Almost 10 Years of Artifact Evaluation

What We Learned So Far

Ben Hermann - Sept. 30, 2020 - ICSME ROSE Festival

Why am I speaking about this?

And who am I anyway…

Assistant Professor for Software Engineering at TU Dortmund (as of tomorrow)

Author of a couple of artifacts

Frequent artifact reviewer

Artifact chair for ISSTA 2018

Community Expectations for Research Artifacts and Evaluation Processes

Ben Hermann
ben.hermann@upb.de
Heinz Nixdorf Institut
Universität Paderborn
Paderborn, Germany

Stefan Winter
sw@cs.tu-darmstadt.de
Dependable Systems and Software
Technische Universität Darmstadt
Darmstadt, Germany

Janet Siegmund
janet.siegmund@informatik.tu-chemnitz.de
Technische Universität Chemnitz
Chemnitz, Germany

ABSTRACT

Background. Artifact evaluation has been introduced into the software engineering and programming languages research community with a pilot at ESEC/FSE 2011 and has since then enjoyed a healthy adoption throughout the conference landscape. *Objective.* In this qualitative study, we examine the expectations of the community toward research artifacts and their evaluation processes. *Method.* We conducted a survey including all members of artifact evaluation committees of major conferences in the software engineering and programming language field since the first pilot and compared the answers to expectations set by calls for artifacts and reviewing guidelines. *Results.* While we find that some expectations exceed the ones expressed in calls and reviewing guidelines, there is no consensus on quality thresholds for artifacts in general. We observe very specific quality expectations for specific artifact types for review and later usage, but also a lack of their communication in calls. We also find problematic inconsistencies in the terminology used to express artifact evaluation's most important purpose – *replicability*. *Conclusion.* We derive several actionable suggestions which can help to mature artifact evaluation in the inspected community and also to aid its introduction into other communities in computer science.

1 INTRODUCTION

In 2016, a replicability crisis became public, when more than 1500 researchers revealed having trouble replicating previous research results [1]. This replicability crisis also reached the software engineering community, as it has embraced the importance of replication for knowledge building [3, 4, 15, 21, 22]. For example, Collberg and Proebsting could not obtain the relevant artifacts to conduct a replication, neither by contacting the authors, the authors' institution, and funding agency [7]. Also, Lung et al. describe their difficulties in conducting an exact replication, even when they were in direct contact with the authors [17]. Glanz et al. describe similar experiences when obtaining research artifacts for comparison and had to reimplement competing approaches in order to replicate results [10]. For the term *artifact*, we follow the definition provided by Méndez et al. [18], describing it as a self-contained work result with a context-specific purpose.

To improve the situation of missing or unusable artifacts, artifact evaluation has become a regular process for scientific conferences in the software engineering and programming language communities. It contributes to the larger trend towards open science in computer science. Since the first piloting of the process at ESEC/FSE 2011, many other conferences have included artifact evaluations as an additional step that authors of accepted papers may take. If their artifact is successfully evaluated the corresponding publication is marked with a *badge* [9, 11] indicating different levels by which the artifact is found to support the presented research results. Successfully evaluated artifacts are listed on the conference website and commonly linked with the paper in publication repositories such as the ACM Digital Library. Except for few venues (i.e., CAV and TACAS), where artifact evaluation is mandatory for tool papers, artifact submission usually is a voluntary activity that authors of accepted publications are invited to participate in. Journals are recently adopting the idea of artifacts as part of open science initiatives. For example, the Empirical Software Engineering journal (EMSE) encourages authors to share their data in a replication package [19]. There is preliminary evidence that papers with an evaluated artifact have higher visibility in the research community [6, 13].

There is, to the best of our knowledge, currently no evidence that artifact evaluation is leading to better artifacts for computer science research communities. The overarching goal of our work is to enable an assessment of the efficacy of artifact evaluations as they have been implemented in software engineering and programming language conferences and to identify possible improvements for these processes. Such an assessment requires criteria according to which we can judge whether artifact evaluations meet their

1252

DISCLAIMER
This talk will not be specifically about this paper, but will contain several results and insights.

