available.

In contrast to the previous research questions, we have not attempted to reverse engineer them, or patch them, to get them running.

Out of the 6 binary artifacts that we were unable to execute, 3 received badges. This includes 1 artifact with an *ACM Artifacts Evaluated Functional* badge, as well as 1 with an *ACM Artifacts Evaluated Reusable* badge, and 1 that received an *ACM Artifacts Available* badge only.

4.7 RQ7: How many artifacts are functional?

In this research question, we will investigate how many of the 145 artifacts that contain software are functional. While, at a first glance, this question may appear straightforward, we generally lack a definition of requirements that would allow us to evaluate the proper functionality of an artifact. Due to the lack of well-defined tests or metrics that we could apply, we will retreat to the most fundamental criteria that define functional software: scripts must be executed by the interpreter without error messages, binaries must be executed by the runtime/operating system without crashes or error messages, and code must be buildable and executable without crashes or error messages. We attempted this for each artifact that contains software, using basic user interactions where required, or example inputs/test cases, where provided in the artifacts. The result of this effort is:

- 41 are functional, either with or without fixes from our side
- 94 are not functional, despite our best efforts to get them running
- for 8 we cannot judge proper functionality, e.g., because they require hardware to which we have no access, or depend on commercial third-party services
- we further skipped 2 due to our effort and resource constraints

Note that an artifact can be functional, but still not be suited for reproduction, e.g., because input data is incomplete or missing. Hence, proper functionality should be considered a prerequisite for reproducibility, rather than a synonym.

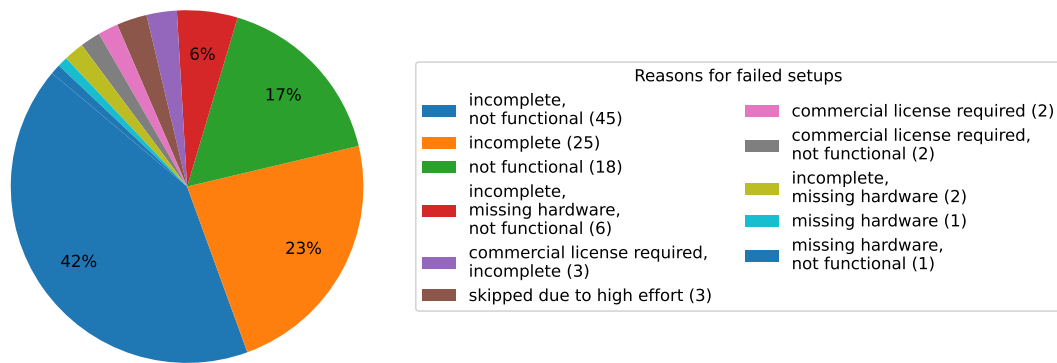
4.8 RQ8: How many artifacts can be used to reproduce paper results?

We answer this research question in three subsequent steps, as described in the following.

Step 1: Filter relevant papers

We first reviewed the 163 papers with available artifacts and selected those that contained evaluations with software-based experiments. In contrast, we skipped, e.g., empirical evaluations that reported on user or field studies, or tool/demo papers that presented prototypes without scientific evaluation. If a paper contained multiple evaluations and only a subset matched our criteria, we included it and focused on the relevant subset of experiments. The result of this is a set of 118 papers. Contained in this set are 24 papers with badges, including the following combinations:

- *ACM Artifacts Available* + *ACM Artifacts Evaluated Reusable*: 14 papers
- *ACM Artifacts Available* + *ACM Artifacts Evaluated Functional*: 3 papers
- *ACM Artifacts Available*: 3 papers
- *ACM Artifacts Available* + *ACM Artifacts Evaluated Functional* + *ACM Artifacts Evaluated Reusable*: 1 paper



■ **Figure 3** Combinations of reasons for why setups for reproducibility failed.

