

Understanding and Improving Artifact Sharing in Software Engineering Research

Christopher S. Timperley · Lauren Herckis · Claire Le Goues · Michael Hilton

Accepted: 14 April 2021

Abstract In recent years, many software engineering researchers have begun to include artifacts alongside their research papers. Ideally, artifacts, including tools, benchmarks, and data, support the dissemination of ideas, provide evidence for research claims, and serve as a starting point for future research. However, in practice, artifacts suffer from a variety of issues that prevent the realization of their full potential.

To help the software engineering community realize the full potential of artifacts, we seek to understand the challenges involved in the creation, sharing, and use of artifacts. To that end, we perform a mixed-methods study including a survey of artifacts in software engineering publications, and an online survey of 153 software engineering researchers. By analyzing the perspectives of artifact creators, users, and reviewers, we identify several high-level challenges that affect the quality of artifacts including mismatched expectations between these groups, and a lack of sufficient reward for both creators and reviewers. Using Diffusion of Innovations (DoI) as an analytical framework, we examine how these challenges relate to one another, and build an understanding of the factors that affect the sharing and success of artifacts. Finally, we make recommendations to improve the quality of artifacts based on our results and existing best practices.

Keywords replication · artifacts · reproducibility · implementation science · replicability · diffusion

Declarations

Funding: Not applicable.

C. S. Timperley, M. Hilton, and C. Le Goues,
School of Computer Science, Carnegie Mellon University, USA
E-mail: {ctimperley,mhilton}@cmu.edu, clegoues@cs.cmu.edu

L. Herckis
Dietrich College of Humanities & Social Sciences, Carnegie Mellon University, USA
E-mail: lrhercki@andrew.cmu.edu

Corresponding author: C. S. Timperley (ctimperley@cmu.edu)

Conflicts of interest: Not applicable.

Availability of data and material: A replication package for this study, including the author survey instrument and results of the publication survey, is available at <https://github.com/ChrisTimperley/se-artifact-sharing>.

Code availability: Not applicable.

Compliance with Ethical Standards

Our study included an online survey of software engineering researchers. We received approval for our survey from Carnegie Mellon University’s Institutional Review Board. We have included our informed consent form as part of the additional materials for this study.

1 Introduction

Artifacts, in the form of tools, benchmarks, data, and more, play an integral role in software engineering research. Tools provide tangible, concrete implementations of abstract concepts and ideas that can be shared, studied, and tested. Benchmarks are the means by which we evaluate and compare the implementations of our abstract concepts and ideas, and serve as a yardstick for measuring progress in a field. Datasets and scripts are used to conduct experiments, test hypotheses, and uncover new insights. Ultimately, all of the claims that we make are with respect to these artifacts. Since ideas and competing thoughts cannot be tested quantitatively, we have to test our hypotheses on concrete implementations of those abstract ideas, however imperfect.

Artifacts provide rich benefits to the research community. They allow independent replication experiments to be performed, enrich the technical understanding of an associated research paper, and allow others to repurpose, reuse, and extend previous work. However, for an artifact to be useable by other researchers, it should be complete, structured, and well documented. This can involve significant effort on the part of authors. Unfortunately, recent work suggests that many artifacts suffer from a variety of issues (e.g., lacking documentation and unstated dependencies) that prevent them from being reused, extended, and replicated by others (Collberg et al., 2015; Collberg and Proebsting, 2016). The controversy (Krishnamurthi, 2013b) around this work suggests that there are no clear standards within the community as to how to create and evaluate artifacts.

In recognition of the importance of high-quality artifacts, several software engineering venues have introduced a formal artifact review and badging process that authors may optionally use. These processes allow the claims of artifacts to be assessed, uncover potential usability issues that may be experienced by others, and provide a signal about the quality of an artifact to the community in the form of a badge. However, the implementation of these processes has been met with both praise and criticism from members of the community (Beller, 2020; Krishnamurthi, 2013a). Given the importance of artifacts to scientific progress within software engineering research, it is vital that artifacts are shared and valued by the community, and that researchers are able to unlock the full potential of artifacts.

In this paper, we report the results of a mixed-methods study to better understand how the community perceives, creates, uses, shares, and reviews artifacts, and the challenges that impede the sharing of high-quality artifacts. We perform a statistical analysis of recent publications in software engineering venues, and conduct an online survey of 153 authors of accepted papers at those venues, including both qualitative and quantitative components. Using Diffusion of Innovations (DoI) (Rogers, 2010) as a framework, we perform a secondary analysis of these methodological components, identify and explore subtle relationships between findings, and provide a basis for making recommendations. Finally, we use principles from Implementation Science (IS) to provide actionable recommendations to specific subpopulations based on the results of our analysis.

We find that artifacts are both valued by the community and shared widely: Almost two thirds of all research-track papers published at the *International Conference on Software Engineering* (ICSE), the *Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (FSE), the *International Conference on Automated Software Engineering* (ASE), and *Empirical Software Engineering* (EMSE) between 2014 and 2018 provide an accompanying artifact. From the results of our author survey and publication survey, we identify a number of high-level challenges that affect the creation, sharing, use, and review of artifacts. These challenges include, among others, a perception that the effort required to create and share artifacts is not worth it, a lack of community standards and guidelines around artifacts and how they should be reviewed, and the need for creators to provide ongoing maintenance.

We show that issues associated with the creation, sharing, use, and review of artifacts are a product of inadequate communication, social systems (e.g., empowerment and reward), effects of time (e.g., “bitrot”), and technical aspects of the artifacts themselves (e.g., ease of use). While the community receives numerous benefits from the sharing of artifacts, we find that the individuals that create and review those artifacts perceive that there is little reward for their efforts. Among other challenges, mismatched expectations, misaligned incentives, and poor communication between the creators, users, and reviewers of artifacts lead to suboptimal outcomes and experiences for all involved, and prevent the full potential of those artifacts from being realized. We argue that Artifact Evaluation Committees (AECs), responsible for reviewing artifacts, are well positioned to tackle many of the issues we have identified, and to elevate and assure the quality of artifacts.

The main contributions of this paper are as follows:

- We conduct a mixed-methods study to understand how artifacts are created, shared, used, and reviewed (Sect. 3).
 - In one part of our mixed-methods study, we analyze all research-track papers published at four software engineering venues between 2014 and 2018 to determine the prevalence and availability of artifacts (Sect. 3.1).
 - In parallel, we conduct a survey of 153 authors to understand the perception of artifacts and to identify the challenges that researchers face when creating, sharing, using, and reviewing artifacts (Sect. 3.2).
- We present the integrated results of our individual study components (Sect. 4).
- We use DoI as a framework to understand how the challenges of creating, sharing, using, and reviewing artifacts relate to one another in terms of com-

munication channels, social systems, time, and characteristics of the artifacts themselves (Sect. 5).

- Based on our findings, we use insights from IS to provide actionable recommendations to specific subpopulations (e.g., creators) (Sect. 6).

We provide the following as a part of our replication package: our author survey materials and quantitative results, the results of our publication survey, and the scripts used to mine publication data and generate our graphs.

<https://github.com/ChrisTimperley/se-artifact-sharing>

2 Background

In this section, we introduce the reader to software engineering research artifacts (Sect. 2.1), efforts to formally recognize artifacts (Sect. 2.2), Diffusion of Innovations (Sect. 2.3), and Implementation Science (Sect. 2.4).

2.1 Software Engineering Artifacts

For the purposes of this study, we broadly define an artifact as any external materials or information provided in conjunction with a research paper via a link. In practice, this consists of any materials developed by authors and linked to from a research paper. For example, this would include replication packages, tools/source code, companion sites, benchmarks, raw data, curated datasets, survey instruments and results, mechanised proofs, and more. The Association for Computing Machinery (ACM) similarly defines artifacts 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.” (Association for Computing Machinery, 2018).

Authors may choose to share artifacts for a variety of motivations: For example, artifacts may be shared to allow others to replicate, reproduce or build upon existing work. Artifacts may be referred to as replication packages or laboratory packages (Shull et al., 2008). Historically, the motivation behind sharing artifacts was to ensure reproducibility by providing a means of exactly repeating an experiment to obtain the same (or a similar) result for the purposes of scrutiny and validation (Shull et al., 2002, 2008; Brooks et al., 2008; Basili et al., 1999).

2.2 Recognition of Software Engineering Artifacts

Motivated by the lack of attention paid to the software, models, and specifications (i.e., artifacts) underlying much of the research within software engineering, the first AEC in the software engineering research community was established at FSE 2011 (Krishnamurthi, 2013a). The AEC was tasked with formally evaluating the associated artifacts of accepted research papers. Since that first AEC, FSE has continued to hold AECs in most editions of the conference, and, most recently, ICSE held its first AEC in the history of the conference in 2019. In both their

current and original form, artifacts are optionally evaluated following paper acceptance (i.e., authors must opt-in), and papers cannot be rejected on the basis of their artifacts.

To promote and reward the formal sharing and review of artifacts, the Association for Computing Machinery (2018) proposed a set of badges for research articles containing artifacts in ACM publications: *Artifacts Available*, *Artifacts Evaluated*, and *Results Validated*. The badging scheme provides structure to the outcome of the artifact evaluation process while allowing conferences and journals to continue to review artifacts as they best see fit. As of July 2020, both FSE and ICSE participate in the ACM’s badging scheme. Awarded badges appear on the front page of the paper itself within the proceedings, and are recorded in the metadata of the ACM’s Digital Library.

Each badge is considered independently, and a paper may be awarded all badges if it meets the appropriate criteria. While most badges are awarded based on review by the AEC, authors technically may request publishers to award them an *Artifacts Available* badge without the need for formal review (e.g., ICSE 18, where two papers have badges despite no formal artifact review process). After a submitted paper has been accepted by the conference, it may be awarded either an *Artifacts Evaluated: Functional* or an *Artifacts Evaluated: Reusable* badge depending on its level of quality and potential for reuse and repurposing. The Association for Computing Machinery (2018) measures the quality of an artifact in terms of the extent to which it is *documented*, *consistent*, *complete*, and *exercisable*, according to the following definitions:

Documented: At minimum, an inventory of artifacts is included, and sufficient description provided to enable the artifacts to be exercised.

Consistent: The artifacts are relevant to the associated paper, and contribute in some inherent way to the generation of its main results.

Complete: To the extent possible, all components relevant to the paper in question are included. (Proprietary artifacts need not be included. If they are required to exercise the package then this should be documented, along with instructions on how to obtain them. Proxies for proprietary data should be included so as to demonstrate the analysis.)

Exercisable: Included scripts and/or software used to generate the results in the associated paper can be successfully executed, and included data can be accessed and appropriately manipulated.

Crucially, the ACM leaves the interpretation of its badging policy and the implementation of an associated artifact evaluation process to individual conferences and communities. Within their artifact review and badging policy (Association for Computing Machinery, 2018), the ACM states, “We believe that it is still too early to establish more specific guidelines for artifact and replicability review. Indeed, there is sufficient diversity among the various communities in the computing field that this may not be desirable at all.”

In addition to badges, several conferences (e.g., ICSE) have introduced distinguished artifact awards to recognize and reward the creation and sharing of high-quality artifacts.¹

¹ <https://2020.icse-conferences.org/info/awards> [Date Accessed: 13th March 2021]

2.3 Diffusion of Innovations

DoI is a framework from the social sciences that seeks to explain how new objects, ideas, and practices spread (Rogers, 2010). A scientific understanding of how and why new ideas take hold and spread rapidly, or are briefly acknowledged and then pass into obscurity, is valuable in disciplines ranging from medicine to information technology to anthropology (e.g., Gómez et al. 2013; Johns 1993; O’Neill et al. 1998; Premkumar et al. 1994; Wright et al. 1995). In software engineering, DoI has been used to study link sharing on stack overflow (Teshima et al., 2013), how to introduce developers to new practices (Green and Hevner, 2000), how developers use mobile development platforms (Miranda et al., 2014), and even to analyze if developers discover new tools on the toilet (Murphy-Hill et al., 2019).

In this paper we use DoI as an analytical framework to integrate findings drawn from diverse components of our mixed-methods research design, explain relationships between them, and develop a more complete understanding of artifacts in software engineering research.

There are four central elements of DoI that we apply in this paper. Below we briefly describe the elements and their meaning in the context of this paper.

- An **innovation** is any novel thing, idea, procedure, or system. It need only be perceived as new by the individual or organization that might adopt it, and may only have one aspect that is novel. In this paper, we consider individual artifacts as innovations.
- **Communication channels** are the various ways that innovations are distributed from a person of origin to a recipient. In this paper, we consider artifact links, AECs, and artifact badges as communication channels between artifact creators, potential users, and reviewers.
- **Time** plays a significant role in the diffusion of innovations. Innovations are not adopted instantly, but instead they spread and must remain relevant over time. In this paper, we consider the factors that affect the availability and usability of artifacts over time.
- All innovation, and all diffusion of innovation, takes place in the context of a **social system**. A social system is a set of interrelated units that are engaged in joint problem solving to accomplish a common goal. In this paper, we consider the social system to be the software engineering research community and its evolving set of norms, practices, members, and values, working towards the common goal of furthering research.

2.4 Implementation Science

IS is an empirical approach to understand the factors that effectively advance research and move research to practice (Bauer et al., 2015). Functionally, IS is a series of iterative processes in which effective practices and policies are identified, trialed, observed, and improved upon by researchers over the course of many studies.