A Bit of History

Humble Beginnings

Sharing is in our nature

People shared data before artifact evaluation

No archiving: Lots of personal and institute websites

PROMISE repository

See: Tim Menzies' MSR'17 Award Acceptance Page

Official Pilot

ESEC/FSE 2011 - Szeged, Hungary

First official artifact evaluation

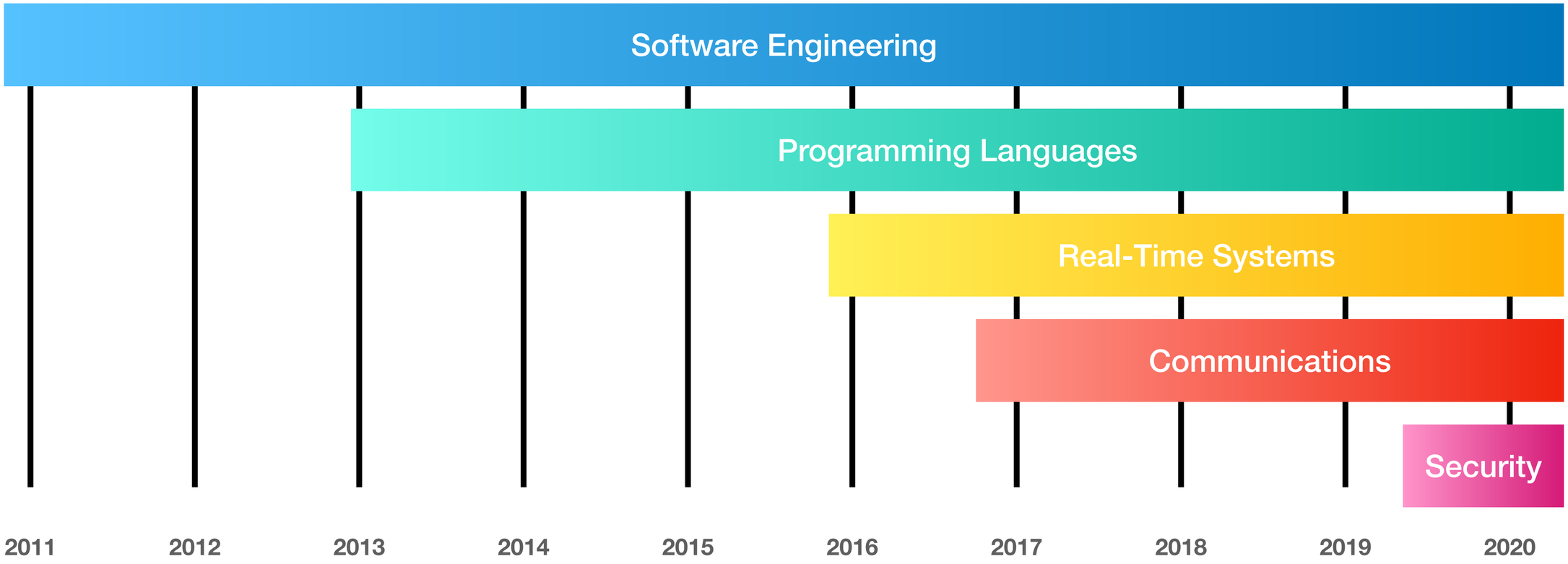
Andreas Zeller asked Carlo Ghezzi and Shriram Krishnamurthi to run AE in 2011

> An aside on naming. Shriram had long wanted to create such a committee and call it the "Program Committee" (ha, ha). However, not only is that name taken, we also wanted to be open-minded to all sorts of artifacts that are not programs (not only models but also data sets, etc.). We therefore called this the Artifact Evaluation Committee (AEC). We hoped that someone would come up with a better name for this eventually, but that has yet to happen while this name seems to be increasingly entrenched.