- *ACM Artifacts Evaluated Functional*: 1 paper
- *ACM Artifacts Evaluated Reusable*: 1 paper
- *OOPSLA Artifact Evaluated*: 1 paper

Step 2: Setup experiments

For each of the 118 papers we proceeded by trying to set them up for evaluation.

For 11 of these 118 artifacts, we found the software required for reproduction to be missing entirely, and flagged them *incomplete* accordingly. None of these papers had received a badge.

In 104 of 118 cases, software was provided with the artifact. We skipped 2 of these artifacts because they required commercial licenses to which we had no (free) access. One of them was assigned an *ACM Artifacts Evaluated Functional* badge, the other one received an *ACM Artifacts Evaluated Reusable* badge. Further, we skipped 1 paper that required specific hardware to which we had no access. It had not been assigned a badge. For 91 of 104 artifacts, our attempt for setup failed because they were either incomplete, non-functional, or both. A total of 18 papers in this set received badges, including 13 papers with an *ACM Artifacts Evaluated Reusable* badge, 2 papers with an *ACM Artifacts Evaluated Functional* badge, and 3 with an *ACM Artifacts Available* badge only. Only the remaining 10 of these 104 artifacts resulted in usable setups, including 7 full setups, and 3 partial setups, that allow for evaluating a subset of the experiments presented by the original authors.

In addition, we also flagged 3 of the 118 artifacts as failed setups that we skipped solely due to effort constraints, none of which received a badge.

Figure 3 provides an overview of the reasons for why setups failed.

Step 3: Run experiments & compare results

We finally executed the experiments of the 7 full setups and 3 partial setups and compared the output to what was reported in the corresponding papers.

The outcome of this is as follows:

- 3 of the 7 complete setups fully reproduced the results presented in the papers. One had received an *ACM Artifacts Evaluated Functional* badge, one paper received an *OOPSLA Artifact Evaluated* badge, and one received an *ACM Artifacts Evaluated Reusable* badge.

- 2 of the 7 complete setups only partially reproduced the results reported in the paper. One of these papers received both an *ACM Artifacts Evaluated Functional* and *ACM Artifacts Evaluated Reusable* badge. It was a performance comparison that repeatedly yielded a different outcome in our experiments as compared to the results presented in the paper, a type of measurement notoriously hard to reproduce on different hardware. The other paper had not received any badges.
- 2 of the 7 complete setups failed entirely to reproduce the results in the papers. Note, however, that this includes one artifact that contains a proof-of-concept app that is supposed to showcase a critical bug in the Android operating system [1], i.e., a pass-or-fail experiment. The authors indicated, to the best of their knowledge, which versions of Android were affected, and we spent our best effort to retrieve a phone whose outdated Android version we assumed would be suitable for the experiment. Despite this effort, reproduction did not succeed. The Android versions on the phones used by the original authors were different, though, so it is likely that, on another phone, reproduction would have worked. Unfortunately, we were unable to find a phone whose operating system version matched exactly. Both papers were not assigned any badges.
- the 3 partial setups only partially reproduced the expected results, due to their incompleteness. They had not been assigned any badges.

The 8 papers whose artifacts allowed for at least partial reproduction were published between 2014 and 2025, and there is no correlation between age and reproducibility in our dataset.

5 Discussion

We acknowledge that we evaluated a set of artifacts with regard to reproducibility that, in many cases, may have never been designed for that purpose. However, we cannot limit ourselves to reproducible artifacts only. In many cases, the original authors do not state what purpose their artifact is supposed to serve. Even artifacts that are referred to as “replication package” by their original authors [21] do not necessarily allow for reproduction.

Further, we flagged artifacts as incomplete from the perspective of researchers attempting to reuse them for reproduction. Authors may see this differently, having deliberately provided artifacts not intended for this purpose.

There may be various reasons for why this was the case. For example, legal reasons, including licensing issues, copyright laws, export restrictions, and so forth may have prevented authors from publishing parts of their code or data [41]. This is a well-known problem and it is unclear how the scientific community wants to deal with this in the long term.