IS is a relatively new discipline, developed to promote the rapid transformation of medical research into more effective medical practice and especially to coordinate research and impact practice (Zerhouni, 2003). In these diverse fields

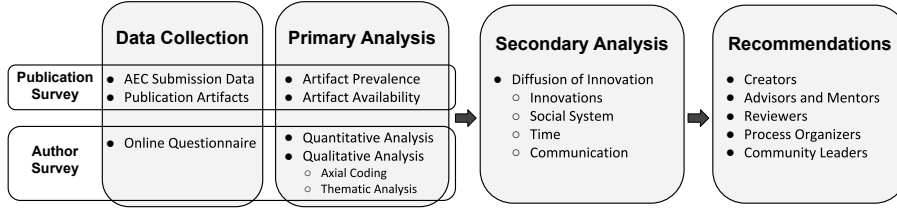


Fig. 1: An overview of our study methodology.

such as public health (Glasgow et al., 1999), business (Frambach and Schillewaert, 2002), and education (Herckis, 2018), researchers and practitioners have an interest in making results of research widely available, enabling others to build on these results, and making a rapid impact on practice. The principles of iterative, system-oriented, evidence-based improvements embedded in IS have been demonstrated to facilitate the diffusion of innovations.

3 Methodology

In this study, we set out to obtain insights into the values, norms, and practices of the software engineering research community. To determine the reasons and extent to which artifacts are created, shared, and used by researchers, we ask:

RQ1 *Does the software engineering research community perceive inherent value in artifacts?*

To understand the challenges that prevent the community from realizing the full potential of artifacts, we ask:

RQ2 *What are the challenges of creating, sharing, using, and reviewing artifacts?*

We split RQ2 into the following subquestions to understand the challenges associated with artifacts from the perspective of users, creators, and reviewers.

RQ2.1 *What challenges are faced by artifact creators?*

RQ2.2 *What challenges are faced by artifact users?*

RQ2.3 *What challenges are faced by artifact reviewers?*

To answer these questions, we used a mixed-methods approach, outlined in Figure 1, consisting of a survey of authors (Sect. 3.2) and a statistical analysis of publication, submission, and evaluation data related to software engineering artifacts (Sect. 3.1). We conduct a secondary analysis using DoI as an analytical framework to identify relationships between the challenges faced by creators, users, and reviewers within the context of the software engineering research community (Sect. 3.3). Finally, we use IS to provide recommendations, targeted at specific subpopulations, based on our analysis and existing best practices from the literature (Sect. 3.4).

3.1 Publication Survey

To better understand the prevalence of artifacts within software engineering research papers, we studied all technical track papers published between 2014 and 2018, inclusive, at three top software engineering conferences (ICSE, FSE, ASE) and one journal (EMSE).

We first used the DBLP archive² to obtain a list of all technical track papers published between 2014 and 2018 (inclusive) at each of these venues. We then downloaded a PDF for each paper and used PDFx (Hager, 2016) to transform that PDF into plaintext, before using regular expressions to find all possible URLs within each paper. After finding a list of possible URLs in each paper, we manually examined each URL to determine if it corresponded to an artifact; in the case where a paper had no URLs corresponding to artifacts, we manually inspected the paper to ensure that an artifact URL had not been missed. In total, we identified 899 artifacts across 1434 papers. Finally, we determined the availability of each artifact by manually checking that its associated URL could be accessed at the time of inspection (between January 29th and February 6th, 2019).

To determine how many papers containing artifacts were submitted to an AEC for review, as well as the associated acceptance rate of submissions, we contacted the AEC organizers for all conference years within our dataset that had an AEC (i.e., FSE 2015–2018) via email.

3.2 Author Survey

We designed and distributed an online questionnaire to members of the software engineering research community to identify the perceived value and challenges of artifact creation, sharing, use, and review. We used a survey containing a total of 28 questions, shown in Table 1, to probe the intersecting subpopulations of artifact creators, users, and reviewers. The survey included both selection-based and open-ended questions. Branching logic was used to identify the subpopulations to which respondents belonged and show questions that were relevant to those subpopulations. For example, respondents who had previously served on an AEC, or who were currently serving on an AEC, were asked questions about that experience; respondents who had not served on an AEC were not asked those questions.

Recruitment To obtain an appropriate sample of the authors in our publication dataset, we identified the subset of authors who had authored at least one technical track publication at ICSE, FSE, ASE, or EMSE in 2018 — the most recent complete year at the time the survey was conducted. We chose this approach because we wanted to ensure that survey respondents would be able to reflect on recent experiences. Collecting email addresses for each of those authors was a time-consuming manual process: We first consulted the contact information in the paper, where available, before using a search engine. Survey respondents were not paid for taking the survey.

In total, we obtained email addresses for 744 authors, to whom we subsequently sent the survey. 46 of the 744 survey emails that we distributed failed to deliver.

² <https://dblp.uni-trier.de> [Date Accessed: March 13th 2021]

#	Survey Question	RQs
Q1	Which would you say best describes your primary role?	
Q2	Roughly how many publications have you authored?	
Q3	Have you ever published a paper that included an artifact?	2.1
Q4	Why have you found that publishing artifacts was not the right choice?	2.1
Q5	Consider your most recently published paper in which you shared an artifact. Select all the reasons below for which you shared that artifact.	1
Q6	Consider up to your three most recent papers published in a software engineering venue. Was there an artifact shared with any of these papers?	2.1
Q7	Consider up to your three most recently published papers. State all of the reasons below for which you chose not to (or were unable to) create or share an artifact associated with those papers.	2.1
Q8	Consider up to your three most recently published papers. Did you experience any difficulties or challenges when creating artifacts associated with those papers?	2.1
Q9	Have you ever submitted to an artifact evaluation committee?	2.1
Q10	For which of the following reasons have you used an artifact created by someone else?	1
Q11	What challenges have you faced when trying to use an artifact other than your own?	2.2
Q12	What are the most important things that an artifact should have, be, or do?	1
Q13	Have you ever evaluated artifacts as a member of an Artifact Evaluation Committee?	2.3
Q14	List up to three problems or challenges with the artifact evaluation process that you are aware of as a former member of an AEC.	2.3
Q15	List up to three ways that the artifact evaluation process could be improved.	2.3
Q16	What other positive or negative experiences with artifacts do you think we should know about?	1, 2.1, 2.2, 2.3

Table 1: A list of our survey questions, mapped to their relevant research questions.

Of the remaining 698 emails that were delivered, 153 recipients completed the survey, producing a very strong 22% response rate that exceeded our expectations. Demographic information for our participants is provided in Table 2.

Qualitative Analysis We analyzed the qualitative components of our survey responses using a descriptive coding approach (Saldaña, 2015), in which responses related to each segment of data is assigned basic labels to create an inventory of codes. This process was undertaken collaboratively by domain experts (Timperley, Hilton, Le Goues) and a methodologist (Herckis), after which adjudication and code mapping were used to refine codes and collapse categories. Finally, we used an axial coding approach to strategically organize data and determine which

Role	#	%	Subpopulation	#	%
Academic Researcher	139	91	Creators	137	90
Industrial Practitioner	6	4	Users	129	84
Industrial Researcher	8	5	Reviewers	32	21

Table 2: The self-reported role and subpopulations of our 153 survey participants.

themes were dominant and which less important, as well as to identify themes that offer opportunities for policy, process, or practice improvement (Charmaz, 2014).

Note that the goal of thematic analysis is to identify the full range of *themes* that characterize some class of experiences. This process entails continued qualitative data gathering and analysis until analysts have reached thematic saturation, when no new properties, dimensions, conditions, or consequences can be identified in the data. A thematic analysis does not tell us how prevalent each of the experiences represented in the exhaustive inventory of themes might be, only that they are present in the social context (Saldaña, 2015).

3.3 Secondary Analysis

These two parallel studies, the survey of publications and survey of authors, resulted in a set of primary results that situate artifacts in the broader context of the software engineering community. We performed a secondary analysis to contextualize and integrate results of these component studies by applying the DoI framework as an analytical tool (Creswell and Clark, 2017). We consider the results of our primary analyses in the context of the DoI framework allowed us to examine relationships between various findings. This analysis results in a robust, descriptive picture of the landscape of norms and practices, which uniquely positions this type of research to inform policy recommendations and the creation of evidence-based guidelines.

3.4 Recommendations

Based on the challenges identified in our primary analysis, and a broader understanding of the context in which artifacts are created, used, shared, and reviewed, we make recommendations that address specific challenges and adhere to IS-based principles. We assess each recommendation to ensure that it does not exacerbate any known challenges, is compatible with existing practices, and is likely to be scalable, sustainable, triable, and observable (Rogers, 2010). Some of our recommendations are supported by existing literature, while others are novel and arose directly from the present research. Each of these recommendations can be implemented, trialed, observed, and evaluated to determine whether it has achieved its intended outcome.

Recommendations are most effective when they target specific subgroups within a social system (Wolfe, 1994). To that end, we tailor our recommendations to the following subgroups:

Creators: Includes both *primary researchers*, typically students and postdocs, who are predominantly responsible for creating and maintaining artifacts, and the *mentors and advisors* of those primary researchers.

Primary Researcher: Primary researchers are extensively involved in most aspects of research projects (e.g., writing code, collecting data, running experiments), including the creation and maintenance of artifacts. This role often, but not always, falls to students and postdocs. As those working most closely with artifacts, primary researchers can take several steps to elevate the quality of their artifacts, reduce unnecessary work for both themselves and their potential users, and ensure that they are credited for their work.

Research Mentors and Advisors: Traditionally, this role belongs to the advisor of the primary researcher, but there may be multiple mentors on a particular research project. Advisors and mentors are typically less involved in the technical aspects of the research, including the creation and maintenance of artifacts, but are well positioned to promote and support primary researchers in their artifact-related efforts.

Reviewers: Potential artifact reviewers include Artifact Evaluation Committee members, journal paper reviewers, and technical program committee members who may have access to the artifact.

Process Organizers: Individuals in this role are responsible for the design, organization, and implementation of artifact review, and include AEC chairs, conference chairs, and journal editors. These individuals have the ability to improve the effectiveness and outcomes of the review process, and, by extension, raise the standard of artifacts within the community.

Community Leaders: Community leaders include steering committees, journal editors, professional organizations, reappointment and promotion committees, hiring committees, and funding agencies. These entities collectively hold significant influence over the research community and, as a result, interventions at this level are capable of achieving wide-reaching, systemic change.

3.5 Threats to Validity

As with all work, it is possible that we could have inadvertently biased our results due to our methodology. We examine our threats to validity, and organize them into the following areas: replicability, construct, internal, and external.

Replicability *Can others replicate our results?* Qualitative studies, in general, can be difficult to replicate. We have made as much of our materials available as possible, while still preserving the confidentiality of the survey respondents. With our accompanying artifact, we have published our publication survey dataset, the design materials (i.e., questionnaire, consent forms, recruitment email) and deidentified quantitative responses for our author survey, and a Jupyter (Kluyver et al., 2016) notebook for reproducing the figures presented in this paper.

Construct *Are we asking the right questions?* One potential threat to this work is how we measured artifacts. We consider a paper that has a link to accompanying work to have an artifact. However, it could be the case that papers could have artifacts that are not documented in the paper, which could potentially be found only by searching, or contacting the authors of the paper. We did not contact

authors for practical reasons, but we acknowledge that there could exist artifacts that were not included in our analysis.

As in every study, the construction of the study can impact the results. To achieve the best results possible, we used a mixed-methods parallel-convergent design incorporating three components: (1) analysis of quantitative data describing artifact publication and evaluation behavior; (2) analysis of survey data describing authors' perceptions, needs, and experiences; and (3) a subsequent integrated analysis. This design was selected in order to obtain different but complementary data on the subject, and to synthesise results for a more complete understanding of artifact sharing (Morse, 1991).

Internal *Did we skew the accuracy of our results with how we collected and analyzed information?* Surveys can be affected by intentional and unintentional bias, both from the survey respondents and from the researchers. To mitigate this concern, we asked respondents to report their own experiences, feelings, values, and needs. We analyzed the qualitative components of our study collaboratively using a concept-orientated approach. We carefully documented our process throughout the phases of data collection and analysis. To ensure the validity of our study, we used a purposeful sampling approach, triangulation of researchers, and triangulation of analyses (Kitto et al., 2008).

The reliability of qualitative and mixed-methods research is determined by the rigor and consistency with which the methodological processes are applied. We collected data through multiple sources and interpreted our results by using multiple conceptual frames (Kitto et al., 2008). We enhanced the reliability of our process and results through constant comparison, comprehensive data use, and use of tables, as proposed by Silverman and Marvasti (2008).

In this work, we define an artifact as being available if the link is still alive. It could possibly be the case that what is there is completely unusable, or perhaps the artifact page links to a download that is no longer available. However, evaluating the quality of artifacts is beyond the scope of this work, and is something that we leave to other researchers.

External *Do our results generalize?* Because of the nature of surveys, we cannot generalize our results beyond our survey population. Perhaps if we had chosen a different population, we would have had different answers. To mitigate this, we sampled authors from several of the top software engineering conferences (ICSE, FSE, ASE), as well as a leading software engineering journal (EMSE).

Our study was designed to describe the values, experiences, and needs of the software engineering research community: It does not return results which can be generalized beyond this particular domain.

4 Results

In this section, we present the results of our primary analysis, organized by each research question. Section 4.1 examines how the software engineering research community perceives and values artifacts, and the reasons for which artifacts are used and shared. Section 4.2 subsequently identifies the challenges that are faced by artifact creators, users, and reviewers.

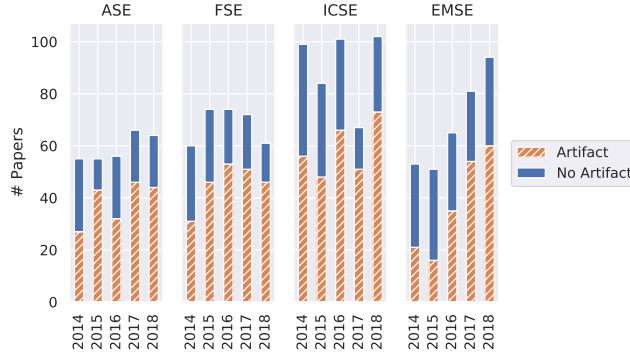


Fig. 2: A breakdown of all technical papers published at ASE, FSE, ICSE, and EMSE, showing the fraction of papers that do contain an artifact (orange, dashed) vs. those that do not (blue), between 2014 and 2018, inclusive.