See: https://www.artifact-eval.org/motivation.html
https://bhermann.github.io/artifact-survey/calls/fse/2011.htm

Adoption

A few examples



Procedural Status Quo

What most committees do…

Paper Acceptance is the prerequisite

Artifact Submission Deadline (around 7 days later)

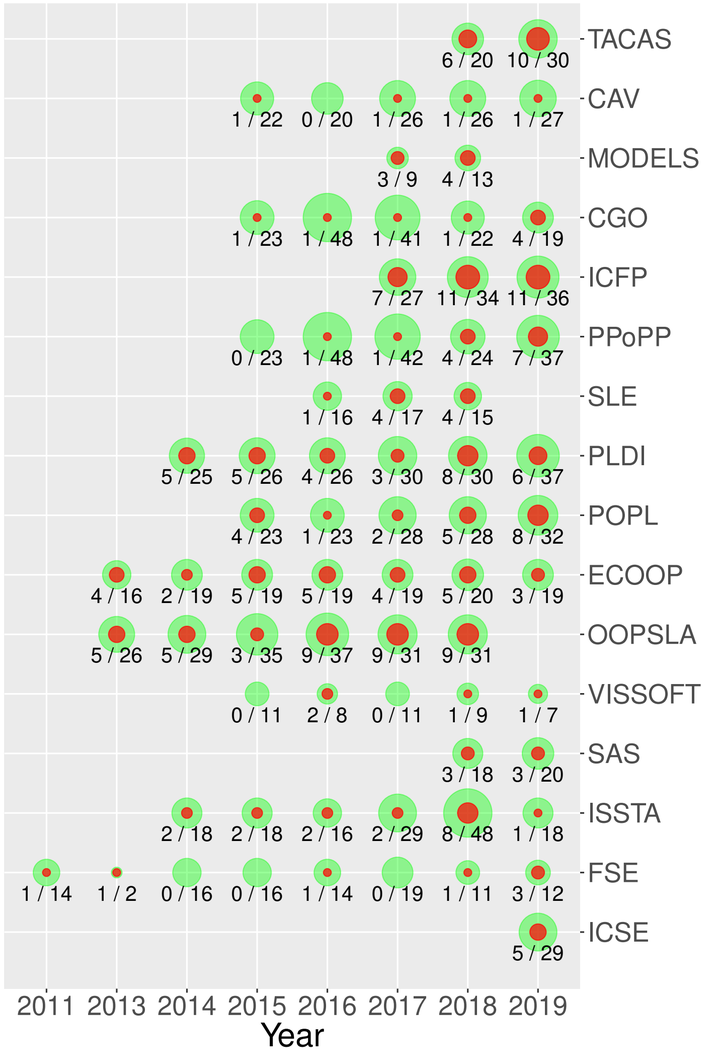
Artifact Evaluation does not influence paper acceptance

As a voluntary service participation rates differ

Some conferences move towards mandatory submissions

Community Effort

Over 1,000 people reviewed artifacts so far in SE and PL



See: Hermann et al. 2020 Preprint

Replication, Reproduction, Confusion

Terminology Evolution

Repeatability

Same team, same experimental setup

Reproducibility

Different team, different experimental setup

Replicability

Different team, same experimental setup

See: ACM Review and Badging Policy

Replication, Reproduction, Confusion

Terminology Evolution

Repeatability

Same team, same experimental setup

Reproducibility

Different team, same experimental setup

Replicability

Different team, different experimental setup

See: ACM Review and Badging Policy

Incentives

ACM Badges

A standard since 2018

There is an artifact
The artifact is archived

The artifact “works”

The artifact might be useful

A new artifact successfully replicated the results of this paper.

A new artifact successfully reproduced the results of this paper.

ACM Badges

A standard since 2018

There is an artifact
The artifact is archived

The artifact “works”

The artifact might be useful

A new artifact successfully replicated the results of this paper.

A new artifact successfully reproduced the results of this paper.

IEEE Badges?

No standard yet

Originally created by Matthias Hauswirth

Have been used in ACM venues
before the ACM badges

Open Science Badges

A peek into other publication cultures

Artifact Evaluation is just one incarnation of Open Science

We are not the only science that produces artifacts

Interesting new opportunity: Registered Reports

See: https://osf.io/
https://2020.msrconf.org/track/msr-2020-Registered-Reports
https://icsme2020.github.io/cfp/RegisteredReportsTrackCFP.html

Did you do it for the badge?