Another important issue is that we also flagged attempts to set up experiments as failed in cases where we skipped the evaluation due to effort constraints, missing hardware, or the requirement to order a commercial license. We deliberately chose this way of presenting our results because these are the same issues that other researchers may encounter in their own efforts to reuse a given artifact. To a certain degree, at least, the authors of artifacts can influence how much effort is required for setup, and what the hardware or license requirements are.

6 Limitations and threats to validity

There are various aspects that limit the generalizability of our results. Most importantly, our dataset is limited to the contents of the ACM Digital Library (2011-2017) and ICSE publications (2018-2025). Although this is still a comprehensive set of papers, there are

other publishers and venues that may have a different publication culture, different standards and procedures when it comes to artifact publication, evaluation, and a generally different approach to address the topic of reproducibility.

We further reduced this limited dataset to papers related to Android, which again reduces the generalizability of our results to other areas of research.

Besides, there are various threats to the validity of our results. Foremost, our evaluation involved significant manual labor. Although supported by tools, we ultimately made the decisions about which papers to include or exclude from our evaluation. The entire assessment was a manual undertaking, involving hundreds of hours of researcher time. Our inability to get certain artifacts running could stem from our own lack of expertise, improperly configured test machines, misunderstood instructions, or other human errors. Another team of researchers may hence achieve different results with the same set of artifacts.

To address these factors, we left comments in our set of results that we provide as artifact to this paper. There we state for individual cases, e.g., why our setup attempt failed.

Another threat is the lack of blinding for papers and artifacts, requiring us to address potential conflicts of interest. Wherever possible, we delegated evaluation tasks to staff with no relations to the respective authors. Further, decisions, e.g., which papers to include or exclude, were group decisions. We also did not communicate our evaluation efforts outside our research group, and we have not reached out to any of the artifacts' authors for support. We can thus reasonably assume that we treated each artifact under comparable conditions.

7 Recommendations

Our study shows that, despite the introduction of artifact evaluation processes, reproducibility is rare. In this section, we hence provide recommendations that are supposed to increase the perceived value of artifacts, and improve their quality, availability, and visibility.

7.1 Artifacts should be archived

Out of our original set of 269 selected papers, we found that only 61% of these papers had their artifacts still available. As we explained, none of the artifacts that were supposed to be available upon request were still retrievable for us. Also, as we showed, artifacts were more likely to remain available when provided via third-party repository hosters. Moreover, there are artifacts that have been revised post-publication, which may make it harder to identify which version was used to generate the research results presented in the original publication to which they belong. For version controlled repositories, this may be possible with extra effort, for artifacts provided without version information, there is uncertainty.

All this makes clear that there is a difference between *availability* and *archiving*. In our view, properly archived artifacts are (a) preserved over time and made available by a time-stable organization, and (b) version-controlled in a way that makes it easy to see which version belongs to the paper.

If an artifact evaluation committee does not account for these aspects when awarding availability badges, the badge largely reduces to confirming the artifact's existence at the time of publication, providing less support for the community than would be desired.

7.2 Artifact evaluation should be mandatory

While artifact evaluation processes have been established at many conferences, they typically remain optional for authors [32, 39, 34]. As a result, authors may claim to provide an artifact without mandatory, independent quality control.

Ironically, this contradicts long-standing consensus on software engineering and deployment. One might argue that authors and their institutions are responsible for quality control, but how can a paper be considered *peer-reviewed* if a key part of its contribution has not undergone comparable evaluation? The many fundamental issues in the artifacts we reviewed, including syntax errors, hard-coded paths, improper directory structures, missing configuration files, and so forth reveal a lack of thoroughness in individual cases.

We thus argue that artifact evaluation should be mandatory whenever authors state they provide an artifact. Treating the paper and artifact as one unit – the full contribution – highlights the artifact’s value and may be a key step in addressing the reproducibility crisis.