4.1 RQ1: Does the software engineering research community perceive inherent value in artifacts?

From the results of our publication survey, illustrated in Figure 2, we find that almost two thirds (62.69%) of papers published at ASE, EMSE, FSE, and ICSE between 2014 and 2018 (inclusive) contain an artifact. Furthermore, across all venues that we studied, the proportion of papers containing an artifact grew from 50.56% in 2014 to 69.47% in 2018, indicating an increasing prevalence of artifacts.

When we asked survey respondents whether they value artifacts and what they think artifacts should have, do, or be, it became clear that the software engineering research community does indeed perceive inherent value in artifacts. While participants recognize that not every paper needs an artifact, numerous participants expressed positive sentiment around the sharing of artifacts, and believe that sharing should be encouraged and rewarded by the community. P89 expresses that they are “[...] a big fan of sharing!” and P20 says that “[s]haring is essential [...].” In the words of P82, “[...] artifacts are moving toward becoming a necessary and important part of publishing a research paper. It allows for meaningful analysis and evaluation of the approach. Also, replication of studies are extremely difficult (and often unreliable) without the appropriate information contained within artifacts.” Indeed, P34 highlights that “[e]mpirical research may need more data artifacts.”

To better understand the motivations for sharing artifacts, we asked participants for the reasons they had shared an artifact with their most recently published paper that contained an artifact. Table 3 provides a high-level summary of the motivations for sharing artifacts based on a thematic analysis of responses.

Title and description	Representative quote
Supporting evidence: Artifacts can be used to provide supporting evidence for the claims in a research paper, and to improve the confidence of readers and reviewers.	“I think publishing artifacts is good choice. It would increase readers’ confidence on how well proposed approaches are validated. I do not fully believe results in some papers, which even are published on top conferences such as ICSE and FSE.” – P62
Community norms: Some researchers see the creation and sharing of artifacts as a commendable pursuit that benefits the community.	“It’s good practice” – P99
Facilitate reuse: Researchers may share tools and datasets with their papers that can be reused and extended by others.	“To provide a dataset” – P130
Improving understanding: Artifacts may be used to provide additional information about a study, such as technical details that are not suitable for a paper, or raw and preprocessed results.	“The paper was about how we created a tool. That tool is now publicly available (open source)” – P136

Table 3: A high-level summary of the motivations for sharing artifacts.

Insights for RQ1
<ul style="list-style-type: none"> • An increasing majority of research papers contain artifacts • Artifacts are perceived as inherently valuable by the research community • Authors create and share artifacts to support their claims, allow others to better understand their work, and to facilitate further research • Creating and sharing of artifacts is an established community norm

4.2 RQ2: What are the challenges of creating, sharing, using, and reviewing artifacts?

Table 4 provides an overview of the high-level challenges that affect the creation, sharing, use, and review of artifacts, that emerged from our thematic analysis of survey responses. In the following sections, we discuss how these challenges affect artifact creators (Sect. 4.2.1), users (Sect. 4.2.2), and reviewers (Sect. 4.2.3).

4.2.1 RQ2.1: What challenges are faced by artifact creators?

Below, we describe the high-level challenges in Table 4 that affect artifact creators.

C1: Not worth it Although artifacts are seen as being beneficial to the community, authors perceive that creating and sharing artifacts can be difficult and time consuming, producing artifacts does little to advance one’s own career, and that, ultimately, the time spent preparing artifacts could be better spent on other

Title and description	Representative quote
<p>■ C1: Not worth it: The time required to prepare an artifact could be better spent on other high-value activities (e.g., paper writing). Additionally, the risks of sharing an artifact may outweigh the rewards of doing so.</p>	<p>“Getting the artifacts (raw data, diagrams, tools) into publishable form requires extra time (e.g. for cleaning, documentation, designing a web page), which I typically don’t have while preparing the main publication. Putting my time into improving the main publication has a clear benefit, whereas the benefit of releasing the artifacts is hard to quantify.” – P137</p>
<p>■ C2: Portability: Artifact may not build or run as intended on other machines due to missing information, poor packaging, a reliance on outdated and hard-to-obtain dependencies, and bugs in the source code.</p>	<p>“The code requires refactoring to be executed independently from the experiment environment.” – P4</p>
<p>■ C3: Maintenance: The creator of the artifact may have moved on (e.g., a graduating Ph.D. student), or is no longer interested in or able to maintain the artifact.</p>	<p>“Some [artifacts] are no longer maintained, which is understandable.” – P107</p>
<p>■ C4: Tacit knowledge: Poor documentation or a lack of documentation, coupled with incomprehensible code, make it difficult to understand how the artifact works, and how it can be extended and reused.</p>	<p>“Huge R files that are not comprehensible at all.” – P48</p>
<p>▲ C5: Artifact does not fit purpose: The artifact does not match the claims in the paper, or is difficult to use as intended.</p>	<p>“In four cases this year, I re-implemented the code and even re-collected the data, because the data was pre-processed and not raw data.” – P14</p>
<p>■ C6: Lack of standards and guidelines: Creators and reviewers are unsure of how artifacts should be packaged, and what standards those artifacts should meet.</p>	<p>“It is quite difficult to create an artifact for evaluation, particularly when the standard of acceptance is not clear.” – P61</p>
<p>■ C7: Hosting: It can be challenging to find a host for the artifact that ensures long-term archival, supports large file sizes, and is affordable or free. Many artifacts are no longer accessible.</p>	<p>“Some artifacts are just too big (like large datasets) and downloading and using them is impossible.” – P8</p>
<p>■ C8: Double-blind review: Submitting an artifact as part of a paper for double-blind review adds additional work and difficulty to the process of sharing artifacts.</p>	<p>“Sharing the artifacts was too much of a hassle due to double blinded review.” – P124</p>
<p>■ C9: External constraints: Certain artifacts are difficult to share due to reasons beyond the control of their creators.</p>	<p>“I typically am unable to include actual data for privacy/legal reasons due to the fact I work at a company.” – P16</p>
<p>★ C10: Lack of reviewer incentives: Reviewing artifacts is seen as a time-consuming activity with little reward</p>	<p>“We usually just make students do AEC. This is good in some ways because it gets students involved in the reviewing process. But it also reinforces the idea that artifacts are not a primary thing and that authors should really focus on the paper.” – P37</p>
<p>★ C11: Technical obstacles: Inherent challenges that make it difficult to evaluate artifacts with limited time and resources.</p>	<p>“It is hard to decide what to do for artifacts that by nature take more time than one can devote to evaluating a single artifact (~ one day).” – P50</p>
<p>★ C12: Limited communication: A lack of open communication between creators and reviewers makes it hard to resolve inevitable technical issues and leads to frustration for both parties.</p>	<p>“Sometimes AECs adopt a single review (as against multiple rounds of contacting authors). This can be really frustrating especially because it takes a considerable amount of time to build an artifact and a minor glitch in the installation instructions should not be grounds for rejection” – P142</p>

Table 4: The high-level challenges of artifact creation, sharing, use, and review that are experienced by creators (■), users (▲), and reviewers (★).

higher-value research activities (e.g., paper writing). That is, there is a high *opportunity cost* associated with creating and sharing artifacts. This cost is at its highest when artifacts are submitted for evaluation: As P1 puts it, “the time required to submit to an artifact track is not rewarded enough. In other words, there is not enough gain in terms of publication quality to justify it.” P23 sees the overheads of artifact review as a potential reason not to submit artifacts at all: “I’ve never been asked to do so, but it sounds like a lot of extra hassle and work. This extra cost would make me re-think whether I wanted to submit artifacts at all.”

Authors also report a variety of risks that outweigh the potential rewards of sharing artifacts. The perceived risks include that poor code quality may harm their reputation, a fear of being “scooped,” a fear of mistakes in the analysis being discovered, the burden of maintaining the artifact, and the possibility that the artifact may be used by no one. For example, when asked why they had not shared artifacts with a recent paper, P87 said that they “didn’t have enough time to publish it, [were] scared the analysis is incorrect, [and] wanted to save it for the next publication.”

Authors may be reluctant to share artifacts that still have some untapped publication value, or may wish to avoid being “scooped” by others by providing their artifacts prematurely. As P120 puts it, “sometimes, one needs to protect a Ph.D. student’s abilities to produce sufficient results prior to making artifacts available to others.”

P61 chose not to share an artifact with a recent paper due to the belief that there is a limited audience: “There is only a small group of people working on the topic of the paper. Perhaps [no]body other [than] myself will keep working on the same topic and use the artifact.” In contrast to this view, P68 highlights the importance of sharing artifacts, particularly for emerging areas of research: “Despite my complaining, moving towards a norm of making artifacts available is incredibly valuable. I know of whole research areas that are dead because the early work had no artifacts available, and the cost of re-implementing their work just so you can move on to something novel is too high.”

Overall, despite the perception that the creation and sharing of artifacts is a poor investment of time and effort, some researchers, such as P5, have seen personal benefit from sharing artifacts with their research:

“I had some experience where some researchers were not in favor of providing research artifacts, either because the effort/time investment was not worth it, or because it may ‘take away’ from their next paper.

On the other hand, in my own experience so far, providing these artifacts (companion website, accompanying blog post, or a replication package) seems to have been very beneficial for the papers and the dissemination of our results.”

C2: Portability Anticipating and preparing for the possible environments in which an artifact may be used is a considerable challenge, as P70 shares: “The major challenge I face is in ensuring that [the] artifact runs in other environments which is hard to verify beforehand.”

Retroactively addressing such concerns in existing artifacts may require extensive refactoring, or the creation of a virtual machine or container image (e.g., Docker [Docker Inc., 2020]). In either case, the repackaged artifact should ideally

be tested in a variety of environments. P37 shares that, “the challenge was to find a suitable way to share the artifacts so that they are accessible/runnable by everyone. Sometimes we have to try the artifacts on different environments to make sure they work. Alternatively we can provide VM or Docker images, which might have their own challenges (e.g., VM images becoming too big, etc). Also, time pressure does not allow documenting everything for running/using the artifacts, making it difficult to make sure everyone can use them eventually.”

C3: Maintenance Sharing an artifact not only requires an upfront investment of time and resources to prepare the artifact, but also an ongoing investment to continually maintain the artifact. As bugs are uncovered, use cases evolve, and inevitable bit rot occurs, there is a need to update the artifact. When asked why they had chosen not to create and share an artifact with a recent paper, P50 said that “[t]ime is the main reason; it takes a lot of time to package [the] artifact so that it can be useful to others,” before going on to point out that “if students graduate, it is hard to find somebody else to prepare the artifact and make it available online.” Indeed, as students are often those responsible for the creation, sharing, and maintenance of artifacts, it can be difficult to maintain those artifacts once those students move on.

C4: Tacit knowledge The process of writing documentation and refactoring code can be challenging and burdensome. As P35 says of creating and sharing artifacts, “the bottleneck tends to be creating the required documentation (both usage instructions and code documentation).” In the absence of concerns around the time and effort required to produce documentation, tacit knowledge remains a major obstacle to artifact creators (Shull et al., 2002). As P133 puts it, “the main difficulty is to provide a highly automated way to utilize the artifact. A lot of knowledge is typically buried in developers’ heads.” The problem of tacit knowledge is compounded by the quickly evolving, prototypical nature of most artifacts, as described by P103: “We usually develop prototype tools and usually as researchers we understand their restrictions and limitations to a certain level. For example, we have not tested them in all environments. It is very difficult to have a manual.”

The need to anticipate the various users apriori at the time of producing the artifact can be a challenge, as P50 identifies “there is always a question [of] what is the best way to package the artifact to be useful to others. Once you take a step forward, it is hard to go back and recreate everything.”

C6: Lack of standards and guidelines Creators identify a lack of clear standards and expectations around the packaging, contents, and quality of artifacts as a difficulty when deciding to share their artifacts for formal review: As P61 says, “it is quite difficult to create an artifact for evaluation, particularly when the standard of acceptance is not clear.”

In some cases, creators are unclear as to what is considered to be an artifact for the purposes of formal review, and how to proceed in cases where the work builds on top of an existing artifact. For example, P34 says that it is “unclear how to handle artifacts that use research data from public datasets / existing artifacts. Should we create a new artifact, or point towards an existing one?” Creators blame such ambiguity and unclear expectations on a lack of sufficient guidance on the contents and packaging of artifacts within the community. P104 shares:

“To me, I think one of the biggest challenges in creating artifacts that accompany papers relates to the amount of work required going from often messy research code to something that can be used with relative ease by a larger body of interested individuals. This can often be a significant amount of work, and I think the community lacks some guidelines related to what actually needs to be included with software artifacts in order for them to be as usable and reproducible as possible. For example, things like setup instructions, contribution guides, sufficient documentation, etc, are often overlooked.”

C7: Hosting To facilitate sharing, creators must find a suitable place for long-term hosting of the artifact that meets their various needs: (1) that the artifact must be reliably available indefinitely, (2) the host must be able to store large volumes of data, and (3) the hosting service must be affordable, or, ideally, without cost to researchers. As P137 states, finding such a host can be difficult: “I am unaware of (free) services that can guarantee long-term accessibility to my artifacts.” Likewise, P110 shares, “It is usually hard to find a good host for them [artifacts] that is ‘respectfu[l]’ and will host them [artifacts] for long,” and P12 says, “one challenge is to share a large dataset that does not fit into a GitHub repo or Dropbox.” Note that, while popular services such as Google Drive,³ GitHub,⁴ and Dropbox⁵ provide a free or inexpensive method of hosting artifacts, they are not intended as an long-term archive, and are ill-suited to large datasets.

C8: Double-blind review Creators report that the double-blind process adds additional difficulties to the process of sharing artifacts in two ways. Firstly, artifacts must be carefully anonymized such that the identities of their authors are not revealed. As P5 puts it, “we need to create the artifacts in such a way that it won’t violate the double-blind review process.” Secondly, the double-blind review process further complicates the difficulties of finding a suitable host for the data by adding the requirement of anonymity. P155 says, “it takes me some time to find a suitable place to anonymously provide artifacts for double-blind review. I use Google Drive most, but it does not provide an option for anonymous share.” Despite the challenges, some creators report success in finding suitable hosting services: P19 says, “before I discovered Zenodo, it was difficult to share artifacts[,] especially when the conferences had [a] double-blind policy.”