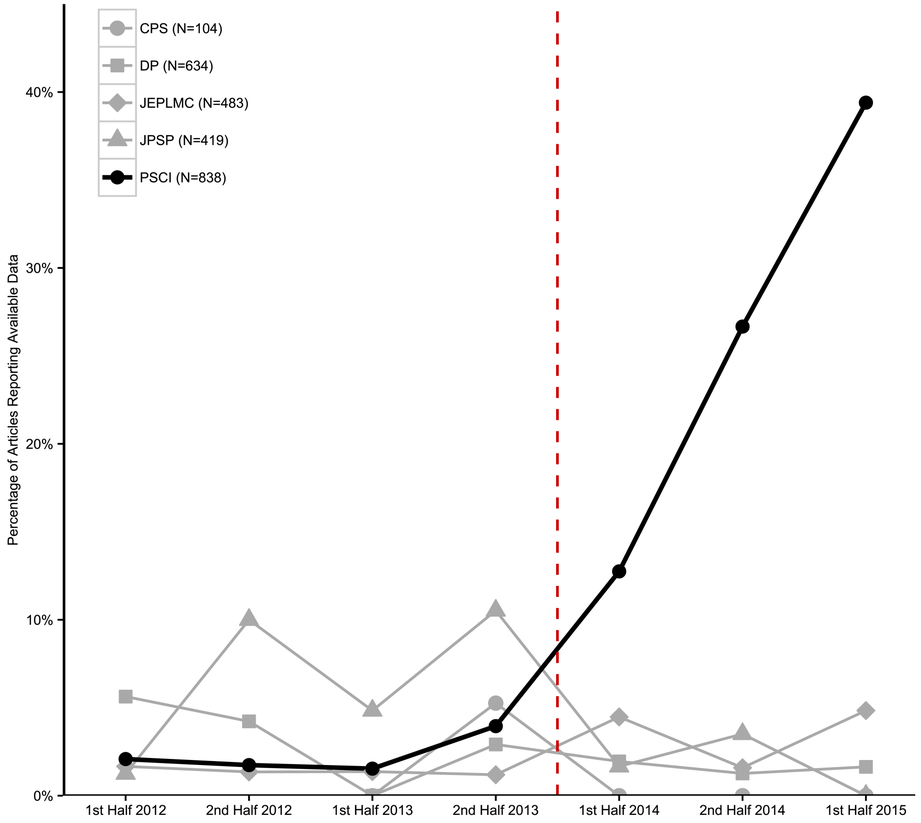
Psychology suggests you might have…

See: Kidwell et al. 2016

You can still have a paper without an artifact

But providing an artifact gives you the nice sign of approval on your paper

And it looks nice, doesn't it.



Boosting Your Bibliometrics?

The spice must flow

Good News: It ***might*** work!

Bad News: This is all very preliminary

Someone really should do a longitudinal study on this!

Noname manuscript No.
(will be inserted by the editor)

Publish or Perish,
but do not Forget your Software Artifacts

This is a post-peer-review, pre-copyedit version of an article published in Springer Emprical Software Engineering. The final authenticated version will soon be available online at http://dx.doi.org/10.1007/s10664-020-09851-6.

Robert Heumüller* · Sebastian Nielebock* · Jacob Krüger · Frank Ortmeier

Received: date / Accepted: date

Abstract Open-science initiatives have gained substantial momentum in computer science, and particularly in software-engineering research. A critical aspect of open-science is the public availability of artifacts (e.g., tools), which facilitates the replication, reproduction, extension, and verification of results. While we experienced that many artifacts are not publicly available, we are not aware of empirical evidence supporting this subjective claim. In this article, we report an empirical study on software artifact papers (SAPs) published at the International Conference on Software Engineering (ICSE), in which we investigated whether and how researchers have published their software artifacts, and whether this had scientific impact. Our dataset comprises 789 ICSE research track papers, including 604 SAPs (76.6 %), from the years 2007 to 2017. While showing a positive trend towards artifact availability, our results are still sobering. Even in 2017, only 58.5 % of the papers that stated to have developed a software artifact made that artifact publicly available. As we did