7.3 Artifacts should be formally classified

It is difficult to argue about any properties of an artifact when the authors do not clearly state what kind of an artifact they intend to provide, and what its use cases are. Prior work has shown that insufficient documentation hinders reviewers and reuse of artifacts [41]. Ideally, artifacts should be provided with structured metadata that document (a) the type of artifact: measurement data (raw, processed), documentation, software (scripts, code, binaries), machine learning models (weights, checkpoints, ...), questionnaires, ..., (b) URL for download, (c) use case (reuse, reproduction, ...), (d) steps for reuse and reproduction.

Based on this information, artifacts could be evaluated with quality metrics that match their individual types. Further, structured metadata would allow automated processing and search, which could increase the visibility of artifacts in digital libraries.

7.4 Artifacts should be self-contained

In several cases, we found artifacts that were not functional, or not usable for reproduction, because they depend on code or data provided by a third party, which is no longer available. In other words, the artifacts were not self-contained, and their usefulness was reduced due to factors outside the control of their authors. We acknowledge that there are cases where this is unavoidable, however, in other cases, such a dependency on external code or data may only be introduced due to a lack of awareness by the original authors. We hence encourage authors to critically evaluate whether their artifact can be made self-contained, and also propose to make this a quality check during artifact evaluation.

7.5 Artifacts should not rely on commercial solutions

Several of the 118 artifacts that we attempted to setup for reproduction required a commercial license for a third-party service. Clearly, such a requirement limits the reusability of an artifact and is at odds with the idea of *open research*. While we acknowledge that there are cases where dependence on a commercial third-party service may be unavoidable, we, as a community, should make sure that this only occurs in cases where there is good justification.

One example that we observe more frequently over time is the use of commercial providers of large language models. We encourage authors to evaluate open-source alternatives and provide the results as part of their studies in cases where they decide to use commercial products instead. Artifact evaluators should pay attention to this matter and request justification in cases where the use of commercial services is not properly motivated.

This recommendation is supposed to support the ideal of open research, making research output more accessible, and back up the open-source community.

7.6 Reproducibility should be a fully automated process

We spent a significant amount of manual effort into our evaluation of artifacts, and a large share of this effort went into tasks one might consider *overhead*: getting familiar with the contents and structure of the artifact, configuring our test environment, retrieving code and data dependencies, sanity and compatibility checks, fixing issues, and so forth. In a narrower sense, none of this is dealing with the scientific contribution of the original authors, but rather boilerplate tasks that need to be done as a prerequisite for evaluation and reuse.

We acknowledge that overhead can never be fully eliminated and that achievable reductions depend on the specific case. However, our experience was that in too many cases, the effort was unnecessarily high. We argue that artifacts should aim for maximal automation, ideally as one-click solutions that automatically execute all reproduction steps.

Since the beginning of the timeframe we considered, in 2011, technology has advanced, rendering this proposal more realistic today than it was almost 15 years ago. Now it is possible for authors to provide pre-populated containers or virtual machines that are ready for use, thus sparing other researchers and evaluation committees from additional efforts.

In fact, properly automated artifacts would potentially be suited for fully automated evaluation by an AI-based evaluation system, a task that currently fails due to various practical limitations [20].

7.7 Establish uniform standards

The fact that ACM has developed a unified framework of badges is an important step for artifact evaluation processes in the research community. However, this framework leaves fundamental aspects of artifact evaluation underspecified, which has the advantage that conference organizers can make individually appropriate decisions, while still being in line with the framework. As stated by ACM, they also consider it “too early to establish more specific guidelines for artifact and replicability review” [12] due to the lack of established consensus regarding artifact submission and evaluation. While we generally agree with this point of view, we think it is possible to take a more differentiated standpoint today than in 2020 when ACM made this last update. For example, this study presents indication that artifacts are more likely to remain available over time if they are deployed via suitable third-party hosters. Therefore, this should be a mandatory requirement across all venues.