Ultimately, the additional effort needed to overcome these difficulties can dissuade creators from sharing their artifacts at the time of paper submission. For example, P124 says that “sharing the artifacts was too much of a hassle due to double blinded review.”

C9: External constraints Circumstances beyond the control of creators may prevent or complicate sharing artifacts with the general public. Such circumstances may be ethical or legal in nature, such as intellectual property restrictions and privacy concerns, as well as more technical circumstances, such as the artifact belonging to a larger ecosystem, making it difficult to share and reproduce.

³ <https://drive.google.com> [Date Accessed: March 13th 2021]

⁴ <https://github.com> [Date Accessed: March 13th 2021]

⁵ <https://dropbox.com> [Accessed: 12 Mar 2021]

Creators working in or collaborating with industry reported being unable to share the associated artifacts (e.g., code and data) of their research with the general public due to intellectual property (IP) restrictions. For example, P140 says, “in our collaboration with an industry partner[,], sharing artifacts was not allowed for contractual reasons.” Fear of liability was also given as a reason for being unable to share artifacts by more than one creator, such as P41 who shares that “the paper that we did not share artifacts for was based on data obtained from our industrial collaborator. We asked them if we could share this in anonymised form but they did not allow this, fearing that the data may still be misused (e.g. to start liability lawsuits).”

Creators also voiced their frustration at reviewers for a perceived lack of understanding of such external circumstances. P22 shares that they were “once criticised for not sharing data for an industry track paper coauthored with practitioners. Artifact should not be used as ritualistic blanket argument to judge papers.”

Privacy concerns and the need to de-identify personal data were also given as difficulties of sharing data artifacts, and, in some cases, a reason for not sharing at all. When asked about the challenges of creating artifacts, P44 cited: “de-identifying survey results and worrying whether we were thorough enough.” Indeed, beyond being an error-prone and time-consuming process, studies have shown that sharing qualitative data risks re-identification even when steps are taken to remove personal identifiers from the data (Narayanan and Shmatikov, 2008; El Emam et al., 2011; Ji et al., 2014). Given the inherent risks of sharing sensitive data, it is perhaps unsurprising that participants reported not sharing data artifacts, such as P44, who states “I don’t share qualitative data, as a rule. It seems too risky (re-identification) and would run counter to IRB conditions.”

Insights for RQ2.1

- Artifact creation and sharing is perceived as a risky and poor investment of time and effort that produces little personal reward
- Sharing certain artifacts necessitates maintenance, which is difficult to provide when students graduate and move elsewhere
- A lack of community standards and incentives makes it difficult for creators to produce high-quality artifacts that can be used by others
- Finding a hosting service that is free, capable of storing large volumes of data, and provides indefinite archival of artifacts can be challenging
- Some artifacts are dangerous, difficult, or impossible to share due to privacy and IP concerns, or belonging to a larger ecosystem

4.2.2 RQ2.2: What challenges are faced by artifact users?

Below, we describe the high-level challenges in Table 4 that affect artifact users. Quantitative results from Q11, “What challenges have you faced when trying to use an artifact other than your own?” presented in Table 5, are discussed within the context of the high-level challenges.

C2: Portability As P9 highlights, artifacts may suffer from a variety of portability issues that prevent them from being reused in other environments (e.g.,

Challenges when using artifacts produced by others		#
C4	A lack of clear instructions for using the artifact	103
C7	The artifact was no longer accessible	81
C2 C3 C4	A lack of clear instructions for building the artifact	79
C2	Couldn't build the artifact due to unavailable software, library, or compiler	69
C2	Couldn't run the artifact due to missing package, library, tool, or software	66
C2 C5	Had to modify the source code of the artifact	60
C2	Encountered a run-time error when using the artifact	55
C2 C5	Had to re-implement the artifact	28
	I have never tried and failed to use an artifact	10
	Other	10

Table 5: A summary of 130 responses to the question: “What challenges have you faced when trying to use an artifact other than your own?” annotated with their frequency and the identifiers of their associated challenges.

hardware, OS, and software differences): “Many artifacts have had hardwired (and undocumented) dependencies on particular files and folders such as the code author’s home directory. Some require the user to be ‘root’.” From our quantitative results, shown in Table 5, we observe that many artifact users have experienced portability-related challenges. 79 users complained about a lack of clear instructions for building an artifact, 69 users were unable to build an artifact due to dependencies that are no longer available (e.g., libraries, compilers, tools, etc.), and 66 users reported being unable to run an artifact due to missing dependencies. In the event where users were able to build and run an artifact, 55 users encountered a run-time error. 60 users reported that they had to modify the source code of an artifact, and, in the most extreme cases, 28 users shared that they had to reimplement an artifact entirely.

C3: Maintenance A lack of maintenance can impede or prevent users from being able to use an artifact. For example, P137 recalls a challenge in using an artifact where the “[r]equired libraries were not exactly unavailable, but hard to find. The artifact was no longer actively maintained and used old version of libraries, which become harder and harder to get to work.”

Despite the difficulties of dealing with unmaintained artifacts, users are sympathetic of the challenges that face creators. P105 acknowledges that “some [artifacts] are no longer maintained, which is understandable,” and P102 stresses that “[t]here is more of a need to recognize that more funding is needed to focus on artifact maintenance.” P20 identifies that, while maintaining artifacts is important, few researchers have the resources required to do so: “Sharing is essential, and it should be more valued by the community. In this ‘publish or perish’ system, few care is given to the paper’s artifacts as the focus is, indeed, on the next paper. IMO, only

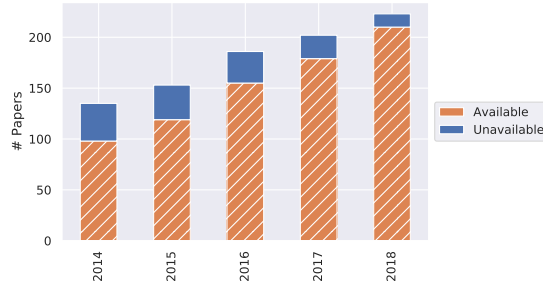


Fig. 3: An overview of the availability of all artifacts within our dataset over time, determined by manually checking whether the artifact URL could be accessed at the time of inspection (between January 29th and February 6th, 2019).

big groups, or only people who have [money] to spend for artifact maintenance, are doing a good job, which is clearly far from being the ideal situation.”

C4: Tacit knowledge A large number of users complain about difficulties in understanding, building, and using artifacts created by others due to a lack of documentation and poor code quality, as shown in Table 5. For example, P70 shares that they “often find the necessary documentation missing[,] which makes it hard to comprehend and run the artifact,” and P33 says that “some authors think that just adding a link to the raw material is sufficient. Guidance is needed to understand artifacts, and this takes time.” The magnitude of this challenge is highlighted by our quantitative results, presented in Table 5, which show that “a lack of clear instructions for using the artifact” and “a lack of clear instructions for building the artifact” are the first and third-most common challenges experienced when attempting to use an artifact produced by someone else.

C5: Artifact does not fit purpose In some cases, artifacts may be difficult for others to use for the purpose stated in their associated research papers. P63 states that “they [artifacts] don’t really serve the purpose stated in papers,” and P2 points out that “most artifacts work for very limited use cases.” Using an artifact for its intended purpose may involve extensive changes on the part of the user: 62 participants report having modified the source code of an artifact, and 28 participants had to reimplement the artifact, as shown in Table 5.

C7: Hosting When asked what challenges they had faced when attempting to use an artifact, 82 users stated that being unable to access an artifact, making it the second-most commonly reported challenge (Table 5).

From the results of our publication survey, given in Figure 3, we find that approximately 14% of all artifacts within our dataset are inaccessible via the URL given in the corresponding paper. As one may expect, older artifacts are more likely to be unavailable (26.47% in 2014), but we also observe that several more-recent artifacts (5.43% in 2018) are also unavailable.

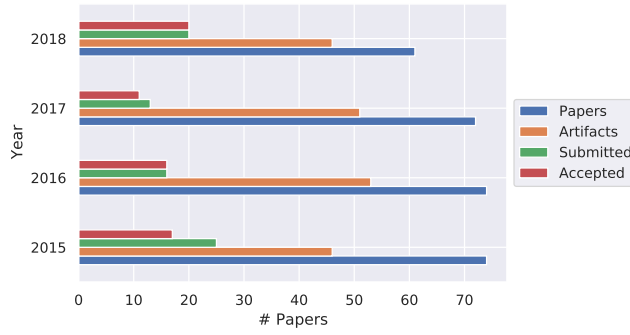


Fig. 4: A summary of the papers accepted at FSE between 2015 and 2018 that were submitted to and accepted by the AEC for the conference where **Papers** is the number of papers accepted at the conference, **Artifacts** is the subset of those papers we deem to contain an artifact, **Submitted** is the subset of papers that contain artifacts that were submitted to the AEC, and **Accepted** is the subset of papers that were accepted by the AEC. Note that neither ICSE nor ASE conducted artifact evaluation between 2015 and 2018.

C9: External constraints Licensing restrictions can create difficulties for those attempting to build upon and extend artifacts. For example, P35 shares that “to use and extend the artifact, I had to recover the source code using a decompiler. Afterwards I couldn’t make available a replication package for the extended artifact due to licensing issues.” Likewise, P9 identifies that some artifacts “require a paid commercial license [for the] auxiliary tools necessary to use [the artifact].”

Insights for RQ2.2

- Artifacts may become unavailable and unmaintained over time
- Missing documentation, non-portable code, and licensing restrictions make it difficult to reuse and extend artifacts
- Users often need to modify artifacts to fit their needs and expectations
- These challenges are exacerbated by the lack of community standards for packaging and sharing artifacts

4.2.3 RQ2.3: What challenges are faced by artifact reviewers?

As the predominant mechanism for assessing and upholding the quality and claims of software artifacts, AECs are well positioned to mitigate the downstream issues faced by those using artifacts. However, despite our observation that almost two thirds of research papers have an associated artifact (Sect. 4.1), we find that relatively few papers with artifacts are submitted for evaluation when it is possible to do so. Of the 281 research papers accepted at conferences that had an AEC (i.e., FSE 2015–2018), of which we deem 196 to contain artifacts via our analysis (69.75%), 74 papers were submitted to the AEC (26.33%), and 64 were subsequently accepted, representing a markedly high acceptance rate of 86.49%, shown in Figure 4.

Below, we explore the low participation rate for artifact evaluation by describing the high-level challenges that affect artifact reviewing. We find that reviewers face many of the same challenges that are encountered when using artifacts (e.g., problems stemming from a lack of portability and documentation). To our surprise, we find that reviewers also face challenges similar to those faced by artifact creators (e.g., inadequate guidelines and incentives). It should be noted that some reviewers reported being happy with the current form of AECs: When asked how the artifact evaluation process could be improved, P157 stated “[i]t’s pretty good I think,” and P26 said that “[t]he current process is quite good.”

C2: Portability Reviewers experience the same portability issues that are experienced by users, and highlight the process of installing artifacts and their dependencies as a significant challenge during review. P121 shares that “[i]nstalling software and dependencies was the biggest challenge as a reviewer,” P35 complains of an “[o]verly complicated process for setting up the artifact and its environment,” and P142 cites the “requirement to install complex and invasive program/system libraries” as a challenge when reviewing. Artifacts are often only tested in the environment in which they were produced, and, in some cases, may be hardcoded to that particular environment (e.g., the author’s machine). P107 recalls dealing with “complicated instructions for installing and executing artifacts, that are platform depend[en]t, and have not been properly tested on settings different than the authors’ machines.”

Virtual machines (VMs) can be used to mitigate these portability challenges and to avoid the security risks associated with executing untrusted code on the reviewer’s host machine. However, VMs are not without their own problems. P50 highlights that “using virtual machines does not solve all problems (because running across different OSes leads to problems and sizes are huge),” and P64 shares that “some AEC folks try to evaluate the performance claims of the paper, but the software is typically in the form of a VM. Being in a VM, it is not really possible to accurately assess performance, nor should the authors be penalised if the reviewer cannot assess the performance.” Containers may be used in lieu of virtual machines to address performance and size concerns, but as P18 identifies: “artifacts, esp. software artifacts degrade over time. Something that compiles today is probably not to compile in five years, because you do not have the same system. And (surprise!) having containers does not always solve this problem.”

C4: Tacit knowledge A lack of adequate documentation to understand, install, and use an artifact can make reviewing difficult or impossible. For example, P6 complains of “poor documentation from authors describing which claims in the paper are supported by the artifact,” and P98 recalls a case where they “could not reproduce the paper analysis relying solely on the artifact instructions, [and] had to read the paper to learn what was missing.” Similarly, P60 shares that “even though that happened only once, I had to replicate a study in which the replication package seemed to be complete. However, when I followed the instructions, I couldn’t find the same results found by the authors. Unfortunately, the package was missing an instruction.”

Some reviewers suggest that automation, in the form of scripts that can replicate the results of an experiment, may be used to mitigate the tacit knowledge problem. However, other reviewers stress that these solutions are not a panacea.

P74 points out that, in the event where an artifact does not meet expectations, “[i]t is often difficult to know whether the submitted device is faulty, or if [you don’t know how to] use it.” In other cases, an automated script may essentially behave as a blackbox that produces the intended results, but provides the reviewer with no insight into whether the process used to arrive at those results is sound. As P25 identifies, a desire for automation may also prevent methodological errors from being noticed: “Artifact evaluation committees – by asking for replicable scripts – are likely to replicate the same mistake (if any) the original authors made.” And while automation may make it easier to replicate the results of an experiment, documentation is still needed to conduct a thorough evaluation of an artifact. As P52 highlights, “[i]t was easy to reproduce the experiments, but difficult to understand where should I add more case studies.”