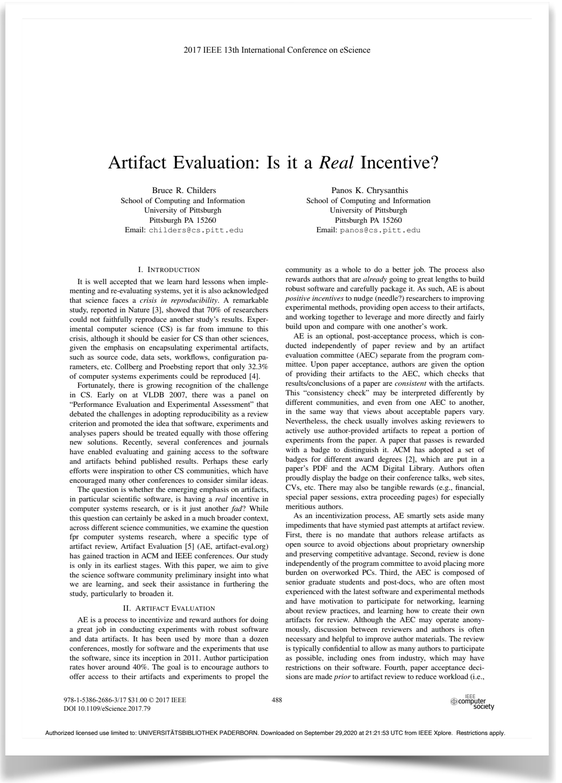
The work of Jacob Krüger has been supported by the German Research Foundation (SA 465/49-3) and an IFI fellowship of the German Academic Exchange Service.

* Both authors contributed equally to the research reported in this article.
Robert Heumüller
Otto-von-Guericke University Magdeburg, Germany
E-mail: robert.heumueller@ovgu.de

Sebastian Nielebock
Otto-von-Guericke University Magdeburg, Germany
E-mail: sebastian.nielebock@ovgu.de

Jacob Krüger
University of Toronto, Canada & Otto-von-Guericke University Magdeburg, Germany
E-mail: jacob.krueger@ovgu.de

Frank Ortmeier
Otto-von-Guericke University Magdeburg, Germany
E-mail: frank.ortmeier@ovgu.de



2017 IEEE 13th International Conference on eScience

Artifact Evaluation: Is it a *Real* Incentive?

Bruce R. Childers
School of Computing and Information
University of Pittsburgh
Pittsburgh PA 15260
Email: childers@cs.pitt.edu

Panos K. Chrysanthis
School of Computing and Information
University of Pittsburgh
Pittsburgh PA 15260
Email: panos@cs.pitt.edu

I. Introduction

It is well accepted that we learn hard lessons when implementing and re-evaluating systems, yet it is also acknowledged that science faces a *crisis in reproducibility*. A remarkable study, reported in Nature [3], showed that 70% of researchers could not faithfully reproduce another study's results. Experimental computer science (CS) is far from immune to this crisis, although it should be easier for CS than other sciences, given the emphasis on encapsulating experimental artifacts, such as source code, data sets, workflows, configuration parameters, etc. Collberg and Proebsing report that only 32.3% of computer systems experiments could be reproduced [4].

Fortunately, there is growing recognition of the challenge in CS. Early on at VLDB 2007, there was a panel on "Performance Evaluation and Experimental Assessment" that debated the challenges in adopting reproducibility as a review criterion and promoted the idea that software, experiments and analyses papers should be treated equally with those offering new solutions. Recently, several conferences and journals have enabled evaluating and gaining access to the software and artifacts behind published results. Perhaps these early efforts were inspiration to other CS communities, which have encouraged many other conferences to consider similar ideas.