Another rule should be that the artifact evaluation committees review artifacts under the same conditions that would apply to any other independent researcher. Specifically, the current badge framework explicitly allows that evaluators interact with artifact authors during evaluation. However, this is often not possible for researchers who, potentially years after the original publication, want to reuse the artifact. The more different the conditions are between evaluators at conferences and researchers in the field, the less meaningful is the badge awarded by the committee. These differences may explain why some of our experimental results were not in line with badges that had been awarded to the respective papers in our dataset. We thus argue that also this should be restricted by the badge framework.

7.8 Award comprehensive certificates

Currently, papers that pass artifact evaluation receive one or more badges, represented as icons on the title page of the publication. Unfortunately, due to an underspecified badge framework and the lack of standards, it is hard to judge post-publication as an outsider what the requirements and conditions of artifact evaluation were for any given paper. The original call for artifacts may not be available anymore several years after publication, and even if it

is, it may be sufficiently underspecified as well. In our dataset, we even encountered two papers that had badge icons on their title pages, but due to their visual representation, we were unable to tell with certainty what kind of badges they were.

One way to address this would be to award comprehensive certificates, rather than badges that are only indicated by icons on the title page. These certificates could contain detailed information about the evaluation process, requirements for submission, and which quality metrics were applied. They could also more elaborately state what the outcome of the evaluation was. Such a certificate could, e.g., be attached as a final page to the paper to which it belongs. This general idea can also be found in the current ACM badge framework: “[...] badges included in PDFs and in ACM Digital Library metadata should be linked to a brief explanation of the particular review process which led to the awarding of the badge” [12].

7.9 Highlight outstanding work

It is reasonable to assume that positive visibility may increase researchers’ motivation to submit high-quality artifacts and participate in evaluation processes. While, over the past years, conference venues have applied changes that certainly improve visibility of artifacts, which includes the entire system of artifact evaluation processes and badges, there may be more that can be done to highlight the importance of artifacts. Ideas may be:

- dedicated artifact tracks with (live) presentations
- hands-on demonstration areas outside of rooms where sessions are held
- public advertising of artifacts, e.g., via social media accounts
- awards for “best artifact” and “best artifact evaluator”
- retrospective evaluations of artifacts and awards for longevity

This is not a comprehensive list, but it should illustrate that the space of opportunities is not yet fully exhausted today when it comes to maximizing the visibility of artifacts. Since this recommendation is not directly based on our data, we hope that future empirical studies will provide more insight into how such efforts may affect artifacts’ quality and availability.

7.10 Incentivize reproduction and replication work

Although the ACM badge framework includes badges for reproduction and replication, none of the papers in our dataset carried them. Also, outside the context of our study, we do not find it a common practice in computer science research to publish confirmatory research. This is striking when one considers that the majority of researchers assume that we have a reproducibility crisis [3]. We thus argue that it is about time to prominently incentivize confirmatory research that reproduces or replicates previously published results. While having dedicated venues would be an option, as well as dedicated tracks at already established venues, it would already be a starting point to explicitly invite such work in future calls for papers. Examples for ongoing efforts include the dedicated track on Reproducibility and Negative Results at EASE 2026 [29], or the track on Replication and Negative Results at ICSME 2026 [18].

8 Conclusion

Prototype implementations underpin many computer science papers by demonstrating novel concepts or generating evaluation data. An increasing number of venues encourage authors to publish code and datasets as supplementary material, and further established artifact evaluation processes. However, prior work suggests that many artifacts may not be suitable for reuse and reproduction, exacerbating an already existing reproducibility crisis.

To shed more light on this, we studied 269 Android-related papers and their artifacts. We found only 163 of these artifacts to be available, and successfully reproduced the expected results for only 8 of them, including partial reproductions. In summary, our study shows that reproducibility remains rare: many papers cannot be independently evaluated, and only a small subset of artifacts are available, functional, and complete. Based on our experience, we provide recommendations for authors and evaluation committees that aim to strengthen the perceived value of artifacts, and to improve their availability, quality, and visibility.