C6: Lack of standards and guidelines Reviewers complain about ambiguous or missing guidelines for reviewing artifacts, and a lack of criteria for what is considered to be a good artifact. For example, P141 says that the “artifact evaluation scheme [is] difficult to understand,” and P74 shares that “evaluation criteria are not always easy to apply.” Likewise, artifact badging schemes may also lack sufficiently clear and detailed criteria: P49 complains that the “description of badge requirement[s] on ACM website is very confusing, both for the reviewer and authors of the artifact,” and P12 says “sometimes it is hard to decide which badge to give since the information in the ACM Artifact Evaluation and Badging Guideline is too general.”

The purpose of the artifact review process itself is often unclear, and reviewers may hold different views. Artifact review may simply involve running a set of automated scripts to replicate a result, or it may involve a more thorough investigation of the assumptions, limitations, and generality of the artifact. P6 points out that:

“[There are] [u]nclear expectations as to how thorough the reviewing process should be. Some AEC members uncritically run the scripts provided by the authors. Others perform a deep-dive into the code and occasionally uncover severe problems with the artifact. These deep dives are extremely time-intensive, however, and often require multiple days of work. But if the AEC doesn’t do it, will anybody ever? Probably not.”

Several reviewers, such as P32, believe that the goal of artifact review “should be to see that each artifact gets accepted (via shepherding).” But, as P6 notes, this can create tensions when artifacts do not meet the expectations set out in the paper: “Unclear relationship between the artifact review process and the paper shepherding process. An improperly implemented benchmark should be grounds for paper rejection.” To resolve this tension, P11 suggests that artifacts “should be evaluated as part of paper if [they are] the main result.” At one extreme, P124 believes that “[p]ublishing artifacts should be a precondition (if there are no legal etc. reasons to not publish them) to get papers accepted.” Similarly, P146 shares that “[a]rtifacts as replication packages should be mandatory to get a paper published in journal and conferences.” At the opposite extreme, other reviewers, such as P143, believe that the artifact review process should be faster and more lightweight means of “rubber stamping” artifacts: “It shou[l]d be less [of] a burd[e]n to get an artifact ‘stamp’. Often, many papers provide links to repositories and so

on. Why could this not be automatically evaluated by a committee as soon as a paper gets accepted?”

C9: External constraints Licensing issues can make it difficult or impossible to review certain artifacts as reviewers may not possess the necessary licenses to install and use the artifact. P65 highlights that there is “no obvious pathway for what to do if the software artifact requires a proprietary component (Windows, a commercial IDE). Some legal clarity around this would be useful.”

C10: Lack of reviewer incentives Dealing with technical obstacles and artifacts that are unsuitable for sharing (e.g., artifacts with missing documentation) requires significant time and effort on behalf of the reviewer. Reviewers highlight that these efforts are not recognized and rewarded to the same extent as paper reviewing. P115 states that “[t]he credit for AE reviewers is low,” and P64 shares, “we usually just make students do AEC. This is good in some ways because it gets students involved in the reviewing process. But it also reinforces the idea that artifacts are not a primary thing and that authors should really focus on the paper.” A lack of incentives and sufficient recognition may lead to inconsistent and low-quality reviews due to reviewers that are not fully invested in the process.

C11: Technical obstacles A variety of obstacles can complicate the process of reviewing artifacts. Several reviewers identify long execution times and a lack of adequate compute resources as complicating factors when reviewing. P50 shares, “it is hard to decide what to do for artifacts that by nature take more time than one can devote to evaluating a single artifact (\sim one day).” Artifacts that require significant disk space, such as virtual machine images, can also pose a number of challenges when obtaining and reviewing artifacts. As P52 highlights, “the authors provided a large VM that didn’t fit in my disk. So I have to uninstall some things on my computer.” Combined with slow download speeds, large file sizes can also make it difficult to obtain artifacts.

In some cases, fundamental technical obstacles make it practically infeasible to replicate an experiment via an artifact: In the words of P64, “an interesting tension is that the reviewers do not often have the capabilities to actually run the artifact. For example, many cloud-based contributions, fuzzers, automated program repair, require significant hardware to actually do anything useful.”

C12: Limited communication The challenges of reviewing are often exacerbated by the lack of open communication channel between creators and reviewers. P72 shares that “sometimes AECs adopt a single review (as against multiple rounds of contacting authors). This can be really frustrating especially because it takes a considerable amount of time to build an artifact and a minor glitch in the installation instructions should not be grounds for rejection.” Indeed, reviewers highlight the importance of communication. P133 says that it is “understandable that an artifact from a researcher is not perfect. An important thing is that the owners of the artifact are willing to help the users resolve outstanding issues,” and P52 points out that “usually, people answer quickly when one report a bug in an artifact that you are using.”

However, in cases where such communication is permitted, issues may still persist due to unresponsive authors. For example, P25 shares that they “found

that there’s a trade-off of putting weight on authors’ and reviewers’ shoulders: Ideally, the reviewers should take into account the authors’ attempts in helping them get the tools to run. But with unresponsive authors and in lack of a clearly defined protocol, the reviewers might have to wait several days without knowing if there will eventually be a solution attempt – which, if successful, is then followed by half a day of performing the actual review.”

Insights for RQ2.3

- Few papers (26%) that contain artifacts are submitted for evaluation
- Most papers (86%) that are submitted for evaluation are accepted
- There is a range of views on the purpose and extent of artifact review
- Certain artifacts are difficult to assess due to time and resource limitations, and licensing issues
- Artifact reviewing is perceived to be less rewarding than paper reviewing
- A lack of open communication between creators and reviewers, coupled with unclear standards and expectations for packaging and assessing artifacts, leads to confusion and frustration between creators and reviewers

5 Secondary Analysis

In this section, we develop a more complete understanding of artifacts by considering the results of our parallel studies in concert and exploring relationships among the themes and components we discover. Specifically, we use DoI to understand the factors that influence the creation, sharing, use, and review of artifacts (Sect. 5.1), the communication channels through which artifacts are understood and shared (Sect. 5.2), how artifacts evolve over time (Sect. 5.3), and the role of artifacts within the social system of the software engineering research community. When we approach this landscape of actors, experiences, and values holistically, it becomes clear that many of the challenges identified in our primary analysis are related to each other, and that challenges impact people differently depending on whether they are creating, sharing, using, or reviewing artifacts.

5.1 Innovations

Below, we describe the qualities of artifacts that influence their diffusion in terms of the characteristics of innovations within DoI theory: *compatibility*, *trialability*, *complexity*, and *relative advantage*.

Compatibility: Artifacts should be familiar enough to fit within the existing needs and expectations of their intended users. Users desire artifacts that achieve this by adhering to existing packaging, interface, and use case conventions.

Trialability: Artifacts should clearly advertise their contents, purpose, and claims to allow potential users to quickly determine whether an artifact suits their individual needs. Respondents identify that ideal artifacts provide a clear directory structure that is described as part of the README, along with a brief summary of the intended use cases, claims, and limitations of the artifact.

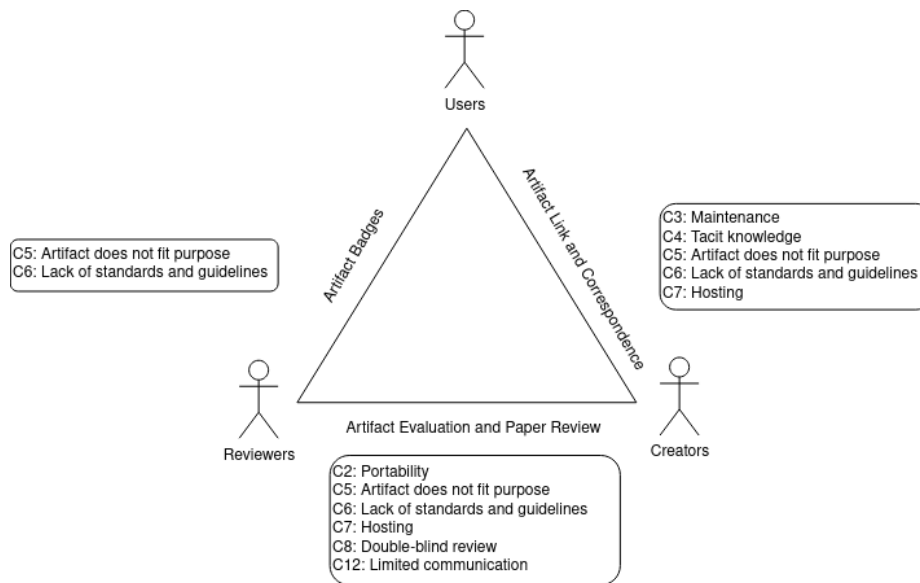


Fig. 5: An overview of the communication channels that exist between artifact creators, users, and reviewers, annotated with the high-level challenges that either affect or are affected by those communication channels.

Complexity: Artifacts should be reasonably easy for their intended audience to understand and use. For tool artifacts, this includes providing ample documentation and examples, including tests, and ensuring the artifacts are self-contained and quick to download and run. For data artifacts, this includes providing both raw and preprocessed data; describing the format of the data, how it was collected, identifying any assumptions that were made, and providing sample queries that demonstrate how the artifact can be used.

Relative Advantage: Artifacts should offer tangible benefits to their users over choosing not to use them, whether that be by providing a deeper understanding of an associated study than is available in the paper, or by providing a reusable tool or dataset and thereby avoiding the need to create one from scratch.

5.2 Communication Channels

Figure 5 provides an overview of the communication channels that exist between artifact creators, users, and reviewers. Artifact evaluation and paper review serve as a channel between creators and reviewers (Sect. 5.2.1), artifact links provide a channel between creators and users (Sect. 5.2.2), and artifact badges act as a channel between reviewers and users (Sect. 5.2.3). Below, we analyze the challenges associated with each of these communication channels.

5.2.1 Artifact Evaluation and Paper Review

Artifact evaluation provides a communication channel between creators and reviewers that serves to assess the quality, claims, and usability of artifacts. The effectiveness of artifact evaluation is hindered, in part, due to the lack of open communication between creators and reviewers (**C8: Double-blind review; C12: Limited communication**). Despite the best efforts of authors to anticipate potential issues, tacit knowledge and unidentified assumptions can complicate or prevent the review process (**C2: Portability; C4: Tacit knowledge**). On the other end, reviewers spend considerable efforts identifying and addressing these issues during evaluation, often with little or no assistance from creators, leaving them with little time for more than a superficial evaluation.

The absence of clear guidelines and expectations around artifacts and artifact review represent a shared source of confusion and frustration between creators and reviewers (**C6: Lack of standards and guidelines**). P128 shares that their “artifact [was] rejected because of wrong expectations about the nature of the artifact,” and P61 says, “[s]ome good artifacts are rejected due to unclear standard of artifacts.” Such negative experiences can deter authors from participating in artifact evaluation entirely, thereby undermining the reason for the existence of AECs. For example, P31, says that “based on wildly different reviewer expectations between different artifact evaluation committees, I have decided to not waste time to prepare a formal artifact package recently, but have focused instead on open source releases.”

In cases where there is no formal AEC, authors may optionally include artifacts during paper submission. However, authors identify that sharing artifacts as part of a double-blind paper submission, rather than artifact evaluation, introduces the additional challenges of removing identifying information and finding a suitable place to anonymously host the artifact (**C7: Hosting; C8: Double-blind review**). In the case that authors opt to provide an artifact with their submission, P64 points out that it is “not obvious to me if even the reviewers look at these artifacts when evaluating the paper.”

5.2.2 Artifact Links

Artifacts are typically distributed by their creators to potential users by a URL that is included in the associated publication (**C7: Hosting**). Users point out that artifacts are often provided with little or no documentation about their contents, how to install and use them, and how they support the associated submission (**C4: Tacit knowledge; C5: Artifact does not fit purpose**). These difficulties, stemming from a lack of documentation and mismatched expectations, lead potential users to abandon their efforts to use artifacts.

Creators are unsure of how their artifacts should be packaged and what documentation should be provided to avoid these difficulties. This is partly due to inherent challenge of tacit knowledge, but is exacerbated by the lack of broadly accepted community-wide standards, guidelines, and norms around what artifacts should have, be, and do (**C4: Tacit knowledge; C6: Lack of standards and guidelines**).

5.2.3 Artifact Badges

Artifact badges serve as a means of signaling the existence, quality, and reusability of an artifact to prospective users. According to Krishnamurthi and Vitek (2015), the presence of AECs “sends a message that artifacts are valued and are an important part of the contribution of papers,” “encourages authors to produce reusable artifacts, which are the cornerstone of future research,” and takes the research community closer to “the point where any published idea that has been evaluated, measured, or benchmarked is accompanied by the artifact that embodies it.”

However, our survey participants complain that the standards of artifact review are often unclear and vary considerably between different conferences and even editions of the same conference (**C6: Lack of standards and guidelines**). This leads to ambiguity in the perceived meaning and value of artifact badges. As a result of this ambiguity, the reliability of artifact badges as an indicator of quality to potential users is diminished (**C5: Artifact does not fit purpose**).

5.3 Time

Over time, artifacts may become inaccessible by others as short-term hosting strategies fail, such as when an artifact hosted on a student website becomes unavailable after the student moves on (**C3: Maintenance**). Even in cases where artifacts are still available, those artifacts are likely to “bitrot” over time and become more unusable as the environments in which they are used become increasingly dissimilar to those in which they were produced (**C2: Portability**).

Continual maintenance is needed to fight against the inevitable process of bitrot. However, these activities are perceived as requiring considerable time and resources while yielding potentially diminishing returns to both author and user (**C1: Not worth it**). The labor and resources costs required to maintain an artifact dissuade some authors from sharing their artifacts in the first place.

Alternatively, authors may mitigate the effects of bitrot by using a VM, container, or another form of virtual environment to package their artifacts in a predictable, ready-to-use state. This requires upfront effort from the authors, as retroactively packaging artifacts as VMs or containers can be challenging and may not accurately represent the environment that was used to conduct experiments (**C2: Portability**).