The question is whether the emerging emphasis on artifacts, in particular scientific software, is having a *real* incentive in computer systems research, or is it just another *fad*? While this question can certainly be asked in a much broader context, across different science communities, we examine the question for computer systems research, where a specific type of artifact review, Artifact Evaluation [5] (AE, artifact-eval.org) has gained traction in ACM and IEEE conferences. Our study is only in its earliest stages. With this paper, we aim to give the science software community preliminary insight into what we are learning, and seek their assistance in furthering the study, particularly to broaden it.

II. Artifact Evaluation

AE is a process to incentivize and reward authors for doing a great job in conducting experiments with robust software and data artifacts. It has been used by more than a dozen conferences, mostly for software and the experiments that use the software, since its inception in 2011. Author participation rates hover around 40%. The goal is to encourage authors to offer access to their artifacts and experiments to propel the community as a whole to do a better job. The process also rewards authors that are *already* going to great lengths to build robust software and carefully package it. As such, AE is about *positive incentives* to nudge (needle?) researchers to improving experimental methods, providing open access to their artifacts, and working together to leverage and more directly and fairly build upon and compare with one another's work.

AE is an optional, post-acceptance process, which is conducted independently of paper review and by an artifact evaluation committee (AEC) separate from the program committee. Upon paper acceptance, authors are given the option of providing their artifacts to the AEC, which checks that results/conclusions of a paper are *consistent* with the artifacts. This "consistency check" may be interpreted differently by different communities, and even from one AEC to another, in the same way that views about acceptable papers vary. Nevertheless, the check usually involves asking reviewers to actively use author-provided artifacts to repeat a portion of experiments from the paper. A paper that passes is rewarded with a badge to distinguish it. ACM has adopted a set of badges for different award degrees [2], which are put in a paper's PDF and the ACM Digital Library. Authors often proudly display the badge on their conference talks, web sites, CVs, etc. There may also be tangible rewards (e.g., financial, special paper sessions, extra proceeding pages) for especially meritious authors.

As an incentivization process, AE smartly sets aside many impediments that have stymied past attempts at artifact review. First, there is no mandate that authors release artifacts as open source to avoid objections about proprietary ownership and preserving competitive advantage. Second, review is done independently of the program committee to avoid placing more burden on overworked PCs. Third, the AEC is composed of senior graduate students and post-docs, who are often most experienced with the latest software and experimental methods and have motivation to participate for networking, learning about review practices, and learning how to create their own artifacts for review. Although the AEC may operate anonymously, discussion between reviewers and authors is often necessary and helpful to improve author materials. The review is typically confidential to allow as many authors to participate as possible, including ones from industry, which may have restrictions on their software. Fourth, paper acceptance decisions are made *prior* to artifact review to reduce workload (i.e.,

978-1-5386-2686-3/17 $31.00 © 2017 IEEE
DOI 10.1109/eScience.2017.79

See: Heumüller et al. 2020 Preprint
Childers et al. 2017

Advancing Your Field?

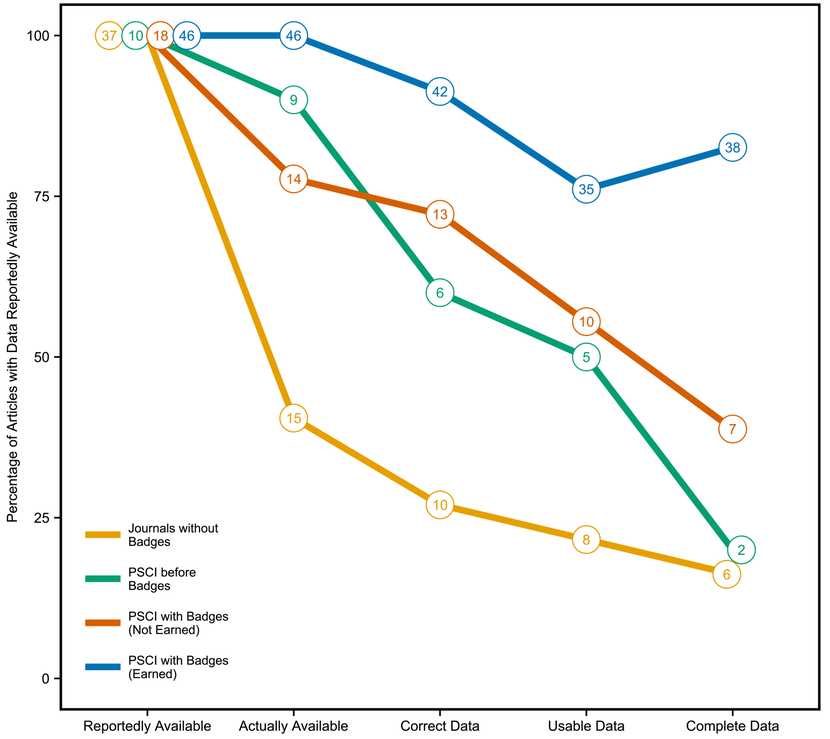
Are we getting better?