Data Availability Statement

This study was conducted using both non-public proprietary data and data that is publicly available yet copyrighted. We used metadata about scientific publications, the publications themselves and also data and software artifacts that were referenced in the publications but hosted on third-party websites and repositories.

Metadata and paper contents up until 2017 were obtained from a proprietary dataset from the ACM, which we were allowed to use but not republish in its raw form. Metadata and paper contents for later papers were collected manually from the proceedings of ICSE and downloaded from the websites of the respective publishers via our institutional accounts.

We used the papers' abstracts and full text to classify whether they might be related to Android research. While we cannot share these contents themselves due to copyright, we make available references – including DOIs or URLs – to all included papers in our artifact. Our methodology in Section 3 describes how our work could be replicated using other datasets.

Each of the included papers contains one or more references to research artifacts that we evaluated in our study. The investigated software and data artifacts were published by the authors on a variety of platforms, often lacking license information, deeming them unsuitable for inclusion in our artifact. However, in our artifact, we provide the list of URLs that we used to obtain the respective artifacts.

We make the full set of results of our evaluation efforts available both in PDF and CSV file formats at <https://doi.org/10.6084/m9.figshare.33256998>.

References

- 1 Steven Arzt, Stephan Huber, Siegfried Rasthofer, and Eric Bodden. Denial-of-app attack: Inhibiting the installation of android apps on stock phones. In *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices*, SPSM '14, pages 21–26, New York, NY, USA, 2014. Association for Computing Machinery. doi:10.1145/2666620.2666621.
- 2 ATVA. ATVA 2025: Artifact evaluation, 2025. Last visited 17. July 2025. URL: <https://atva-conference.org/2024/artifact-evaluation/>.
- 3 Ivanildo Azevedo, Ana Paula Vasconcelos, Eudis Teixeira, and Sergio Soares. Survey-based insights into the replication crisis and the 3r in software engineering.
- 4 Monya Baker. 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604):452–454, May 2016. doi:10.1038/533452a.
- 5 Bruce R. Childers and Panos K. Chrysanthis. Artifact evaluation: Is it a real incentive? In *2017 IEEE 13th International Conference on e-Science (e-Science)*, pages 488–489, 2017. doi:10.1109/eScience.2017.79.
- 6 Christian Collberg, Todd Proebsting, and Alex M Warren. Repeatability and benefaction in computer systems research. *University of Arizona TR*, 14(4):1–68, 2015.