5.4 Social System

We find that the community, as a whole, benefits when high-quality artifacts are shared, as they provide greater knowledge, encourage transparency, and catalyze further research. However, the traditional academic system often does not reward artifact authors for the creation or maintenance of high-quality artifacts, which results in insufficient time and resources devoted to artifact sharing (**C1: Not worth it**). The lack of incentives for both creators and reviewers can often lead to the creation of lower-quality artifacts that are not maintained over time (**C3: Maintenance**), and a less-effective artifact evaluation process that does not identify and address potential issues (**C10: Lack of reviewer incentives**). This sit-

uation prevents the community from enjoying all of the full, long-term benefits of artifacts.

Just as the creation and sharing of artifacts is perceived to be a low-reward activity, reviewers also note a lack of reward and recognition for their efforts, and point out that artifact review is less valued than paper reviews. Since artifact review is often less valued than technical track reviewing, artifact reviewers tend to be less experienced and have fewer resources. This can lead to a situation in which neither the authors or reviewers are fully invested in the process of artifact evaluation, and when combined with the lack of communication between the two parties (**C12: Limited communication**), may ultimately produce inconsistent reviews and a failure to improve and ensure the quality of artifacts. Ultimately, the lack of incentives for both authors and reviewers lead to problems downstream for artifact users that are not identified and addressing during review, such as a lack of portability, incomplete documentation, and mismatched expectations (**C2: Portability; C4: Tacit knowledge; C6: Artifact does not fit purpose**).

6 Recommendations

In this section, we follow IS principles to make recommendations to improve the experience and outcomes of artifact creation, sharing, use, and review. Our recommendations, presented in Table 6, are derived from the challenges identified in our primary analysis (Sect. 4), based on an understanding of how those challenges are related within the wider context of the research community (Sect. 5), and are targeted at specific subpopulations. We discuss our recommendations below.

Note that our recommendations are based on popular implementations of AECs. If the fundamental role and purpose of AECs changes in the future, then some recommendations may no longer apply. For example, if artifacts are explicitly reviewed as part of the paper review process, as we believe they should be.

	Recommendation	Derived from Results	Supported By Literature
■	R1 Describe the contents, structure, and purpose of artifacts <ul style="list-style-type: none"> • provide citation guidelines • consider a permissive license that ensures creators are credited 	C1, C4, C5, C13, Communication, Social System, Innovation	Basili et al. (2007) • Stodden (2009b) • Collberg and Proebsting (2016) • Flittner et al. (2017) • Rougier et al. (2017) • Heumüller et al. (2020)
■	R2 Create a self-contained artifact <ul style="list-style-type: none"> • consider providing a VM or container image for the artifact • aim to produce an artifact that is amenable to review 	C2, C11, Communication	Heumüller et al. (2020)
■ ♣	R3 Establish a long-term plan for creating and sharing artifacts <ul style="list-style-type: none"> • use a long-term archival service to store artifacts indefinitely • establish and state maintenance expectations • foster a work environment that values artifacts 	C1, C3, C7, Time, Communication	Basili et al. (2007) • Collberg and Proebsting (2016) • Smith et al. (2016) • Rougier et al. (2017) • Heumüller et al. (2020)
★	R4 Obtain and use a clear rubric to evaluate artifacts <ul style="list-style-type: none"> • determine who is responsible for reviewing artifacts • obtain a rubric from the AEC or conference chairs • or else devise a rubric and include it in the review 	C6, Communication	
♠	R5 Align author and reviewer expectations <ul style="list-style-type: none"> • establish and communicate the purpose of artifact evaluation • communicate evaluation criteria to both reviewers and authors • improve review consistency by building institutional knowledge 	C1, C6, Communication, Social System	Hermann et al. (2020)
♠	R6 Reduce the opportunity cost of reviewing artifacts <ul style="list-style-type: none"> • improve communication between authors and reviewers • consider introducing a “Best Artifact Reviewer” award • consider providing compute resources to reviewers 	C10, C11, C12, Communication, Social System	Hermann et al. (2020)
♦	R7 Recognize and fund artifact-related activities <ul style="list-style-type: none"> • consider artifacts during hiring, reappointment, and promotion • consider artifacts when reviewing grant applications • consider providing funding to support artifact hosting, maintenance, and review 	C1, C7, Social System, Communication, Innovation, Time	NSF (2011) • Engineering and Physical Sciences Research Council (2011) • European Commission (2016, 2020)
♦	R8 Devise a long-term strategy for artifact sharing and evaluation <ul style="list-style-type: none"> • establish a common artifact description format 	C1, C4, C6, C10, Social System, Communication	Hermann et al. (2020)

Table 6: An overview of our recommendations, annotated with the challenges that they tackle and their relevant DoI dimensions, and a list of supporting studies and aligned proposals from the literature. Recommendations are targeted at creators (■), advisors and mentors (♣), reviewers (★), process organizers (♠), and community leaders (♦).

6.1 R1: Describe the contents, structure, and purpose of artifacts

Many of the issues experienced by artifact users and reviewers ultimately stem from mismatched expectations over what the artifact should have, be, and do. To minimize such issues, creators should improve communication by clearly documenting (e.g., via a README) the important details of their artifacts. Below, we take inspiration from the ideas of sharing contracts (Collberg and Proebsting, 2016), meta-artifacts (Flittner et al., 2017), and data sharing agreements (Basili et al., 2007), discussed in further detail in Section 7, to identify some of those important details.

For all artifacts, creators should provide a clear description of the contents, structure and purpose of the artifact, and indicate how the artifact supports the claims made in the associated paper. To ensure that creators are credited for their efforts, artifacts should be packaged with citation guidelines that make it clear how those using and extending the artifact should cite the work. Additionally, creators should consider using a licensing framework that encourages others to build upon their work while ensuring that creators are credited, such as the Reproducible Research Standard (Stodden, 2009a,b), Creative Commons Attribution License (Creative Commons, 2013), and Apache License, Version 2 (Apache, 2004).

Tool artifacts should follow existing software engineering best practices by including details about the intended uses, limitations, and overall generality of the tool, and providing example uses. Data artifacts should state the contents and format of the dataset (e.g., columns and units of measurement), and describe the methodology that was used to collect the data (i.e., provenance).

6.2 R2: Create a self-contained artifact

Creators of tool artifacts can take steps to anticipate, identify, and address portability issues that may be encountered by users and reviewers. At a minimum, creators should provide installation instructions, a list of dependencies, and a description of the relevant details of the systems on which the tool was developed and used (e.g., operating system, machine specifications, compiler versions). These instructions should either be checked manually, or, better yet, automatically over the course of the project using a continuous integration service such as TravisCI⁶ or GitHub Actions.⁷ Creators can simplify their installation process by using popular build and package management systems (e.g., CMake [Kitware, 2021], Gradle [Gradle, 2021], pip [PyPA, 2021], make [Free Software Foundation, 2021]), and avoiding custom or exotic solutions. Issues due to missing dependencies, mismatched versions, and platform and environment incompatibilities can be mitigated by using virtualization (e.g., VirtualBox [Oracle, 2021], QEMU [QEMU, 2021]) or containerization (e.g., Docker, Podman [RedHat, 2021]) to package artifacts and run experiments with greater reproducibility.

To maximize the effectiveness of the artifact review process, creators should strive to produce artifacts that are amenable for evaluation by reviewers. This can be partly achieved by establishing expectations, stating claims, and taking

⁶ <https://travis-ci.org> [Date Accessed: March 13th 2021]

⁷ <https://github.com/features/actions> [Date Accessed: March 13th 2021]

steps to avoid potential portability issues. In the case where an artifact involves reproducing lengthy and expensive experiments (e.g., program repair), creators should try to provide a representative proof-of-concept form of the artifact that provides partial evidence of its claims on a smaller dataset.

6.3 R3: Establish a plan for creating and sharing artifacts

At the beginning of a research project, all members of the research team, including primary researchers, advisors, and mentors, should establish a plan for the creation and dissemination of any associated artifacts (e.g., tools, datasets, source code).

Plans should ensure that enough time is allocated to artifact-related efforts (e.g., writing documentation, tests, installation instructions) throughout the entire research project, rather than deferring activities until after paper submission and acceptance. As part of that effort, advisors and mentors should work to establish a culture of valuing artifacts within their research groups that recognizes the effort required to produce and maintain high-quality artifacts. Just as students are often taught how to do good research and write valuable reviews, mentors and advisors should provide instruction and guidance as to how to prepare and evaluate artifacts.

Plans should also be made for the long-term maintenance of the artifact. These plans should establish expectations around how much support will be given for an artifact, for how long, and by whom the artifact will be maintained. Advisors, in particular, should consider making transition plans for the case that key maintainers move onto other organizations. To avoid mismatched expectations between creators and users, maintenance plans should be included as part of the artifact description, together with contact information for the artifact maintainers.

To avoid long-term availability issues, creators should plan for the hosting and long-term archival of their artifacts. Where possible, creators should use archival services such as Zenodo (European Organization For Nuclear Research and OpenAIRE, 2013), Figshare,⁸ and Software Heritage (Cosmo and Zacchiroli, 2017) to guarantee the long-term availability of a snapshot of their artifacts via a unique identifier. Traditional file sharing services (e.g., Dropbox, Google Drive) and temporary hosting solutions (e.g., a university website) should be avoided as they are not designed for reliable, long-term storage. Naturally, some artifacts (e.g., tools) evolve over time and may be used in multiple paper submissions. VCS-based hosting services (e.g., GitHub, GitLab,⁹ BitBucket)¹⁰ may be used to host the working version of the artifact and to show its development history, but the authors should also archive a snapshot which corresponds to a specific paper using archival services.

6.4 R4: Obtain and use a clear rubric to evaluate artifacts

Reviewers should first establish who, if anyone, is responsible for reviewing the artifacts associated with a paper. For conferences that have an AEC, technical

⁸ <https://figshare.com> [Date Accessed: March 13th 2021]

⁹ <https://www.gitlab.com> [Date Accessed: March 13th 2021]

¹⁰ <https://www.bitbucket.org> [Date Accessed: March 13th 2021]

reviewers should explicitly define the aspects of the artifact that they are reviewing. For (aspects of) artifacts that they do not review, technical reviewers may encourage authors to submit their artifacts for evaluation. For conferences that do not have an AEC, technical reviewers may consider reviewing artifacts, to a limited extent, themselves.

Before embarking on the review process, reviewers should ensure that they have a clear rubric or set of criteria for evaluating artifacts. Ideally, this should come from the conference chairs, AEC chairs, or journal editors, and be available to authors ahead of paper submission, reviewers prior to reviewing, and all consumers of accepted papers. Reviewers should become familiar with the rubric before reviewing, and apply it consistently. If they do not have access to a set of criteria, they should ask the AEC chair, conference chair, or journal editor if there is one available, or collaborate with other committee members to create one. In cases where that is not possible, reviewers should devise their own rubric, apply it consistently during review, and communicate the criteria that was used to evaluate the artifact to its creators. To enhance the overall effectiveness of the review process, reviewers may also consider using their position to ask the conference, AEC, or journal to develop criteria for evaluating artifacts before accepting an invitation to review.

6.5 R5: Align author and reviewer expectations

To align expectations between authors and reviewers, conferences and journals should release clear and consistent evaluation guidelines for artifacts as part of their call for papers. These guidelines should include a description of the high-level goals and purpose of artifact review, the extent to which artifacts will be reviewed, and the criteria according to which artifacts will be evaluated. For example, guidelines should state whether reviewers should attempt to replicate the same result as the original experiment, or if they should attempt to explore the effects of changing parameters and/or using different data.

Authors point out that there is not enough time to prepare artifacts. For example, P101 shares that “time constraints made it impossible to put together a high-quality, reproducible artifact before the submission deadline.” To encourage authors to proactively develop their artifacts and allow them to better manage their time and efforts, conferences should consider including artifact submission dates as part of their call for research papers.

Hermann et al. (2020) find that most AECs are composed of reviewers who have never served on an AEC before. The resulting lack of institutional knowledge, coupled with a lack of guidelines and clear purpose for artifact evaluation, can lead to inconsistent reviewing and, in some cases, discourage authors from submitting their artifacts for evaluation. To build institutional knowledge and improve the quality and consistency of reviews over time, AECs should consider having a way for reviewers to hand off information from one “generation” to the next. As part of this process, AECs may hold a retrospective at the end of reviewing, and make suggestions for improving their guidelines and implementation. Journals and conferences without a separate AEC may also consider conducting such a retrospective and periodic evaluation of guidelines. Consistency between artifact and paper re-

views may also be improved by allowing technical reviewers to leave confidential remarks to the AEC.

6.6 R6: Reduce the opportunity cost of reviewing

Artifact evaluation is often a labor-intensive activity that, compared to traditional technical reviewing, has fewer perceived rewards (i.e., reviewing has a high opportunity cost). This situation can lead to lower-quality reviews, and consequently, a less-effective review process. To address this situation, AECs should take steps to reduce the burden on reviewers and create incentives. Providing clear reviewing guidelines, building institutional knowledge, and offering adequate compute resources (e.g., cloud compute credits) can make it easier for reviewers to review artifacts effectively. AECs may also consider introducing a “lazy second evaluation” where artifacts are only assigned a second reviewer if the first has concerns about building, running, or evaluating the artifact. Improving communication between authors and reviewers may reduce the number of cases where a second reviewer is needed.

The overall efficiency and effectiveness of the review process can be improved by eliminating communication issues between authors and reviewers. One way to both improve communication and align expectations between authors and reviewers is to require authors to provide a separate document that describes their artifact. This document should contain the details outlined in Section 6.1, including a description of the contents, claims, and limitations of the artifact, and how it relates to the results of the associated paper. The artifact review process can then be used as an opportunity to “certify” the associated document for the artifact. Users of the artifact can then use certified documents to help them quickly establish expectations around what a particular artifact should have, be, and do.