This is not new!

Several successful tools and datasets have inspired new research

But is artifact evaluation helping here?

We actually do a study on this!



Again: Kidwell et al. 2016

Brand Recognition?

Guess the Tool!

Brand Recognition?

Guess the Tool!

Three Dilemmas

The PhD Student Dilemma

Why you should care even if you leave academia

Hey congrats on your accepted paper! Can you upload the artifact on Monday when you come to the office?

Oh! I didn't know we want to submit an artifact too… I have nothing prepared for that, but I also doubt that anyone would want it anyway. What do you think?

You never know that before! Reuse happens after graduation!

You SRSLY think so? Well, I get on it next week.

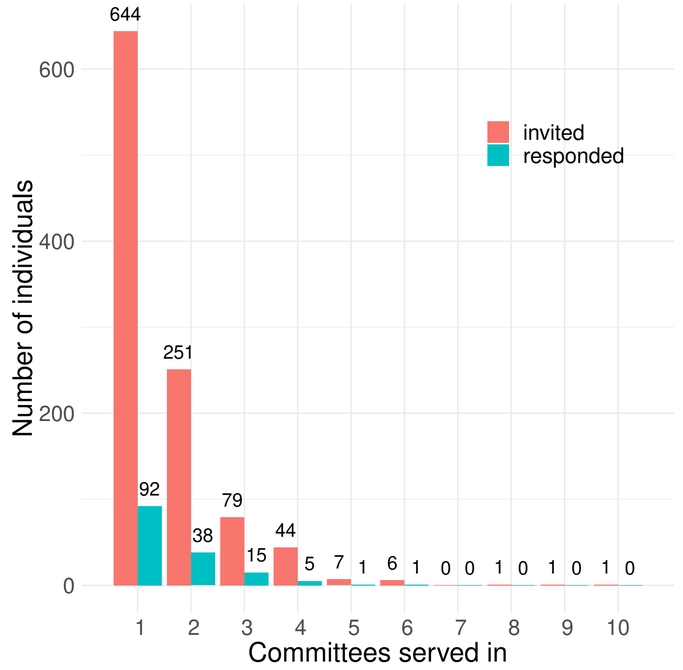
AECs are Junior Researchers

You only serve once…

62% of all artifact reviewers only serve once

34% of reviewers report to have never submitted an artifact

You cannot run an AEC as you would run a PC



See: Hermann et al. 2020 Preprint

The Reviewer Dilemma

A plea for bigger committees and more recognition

Remember the goal of AE: Replicability

Now imagine you need to assess this for several artifacts within three weeks

And the artifact is quite cryptic and its experiments take a long time

And your supervisor says you should be writing

But in the end: The paper is already accepted

What is your most likely behavior?

See: Moritz Beller's Experience Report

The Quality Dilemma

What should our goal a community be?

Many Available Artifacts

Less, but High-Quality Artifacts

The Perfect Artifact

Archival

Providing your artifact long after you left academia

Very cool GitHub integration!