- 7 Christian Collberg and Todd A Proebsting. Repeatability in computer systems research. *Communications of the ACM*, 59(3):62–69, 2016. doi:10.1145/2812803.
- 8 CORE. Core2023: International conference on software engineering, 2023. Last visited 18. July 2025. URL: <https://portal.core.edu.au/conf-ranks/1209/>.
- 9 CORE. Core2023 software engineering, 2023. Last visited 18. July 2025. URL: <https://portal.core.edu.au/conf-ranks/?search=4612&by=all&source=CORE2023&sort=arank>.
- 10 Nadia Daoudi, Kevin Allix, Tegawendé F. Bissyandé, and Jacques Klein. Lessons learnt on reproducibility in machine learning based android malware detection. *Empirical Software Engineering*, 26(4):74, May 2021. doi:10.1007/s10664-021-09955-7.
- 11 dblp. dblp: Publications per year, 2026. Last visited 10. May 2026. URL: <https://dblp.org/statistics/publicationsperyear.html>.
- 12 Association for Computing Machinery. Artifact review and badging - current, 2020. Last visited 9. July 2025. URL: <https://www.acm.org/publications/policies/artifact-review-and-badging-current>.
- 13 Eitan Frachtenberg. Research artifacts and citations in computer systems papers. *PeerJ Computer Science*, 8:e887, 2022. doi:10.7717/PEERJ-CS.887.
- 14 Google. Android developer documentation - build.version_codes, 2025. Last visited 16. July 2025. URL: https://developer.android.com/reference/android/os/Build.VERSION_CODES.
- 15 Cathy Guevara-Vega, Beatriz Bernárdez, Margarita Cruz, Amador Durán, Antonio Ruiz-Cortés, and Martin Solari. Research artifacts for human-oriented experiments in software engineering: An acm badges-driven structure proposal. *Journal of Systems and Software*, 218:112187, 2024. doi:10.1016/j.jss.2024.112187.
- 16 Ben Hermann, Stefan Winter, and Janet Siegmund. Community expectations for research artifacts and evaluation processes. In *Proceedings of the 28th ACM joint meeting on european software engineering conference and symposium on the foundations of software engineering*, pages 469–480, 2020. doi:10.1145/3368089.3409767.
- 17 Robert Heumüller, Sebastian Nielebock, Jacob Krüger, and Frank Ortmeier. Publish or perish, but do not forget your software artifacts. *Empirical Software Engineering*, 25(6):4585–4616, 2020. doi:10.1007/S10664-020-09851-6.
- 18 ICSME. ICSME 2026 replication and negative results - call for papers, 2026. Last visited 14. August 2026. URL: <https://conf.researchr.org/track/icsme-2026/icsme-2026-replication-and-negative-results?#Call-for-Papers>.
- 19 ISCA. ISCA 2025: Artifact evaluation, 2025. Last visited 17. July 2025. URL: <https://www.iscaconf.org/isca2025/submit/ArtifactEvaluation.php>.
- 20 Jenny T Liang, Carmen Badea, Christian Bird, Robert DeLine, Denae Ford, Nicole Forsgren, and Thomas Zimmermann. Can gpt-4 replicate empirical software engineering research? *Proceedings of the ACM on Software Engineering*, 1(FSE):1330–1353, 2024. doi:10.1145/3660767.
- 21 Mario Linares-Vásquez, Gabriele Bavota, Carlos Eduardo Bernal Cárdenas, Rocco Oliveto, Massimiliano Di Penta, and Denys Poshyvanyk. Optimizing energy consumption of guis in android apps: A multi-objective approach. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, pages 143–154, 2015. doi:10.1145/2786805.2786847.
- 22 Mugeng Liu, Xiaolong Huang, Wei He, Yibing Xie, Jie M. Zhang, Xiang Jing, Zhenpeng Chen, and Yun Ma. Research artifacts in software engineering publications: Status and trends. *Journal of Systems and Software*, 213:112032, 2024. doi:10.1016/j.jss.2024.112032.
- 23 Jean-Marie Mineau and Jean-François Lalande. Evaluating the Reusability of Android Static Analysis Tools. In *LNCS*, volume LNCS 14614 of *LNCS*, pages 153–170, Limassol, Cyprus, June 2024. Springer. doi:10.1007/978-3-031-66459-5_10.
- 24 Archan Misra and Rajesh Krishna Balan. Livelabs: Initial reflections on building a large-scale mobile behavioral experimentation testbed. *ACM SIGMOBILE Mobile Computing and Communications Review*, 17(4):47–59, 2013. doi:10.1145/2557968.2557975.