For some AECs, communication between authors and reviewers is one-way. Given the inevitability of technical difficulties, this has a potential to create an adversarial situation that fails to thoroughly identify and address issues that may be experienced by eventual users of the artifact. AECs should consider using continuous, two-way communication (i.e., shepherding) that allows reviewers to more easily evaluate the claims of the artifact, and gives room for artifacts to be improved. One way to implement such communication without revealing the identity of the reviewer is through the use of an anonymizing mailbox that allows creators to respond to reviewer questions.

Just as several conferences have recently introduced “best reviewer” rewards, AECs may consider a similar award for artifact reviewers. This would increase the potential reward and prestige of artifact reviewing, and with reductions to the cost of reviewing, would decrease the opportunity cost of reviewing.

6.7 R7: Recognize and fund artifact-related activities

In the words of P64, “artifacts are useful to the community, but for purposes of academic promotion it is not clear they have any value there. Unless artifacts are considered as part of career growth I do not think it will see wide adoption.”

To better align incentives, hiring and reappointment and promotion committees should reward those who share their artifacts consistently with the community. To that end, organizations such as the Computing Research Association (CRA) should consider developing best practices for incentivizing and evaluating the impact of artifact release and sustainment for tenure and promotion policies (c.f., Patterson et al. 1999).

Funding agencies could ask (co-)principal investigators to provide artifact creation and sharing plans in their grant applications, and may consider the applicants track record of sharing formally evaluated artifacts during review. Several funding agencies, including the National Science Foundation (NSF), National Institutes of Health (NIH), Engineering and Physical Sciences Research Council (EPSRC), and the European Commission (EC) have already implemented explicit data management policies to encourage the sharing and reuse of artifacts (NSF, 2011; NIH, 2003; Engineering and Physical Sciences Research Council, 2011). The NSF, for example, requires that proposals provide a two-page data management plan (DMP) that describes “the data, metadata, scripts used to generate the data or metadata, experimental results, samples, physical collections, software, curriculum materials, or other materials to be produced in the course of the project.” These efforts could be supported by grants supplements for the long-term maintenance and archival of artifacts. For example, DMPs are optional for European Union Horizon 2020 proposals, but may be used to cover certain costs related to open access (European Commission, 2020, 2016).

To tackle some of the technical obstacles of artifact reviewing, organizations such as the ACM and IEEE should consider providing funding for compute resources (e.g., in the form of cloud-compute credits) to conferences and journals.

6.8 R8: Establish a long-term strategy for artifact sharing and evaluation

Editors and steering committees have experience and influence that span multiple years, and are responsible for overseeing the long-term continuity and direction of their respective venue. We recommend that these entities devise a long-term strategy for the sharing and evaluation of artifacts, and, in the case of conferences, leave the implementation of that strategy to the AEC chairs.

Professional organizations also play a major role in the long-term future of artifact sharing and evaluation. The ACM, in particular, has pioneered initiatives to enhance the visibility, usability, and availability of artifacts, such as its badging policy and long-term archival of artifacts via the ACM Digital Library (Association for Computing Machinery, 2020, 2018). Going forward, professional organizations should work with community leaders to develop a standard artifact description format that communicates the important details of artifacts.

To ensure that guidance maintains relevant and reflective of the community over time, journal editors, steering committees, and professional organizations should periodically conduct a systematic review of artifact evaluation to identify new challenges and disseminate best practices within the community.

7 Related Work

7.1 Replicability

The concept of software artifacts today traces its lineage back to the idea of “laboratory packages” which describe an “experiment in specific terms and provides materials for replication” (Shull et al., 2002, 2008; Brooks et al., 2008; Basili et al., 1999). Artifacts, as we understand them, are both the materials for reuse, repurposing, and replication, and also, in some cases, the laboratory package itself. There are opposing views on the importance of laboratory packages within the literature. Lindvall et al. (2005) find that it is cheaper for researchers to reproduce an experiment, even with small modifications, by using a laboratory package, rather than designing the entire replication from scratch. Shull et al. (2008) argue that laboratory packages allow others to inexpensively (1) ensure that a given result is reproducible and thereby increase confidence in that result, and (2) understand the sources of variability that affect a given result so as to understand its scope and limitations. Shull et al. consider conceptual replications (i.e., a different lab provides an independent implementation and conducts their own experimental setup to confirm results) to be too expensive to be considered the norm. This view appears to be shared by many artifact evaluation committees: Reflecting on their experiences of running AECs at multiple conferences over several years, Krishnamurthi and Vitek (2015) state that “repeatability is an inexpensive and easy test of a paper’s artifacts, and clarifies the scientific contribution of the paper,” and that reproducibility (i.e., conceptual replications) “is an expensive undertaking and not something we are advocating.” Opposed to this view, Kitchenham (2008) cautions that laboratory packages allow flaws in the original experiment design to be repeated and can be expensive in the long run, and that exact replications are of limited use as they cannot be used in meta-analyses. We observe a similar spectrum of views on the role of artifacts within the responses of our survey participants. To that end, we encourage conferences, journals, and AECs, where applicable, to explicitly state the high-level goals and purpose of their artifact evaluation process.

7.2 Existing Recommendations

A number of proposals have been made to improve artifacts: Collberg and Proebsting (2016) propose that authors include “sharing contracts” with their papers that give basic information about the artifacts for that paper, including the length of time and extent (e.g., bug fixes, feature requests) to which those artifacts will be supported. In the field of Software Defined Networks, Flittner et al. (2017) propose that artifacts should be accompanied by *meta-artifacts*, which describe the tools and parameters that were used during an evaluation. Stodden (2009b,a) proposes the Reproducible Research Standard, a licensing framework that promotes the sharing of artifacts and ensures that authors are credited for their work. Numerous public platforms have been proposed that use virtualization and containerization to facilitate the long-term persistence and replication of data and source code artifacts (Brammer et al., 2011; Austin et al., 2011; Jimenez et al., 2017; Meng et al., 2016; Timperley et al., 2018; Fursin et al., 2016). Coding conventions and

best practices have also been proposed to make it easier for authors to produce high-quality artifacts that can be more easily understood, reused, and replicated by others (Li-Thiao-Té, 2012; Krishnamurthi, 2014; Stodden et al., 2014). The majority of these various proposals are technical in nature and are largely aimed towards artifact creators. We incorporate elements of these proposals into our own recommendations, but also include sociotechnical suggestions, and make tailored recommendations to particular subpopulations of the research community.

Carver (2010) proposes a set of guidelines for reporting replications of software engineering studies, which include describing the original study, providing the motivation and important details of the replication, and reporting both consistent and inconsistent results. Carver et al. (2014) perform a user study to determine the effectiveness of the proposed guidelines and find that, overall, both reviewers and authors view them positively, provided that they do not prescribe specific paper outlines or exact content. Our recommendations are aimed at the various actors responsible for directly and indirectly creating and sharing artifacts. We do not make recommendations to artifact users or those performing replication studies.

Basili et al. (2007) identify six important properties related to the sharing of artifacts based on their personal experiences across several projects: Permission, credit, feedback, protection, collaboration, and maintenance. Our results corroborate those data sharing properties and identify additional challenges related to the creation, sharing, use, and review of artifacts. Based on those data sharing properties, they outline the notion of data sharing agreements, which help to establish expectations around artifacts (e.g., attribution, maintenance, costs), and encourage the community find a means to collect and publish such documents. We believe that Artifact Evaluation Committees, which were not introduced until several years after Basili et al.’s recommendations, may be a successful vehicle for implementing such proposals on a larger scale within the community, and we include the ideas of data sharing agreements in our recommendations.

7.3 Artifact Studies

Childers and Chrysanthis (2017, 2018) investigated the incentives of submitting artifacts for evaluation. They looked at all publications at three conferences (ECOOP, OOPSLA, and PLDI) between 2013 and 2016 and compared the average citation count of papers that were accepted by an AEC (AE papers) against papers that were not (non-AE papers). Their results, while not conclusive, suggest that AE papers may be correlated with a slightly higher citation count.

Similarly, in a study of papers published at ICSE between 2007 and 2017, Heumüller et al. (2020) find that linking artifacts to a paper leads to a small, but statistically significant increase in citation count. The authors determined that approximately 76.6% of papers describe an artifact, but that only 48% were linked through a link in the paper, and only 56.4% of those artifacts were available at the links provided. In this study, we conducted a similar analysis of artifact availability across several venues, including, over a more recent, five-year period (2014–2018), and observed a linear increase in the linking and availability of artifacts over time.

Kotti et al. (2020) conducted a study of data papers published at the *International Conference on Mining Software Repositories* (MSR) between 2005 and 2018 to determine how often, by whom, and for what purpose researchers reuse

their associated artifacts. They found that 65% of data papers have been used in other studies, but that those papers are cited less often than technical papers at the same conference.

Hermann et al. (2020) conducted a survey of individuals who had served on an AEC between 2011 and 2019 at one of several venues to understand community expectations and perceptions about artifact use and evaluation. They observe a similar set of perceptions among artifact reviewers and users to those reported in our study, and make recommendations that align with our own.

Collberg et al. analyze 601 papers published at several Computer Systems conferences and journals to determine whether their accompanying code artifacts, if any, can be obtained and built by others with reasonable efforts (Collberg et al., 2015; Collberg and Proebsting, 2016). This effort generated quite a controversy among the community, and led to an effort to examine “Reproducibility in Computer Science” (Krishnamurthi, 2013b), which found different results from the original paper’s results. We did not attempt to build or use artifacts as part of our study methodology, but found similar challenges to those reported by Collberg et al. through an analysis of author survey responses (e.g., dead links, missing documentation, and lacking portability).

Shull et al. (2002) describe how “tacit knowledge” (i.e., “the transfer of experimental know-how”) can be challenging even when good artifacts are provided. Our survey participants report similar challenges in using and reviewing artifacts. We see artifact evaluation as a potential means of identifying and addressing the presence of tacit knowledge and unstated assumptions.

7.4 Experience Reports

Krishnamurthi and Vitek (2015) report their experiences of running artifact evaluation committees for five major programming languages and software engineering conferences between 2011 and 2014. They highlight how, between those years, participation greatly increased, and at OOPSLA’14, 21 of 50 accepted papers (42%) were submitted for evaluation. From our private correspondence with AEC chairs at FSE between 2015 and 2018, we found that participation was generally much lower (18–33%). In the context of our results, this difference suggests that the implementation of AECs within software engineering has room for improvement. To that end, we outline recommendations to enhance the effectiveness of AECs.

8 Conclusion

In this paper, we conducted a mixed-methods study to understand how researchers create, share, and use artifacts that accompany research papers (e.g., tools, source code, data), and the challenges that prevent the community from realizing the full benefits of those artifacts.

We find that artifact sharing is an established norm within the software engineering research community, and that an increasing majority of research papers published at ASE, FSE, ICSE, and EMSE between 2014 and 2018 included an artifact. Artifacts are highly valuable to the community as a whole, but the act of creating and sharing artifacts is perceived to be a poor investment of time

that yields relatively few career benefits compared to paper writing. The lack of incentives for creators, coupled with the technical challenges of creating and maintaining artifacts (e.g., hosting and portability) and a lack of community standards and expectations around artifacts, hinder the production of high-quality artifacts. As a result, potential users must overcome related challenges to use artifacts, and often need perform modifications to the artifact to fit their needs and expectations.

AECs are a promising mechanism for preemptively identifying and addressing artifact usability concerns. However, we observe that relatively few papers that contain artifacts are submitted for evaluation. In cases where artifacts are submitted, the efficiency and effectiveness of the evaluation process is hampered by several confounding challenges, including limited communication between creators and reviewers, missing documentation, and a lack of institutional knowledge.

We propose several recommendations, derived from our results and existing proposals from the literature, to raise the quality of artifacts and enhance the effectiveness and efficiency of artifact review. Following IS principles, we tailor our recommendations to specific groups based on a understanding of how artifact creation, sharing, use, and review takes place within the community over time.

In future work, we plan to work with process organizers (e.g., AEC chairs) to develop, evaluate, and revise evidence-based interventions to improve the effectiveness of artifact review using IS principles. We also plan to work with community leaders to disseminate identified best practices to all members of the research community.

References

- Apache. 2004. Apache License, Version 2.0. <https://www.apache.org/licenses/LICENSE-2.0> [Date Accessed: July 21st 2020].
- Association for Computing Machinery. 2018. Artifact Review and Badging. <https://www.acm.org/publications/policies/artifact-review-badging> [Date Accessed: March 30th 2020].
- Association for Computing Machinery. 2020. Software and Data Artifacts in the ACM Digital Library. <https://www.acm.org/publications/artifacts> [Date Accessed: July 29th 2020].
- J. Austin, T. Jackson, M. Fletcher, M. Jessop, B. Liang, M. Weeks, L. Smith, C. Ingram, and P. Watson. 2011. CARMEN: code analysis, repository and modeling for e-neuroscience. *Procedia Computer Science* 4 (2011), 768–777.
- V. R. Basili, F. Shull, and F. Lanubile. 1999. Building knowledge through families of experiments. *Transactions on Software Engineering* 25, 4 (1999), 456–473.
- V. R. Basili, M. V. Zelkowitz, D. I. K. Sjøberg, P. Johnson, and A. J. Cowling. 2007. Protocols in the use of empirical software engineering artifacts. *Empirical Software Engineering* 12, 1 (01 Feb 2007), 107–119.
- M. S. Bauer, L. Damschroder, H. Hagedorn, J. Smith, and A. M. Kilbourne. 2015. An introduction to implementation science for the non-specialist. *BMC Psychology* 3, 1 (2015), 32.
- M. Beller. 2020. Why I will never join an Artifacts Evaluation Committee Again. <https://inventitech.com/blog/why-i-will-never-review-artifacts-again> [Date Accessed: July 16th 2020].