Archive with a DOI

GitHub has no retention policy

Versioning

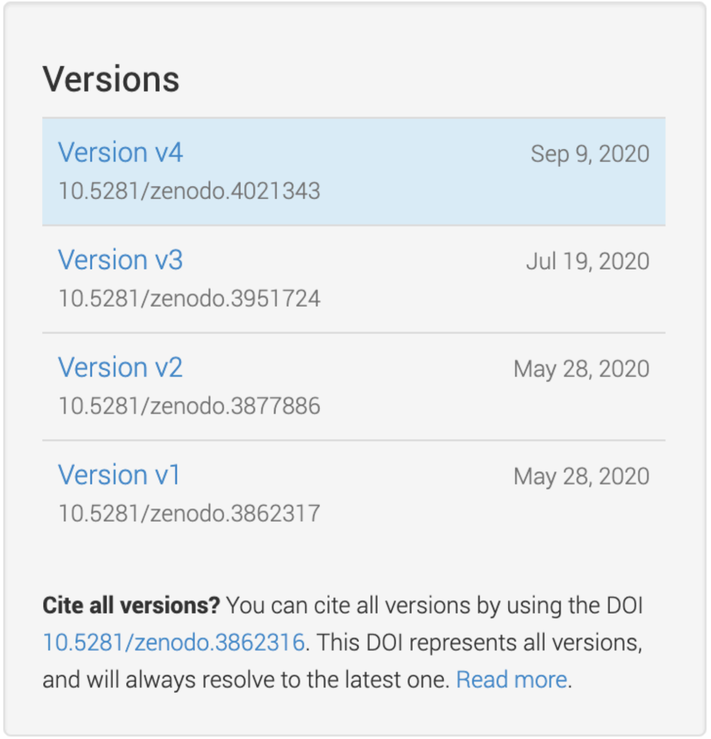
… an underrated feature

Artifacts can be improved over time

References in the paper can be outdated

Versioning enables hints for your future readers

You can include things your reviewers found



Passing Evaluation

Planning artifacts ahead of time

Package your automated experiments while creating them

Document your co-authors artifacts, not your own
Also, do this while testing them

Think of reviewers that don't have that amazing experiment cloud you have access to - Subsetting helps

Passing evaluation is not your goal

Passing evaluation should come naturally

Long Term Success

And Its Costs

Maintaining a successful artifact / tool / project is a tough job

You need time, people, nerves, funding, …

Only maintain your champions
not your pet projects

Summary

… in shiny boxes, text, and pictures

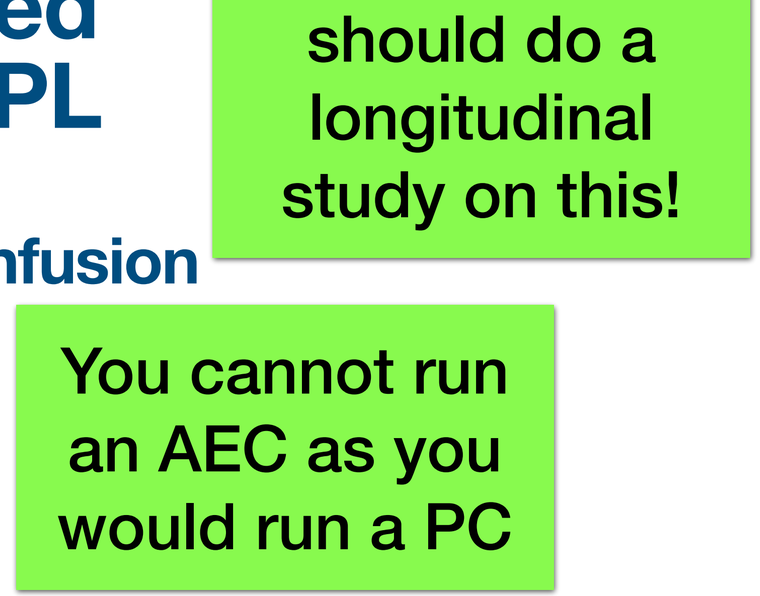
Over 1,000 people reviewed artifacts so far in SE and PL

Replication, Reproduction, Confusion

The Quality Dilemma

GitHub has no retention policy

Someone really should do a longitudinal study on this!



You cannot run an AEC as you would run a PC

Passing evaluation is not your goal

You never know that before! Reuse happens after graduation!