- 25 Tsuyoshi Miyakawa. No raw data, no science: another possible source of the reproducibility crisis, 2020.
- 26 Al Muttakin, Saikat Mondal, and Chanchal K Roy. The state of open science in software engineering research: A case study of icse artifacts. *arXiv preprint*, 2026. [arXiv:2601.02066](#).
- 27 National Academies of Sciences, Medicine, Policy, Global Affairs, Board on Research Data, Information, Division on Engineering, Physical Sciences, Committee on Applied, Theoretical Statistics, et al. *Reproducibility and replicability in science*. National Academies Press, 2019.
- 28 Daniel Olszewski, Allison Lu, Carson Stillman, Kevin Warren, Cole Kitroser, Alejandro Pascual, Divyajyoti Ukirde, Kevin Butler, and Patrick Traynor. "get in researchers; we're measuring reproducibility": A reproducibility study of machine learning papers in tier 1 security conferences. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 3433–3459, 2023. doi:10.1145/3576915.3623130.
- 29 International Conference on Evaluation and Assessment in Software Engineering (EASE). EASE 2026 call for papers - reproducibility and negative results track, 2026. Last visited 13. August 2026. URL: <https://conf.researchr.org/track/ease-2026/ease-2026-reproducibility-and-negative-results?#Call-for-Papers>.
- 30 International Conference on Software Engineering (ICSE). ICSE 2024: Artifact evaluation, 2024. Last visited 17. July 2025. URL: <https://conf.researchr.org/track/icse-2024/icse-2024-artifact-evaluation>.
- 31 International Conference on Software Engineering (ICSE). ICSE 2025: Artifact evaluation, 2025. Last visited 17. July 2025. URL: <https://conf.researchr.org/track/icse-2025/icse-2025-artifact-evaluation>.
- 32 International Conference on Software Engineering (ICSE). ICSE 2026: Call for papers, 2025. Last visited 18. July 2025. URL: <https://conf.researchr.org/track/icse-2026/icse-2026-research-track>.
- 33 International Conference on the Foundations of Software Engineering (FSE). FSE 2025: Call for artifacts, 2025. Last visited 17. July 2025. URL: <https://conf.researchr.org/track/fse-2025/fse-2025-artifacts>.
- 34 International Conference on the Foundations of Software Engineering (FSE). FSE 2026: Call for papers, 2025. Last visited 18. July 2025. URL: <https://conf.researchr.org/track/fse-2026/fse-2026-research-papers>.
- 35 Roger Peng. The reproducibility crisis in science: A statistical counterattack. *Significance*, 12(3):30–32, 2015.
- 36 PLDI. PLDI research artifacts - call for artifacts, 2025. Last visited 17. July 2025. URL: <https://pldi25.sigplan.org/track/pldi-2025-pldi-research-artifacts#Call-for-Artifacts>.
- 37 Hans E Plesser. Reproducibility vs. replicability: a brief history of a confused terminology. *Frontiers in neuroinformatics*, 11:76, 2018. doi:10.3389/FNINF.2017.00076.
- 38 POPL. POPL 2025: Artifact evaluation, 2025. Last visited 17. July 2025. URL: <https://popl25.sigplan.org/track/POPL-2025-artifact-evaluation#Author-Recommendations>.
- 39 POPL. POPL 2026: Call for papers, 2025. Last visited 18. July 2025. URL: <https://conf.researchr.org/track/POPL-2026/POPL-2026-popl-research-papers>.
- 40 Bradley Reaves, Jasmine Bowers, Sigmund Albert Gorski III, Olabode Anise, Rahul Bobhate, Raymond Cho, Hiranava Das, Sharique Hussain, Hamza Karachiwala, Nolen Scaife, Byron Wright, Kevin Butler, William Enck, and Patrick Traynor. *droid: Assessment and evaluation of android application analysis tools. *ACM Comput. Surv.*, 49(3), October 2016. doi:10.1145/2996358.
- 41 Christopher S. Timperley, Lauren Herckis, Claire Le Goues, and Michael Hilton. Understanding and improving artifact sharing in software engineering research. *Empirical Software Engineering*, 26(4):67, May 2021. doi:10.1007/s10664-021-09973-5.

- 42 Stefan Winter, Christopher S Timperley, Ben Hermann, Jürgen Cito, Jonathan Bell, Michael Hilton, and Dirk Beyer. A retrospective study of one decade of artifact evaluations. In *Proceedings of the 30th ACM joint European software engineering conference and symposium on the foundations of software engineering*, pages 145–156, 2022.
- 43 Yingqi Yue, Kaiwen Zhang, and Hans-Arno Jacobsen. Smart phone application for connected vehicles and smart transportation. In *Proceedings Demo & Poster Track of ACM/I-FIP/USENIX International Middleware Conference*, pages 1–2, 2013. doi:10.1145/2541614.2541627.