- G. R. Brammer, R. W. Crosby, S. J. Matthews, and T. L. Williams. 2011. Paper mâché: Creating dynamic reproducible science. *Procedia Computer Science* 4 (2011), 658–667.
- A. Brooks, M. Roper, M. Wood, J. Daly, and J. Miller. 2008. *Replication's Role in Software Engineering*. Springer London, 365–379.
- J. C. Carver. 2010. Towards reporting guidelines for experimental replications: A proposal. In *International Workshop on Replication in Empirical Software Engineering Research (RESER '10)*.
- J. C. Carver, N. Juristo, M. T. Baldassarre, and S. Vegas. 2014. Replications of software engineering experiments. *Empirical Software Engineering* 19, 2 (01 Apr 2014), 267–276.
- K. Charmaz. 2014. *Constructing grounded theory*. sage.
- B. R. Childers and P. K. Chrysanthis. 2017. Artifact Evaluation: Is It a Real Incentive?. In *International Conference on e-Science (e-Science '17)*. 488–489.
- B. R. Childers and P. K. Chrysanthis. 2018. Artifact Evaluation: FAD or Real News?. In *International Conference on Data Engineering (ICDE '18)*. 1664–1665.
- C. Collberg and T. A. Proebsting. 2016. Repeatability in Computer Systems Research. *Commun. ACM* 59, 3 (Feb. 2016), 62–69.
- C. Collberg, T. A. Proebsting, and A. M. Warren. 2015. *Repeatability and Benefaction in Computer Systems Research: A Study and Modest Proposal*. Technical Report TR 14-04. University of Arizona. <http://reproducibility.cs.arizona.edu/v2/RepeatabilityTR.pdf>
- R. Di Cosmo and S. Zacchiroli. 2017. Software Heritage: Why and How to Preserve Software Source Code. In *International Conference on Digital Preservation (iPRES '17)*.
- Creative Commons. 2013. Attribution 4.0 International (CC BY 4.0). <https://creativecommons.org/licenses/by/4.0> [Date Accessed: July 21st 2020].
- J. W. Creswell and V. L. P. Clark. 2017. *Designing and Conducting Mixed Methods Research*. Sage Publications.
- Docker Inc. 2020. Docker. <https://www.docker.com> [Date Accessed: December 11th 2020].
- K. El Emam, E. Jonker, L. Arbuckle, and B. Malin. 2011. A systematic review of re-identification attacks on health data. *PLOS One* 6, 12 (2011), e28071.
- Engineering and Physical Sciences Research Council. 2011. EPSRC policy framework on research data. <https://epsrc.ukri.org/about/standards/researchdata> [Date Accessed: July 31st 2020].
- European Commission. 2016. H2020 Programme: Guidelines on FAIR Data Management in Horizon 2020 (Version 3.0). https://ec.europa.eu/research/participants/data/ref/h2020/grants_manual/hi/oa_pilot/h2020-hi-oa-data-mgt_en.pdf [Date Accessed: July 31st 2020].
- European Commission. 2020. Horizon 2020: Open Access. https://ec.europa.eu/research/participants/docs/h2020-funding-guide/cross-cutting-issues/open-access-data-management/open-access_en.htm [Date Accessed: July 31st 2020].
- European Organization For Nuclear Research and OpenAIRE. 2013. Zenodo. <https://doi.org/10.25495/7GXXK-RD71>
- M. Flittner, R. Bauer, A. Rizk, S. Geißler, T. Zinner, and M. Zitterbart. 2017. Taming the Complexity of Artifact Reproducibility. In *Reproducibility Workshop*

- (*Reproducibility '17*). 14–16.
- R. T. Frambach and N. Schillewaert. 2002. Organizational innovation adoption: A multi-level framework of determinants and opportunities for future research. *Journal of Business Research* 55, 2 (2002), 163–176.
- Free Software Foundation. 2021. GNU Make. <https://www.gnu.org/software/make> [Date Accessed: March 13th 2021].
- G. Fursin, A. Lokhmotov, and E. Plowman. 2016. Collective Knowledge: towards R&D sustainability. In *Design, Automation & Test in Europe Conference & Exhibition (DATE '16)*. 864–869.
- R. E. Glasgow, T. M. Vogt, and S. M. Boles. 1999. Evaluating the public health impact of health promotion interventions: the RE-AIM framework. *American Journal of Public Health* 89, 9 (1999), 1322–1327.
- C. Gómez, B. Cleary, and L. Singer. 2013. A study of innovation diffusion through link sharing on stack overflow. In *Working Conference on Mining Software Repositories (MSR '13)*. 81–84.
- Gradle. 2021. Gradle Build Tool. <https://gradle.org> [Date Accessed: March 13th 2021].
- G. C. Green and A. R. Hevner. 2000. The successful diffusion of innovations: guidance for software development organizations. *IEEE Software* 17, 6 (2000), 96–103.
- C. Hager. 2016. PDFx. <https://github.com/metachris/pdfx/releases/tag/v1.3.0> [Date Accessed: March 28th 2021].
- L. Herckis. 2018. Passing the baton: Digital literacy and sustained implementation of elearning technologies. *Current Issues in Emerging eLearning* 5, 1 (2018), 4.
- B. Hermann, S. Winter, and J. Siegmund. 2020. Community Expectations for Research Artifacts and Evaluation Processes. In *Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '20)*. 469–480.
- R. Heumüller, S. Nielebock, J. Krüger, and F. Ortmeier. 2020. Publish or Perish, but do not Forget your Software Artifacts. *Empirical Software Engineering* 25, 6 (2020), 4585–4616.
- S. Ji, W. Li, M. Srivatsa, and R. Beyah. 2014. Structural data de-anonymization: Quantification, practice, and implications. In *Conference on Computer and Communications Security (SIGSAC '14)*. 1040–1053.
- I. Jimenez, M. Sevilla, N. Watkins, C. Maltzahn, J. Lofstead, K. Mohror, A. Arpaci-Dusseau, and R. Arpaci-Dusseau. 2017. The Popper convention: Making reproducible systems evaluation practical. In *International Parallel and Distributed Processing Symposium Workshops (IPDPSW '17)*. 1561–1570.
- G. Johns. 1993. Constraints on the adoption of psychology-based personnel practices: Lessons from organizational innovation. *Personnel Psychology* 46, 3 (1993), 569–592.
- B. Kitchenham. 2008. The role of replications in empirical software engineering—a word of warning. *Empirical Software Engineering* 13 (04 2008), 219–221.
- S. C. Kitto, J. Chesters, and C. Grbich. 2008. Quality in qualitative research. *Medical Journal of Australia* 188, 4 (2008), 243–246.
- Kitware. 2021. CMake. <https://cmake.org> [Date Accessed: March 13th 2021].
- T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, C. Willing, and Jupyter Development Team. 2016. Jupyter Notebooks - a publish-

- ing format for reproducible computational workflows. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. IOS Press, 87–90.
- Z. Kotti, K. Kravvaritis, K. Dritsa, and D. Spinellis. 2020. Standing on shoulders or feet? An extended study on the usage of the MSR data papers. *Empirical Software Engineering* 25, 5 (2020), 3288–3322.
- S. Krishnamurthi. 2013a. Artifact Evaluation for Software Conferences. *SIGSOFT Software Engineering Notes* 38, 3 (May 2013), 7–10.
- S. Krishnamurthi. 2013b. Examining “Reproducibility in Computer Science”. <http://cs.brown.edu/~sk/Memos/Examining-Reproducibility/> [Date Accessed: January 6th 2020].
- S. Krishnamurthi. 2014. Guidelines for Packaging AEC Submissions. <https://www.artifact-eval.org/guidelines.html> [Date Accessed: July 17th 2020].
- S. Krishnamurthi and J. Vitek. 2015. The Real Software Crisis: Repeatability As a Core Value. *Commun. ACM* 58, 3 (Feb. 2015), 34–36.
- S. Li-Thiao-Té. 2012. Literate program execution for reproducible research and executable papers. *Procedia Computer Science* 9 (2012), 439–448.
- M. Lindvall, I. Rus, F. Shull, M. Zelkowitz, P. Donzelli, A. Memon, V. Basili, P. Costa, R. Tvedt, L. Hochstein, S. Asgari, C. Ackermann, and D. Pech. 2005. An evolutionary testbed for software technology evaluation. *Innovations in Systems and Software Engineering* 1, 1 (01 Apr 2005), 3–11.
- H. Meng, D. Thain, A. Vyushkov, M. Wolf, and A. Woodard. 2016. Conducting reproducible research with Umbrella: Tracking, creating, and preserving execution environments. In *International Conference on e-Science*. 91–100.
- M. Miranda, R. Ferreira, C. R. B. de Souza, F. M. F. Filho, and L. Singer. 2014. An exploratory study of the adoption of mobile development platforms by software engineers. In *International Conference on Mobile Software Engineering and Systems (MOBILESoft ’14)*. 50–53.
- J. M. Morse. 1991. Qualitative nursing research: A free-for-all. *Qualitative nursing research: A contemporary dialogue* (1991), 14–22.
- E. R. Murphy-Hill, E. K. Smith, C. Sadowski, C. Jaspan, C. Winter, M. Jorde, A. Knight, A. Trenk, and S. Gross. 2019. Do developers discover new tools on the toilet?. In *International Conference on Software Engineering (ICSE ’19)*. 465–475.
- A. Narayanan and V. Shmatikov. 2008. Robust de-anonymization of large sparse datasets. In *Symposium on Security and Privacy (SP ’08)*. 111–125.
- NIH. 2003. NIH Data Sharing Policy and Implementation Guidance. https://grants.nih.gov/grants/policy/data_sharing/data_sharing_guidance.htm [Date Accessed: July 21st 2020].
- NSF. 2011. Dissemination and Sharing of Research Results. <https://www.nsf.gov/bfa/dias/policy/dmp.jsp> [Date Accessed: July 21st 2020].
- H. M. O’Neill, R. W. Poudier, and A. K. Buchholtz. 1998. Patterns in the diffusion of strategies across organizations: Insights from the innovation diffusion literature. *Academy of Management Review* 23, 1 (1998), 98–114.
- Oracle. 2021. VirtualBox. <https://www.virtualbox.org> [Date Accessed: March 13th 2021].
- D. Patterson, L. Snyder, and J. Ullman. 1999. Evaluating Computer Scientists and Engineers For Promotion and Tenure. *Computing Research News* (August 1999).

- G. Premkumar, K. Ramamurthy, and S. Nilakanta. 1994. Implementation of electronic data interchange: an innovation diffusion perspective. *Journal of Management Information Systems* 11, 2 (1994), 157–186.
- PyPA. 2021. pip – The Python Package Installer. <https://pip.pypa.io/en/stable> [Date Accessed: March 13th 2021].
- QEMU. 2021. QEMU. <https://www.qemu.org> [Date Accessed: March 13th 2021].
- RedHat. 2021. Podman. <https://github.com/containers/podman> [Date Accessed: March 13th 2021].
- E. M. Rogers. 2010. *Diffusion of innovations*. Simon and Schuster.
- N. P. Rougier, K. Hinsén, F. Alexandre, T. Arildsen, L. A. Barba, F. C. Y. Benureau, C. Titus Brown, P. de Buyl, O. Caglayan, A. P. Davison, M. Delsuc, G. Detorakis, A. K. Diem, D. Drix, P. Enel, B. Girard, O. Guest, M. G. Hall, R. N. Henriques, X. Hinaut, K. S. Jaron, M. Khamassi, A. Klein, T. Manninen, P. Marchesi, D. McGlinn, C. Metzner, O. L. Petchey, H. E. Plesser, T. Poisot, K. Ram, Y. Ram, E. B. Roesch, C. Rossant, V. Rostami, A. Shifman, J. Stachelek, M. Stimberg, F. Stollmeier, F. Vaggi, G. Viejo, J. Vitay, A. E. Vostinar, R. Yurchak, and T. Zito. 2017. Sustainable computational science: the ReScience initiative. *PeerJ Computer Science* 3 (2017), 1–8. Issue 12.
- J. Saldaña. 2015. *The coding manual for qualitative researchers*. Sage.
- F. Shull, V. Basili, J. Carver, J. C. Maldonado, G. H. Travassos, M. Mendonca, and S. Fabbri. 2002. Replicating software engineering experiments: addressing the tacit knowledge problem. In *International Symposium on Empirical Software Engineering (ESEM '02)*. 7–16.
- F. J. Shull, J. C. Carver, S. Vegas, and N. Juristo. 2008. The role of replications in Empirical Software Engineering. *Empirical Software Engineering* 13, 2 (01 Apr 2008), 211–218.
- D. Silverman and A. Marvasti. 2008. *Doing qualitative research: A comprehensive guide*. Sage.
- A. M. Smith, D. S. Katz, and K. E. Niemeyer. 2016. Software Citation Principles. *PeerJ Computer Science* 2 (2016), e86.
- V. Stodden. 2009a. Enabling Reproducible Research: Open Licensing for Scientific Innovation. *International Journal of Communications Law and Policy* 13 (2009).
- V. Stodden. 2009b. The Legal Framework for Reproducible Scientific Research: Licensing and Copyright. *Computing in Science & Engineering* 11, 1 (2009), 35–40.
- V. Stodden, F. Leisch, and R. D. Peng. 2014. *Implementing reproducible research*. CRC Press.
- C. G. Teshima, B. Cleary, and L. Singer. 2013. A study of innovation diffusion through link sharing on stack overflow. In *Working Conference on Mining Software Repositories (MSR '13)*. 81–84.
- C. S. Timperley, S. Stepney, and C. Le Goues. 2018. BugZoo: A platform for studying software bugs. In *International Conference on Software Engineering: Companion Proceedings*. 446–447.
- R. A. Wolfe. 1994. Organizational innovation: Review, critique and suggested research directions. *Journal of Management Studies* 31, 3 (1994), 405–431.
- R. E. Wright, J. C. Palmar, and D. C. Kavanaugh. 1995. The importance of promoting stakeholder acceptance of educational innovations. *Education* 115, 4 (1995), 628–633.

E. Zerhouni. 2003. The NIH roadmap.