Reflections on Artifact Evaluation

Eric Eide
University of Utah

Artifact evaluation

doi> 10.1145/2658987

viewpoints

DOI:10.1145/2658987

Shriram Krishnamurthi and Jan Vitek

Viewpoint

The Real Software Crisis: Repeatability as a Core Value

Sharing experiences running artifact evaluation committees for five major conferences.

WHERE IS THE software in programming language research? In our field, software artifacts play a central role: they are the embodiments of our ideas and contributions. Yet when we publish, we are evaluated on our ability to describe informally those artifacts in prose. Of-

If a paper makes, or implies, claims that require software, those claims

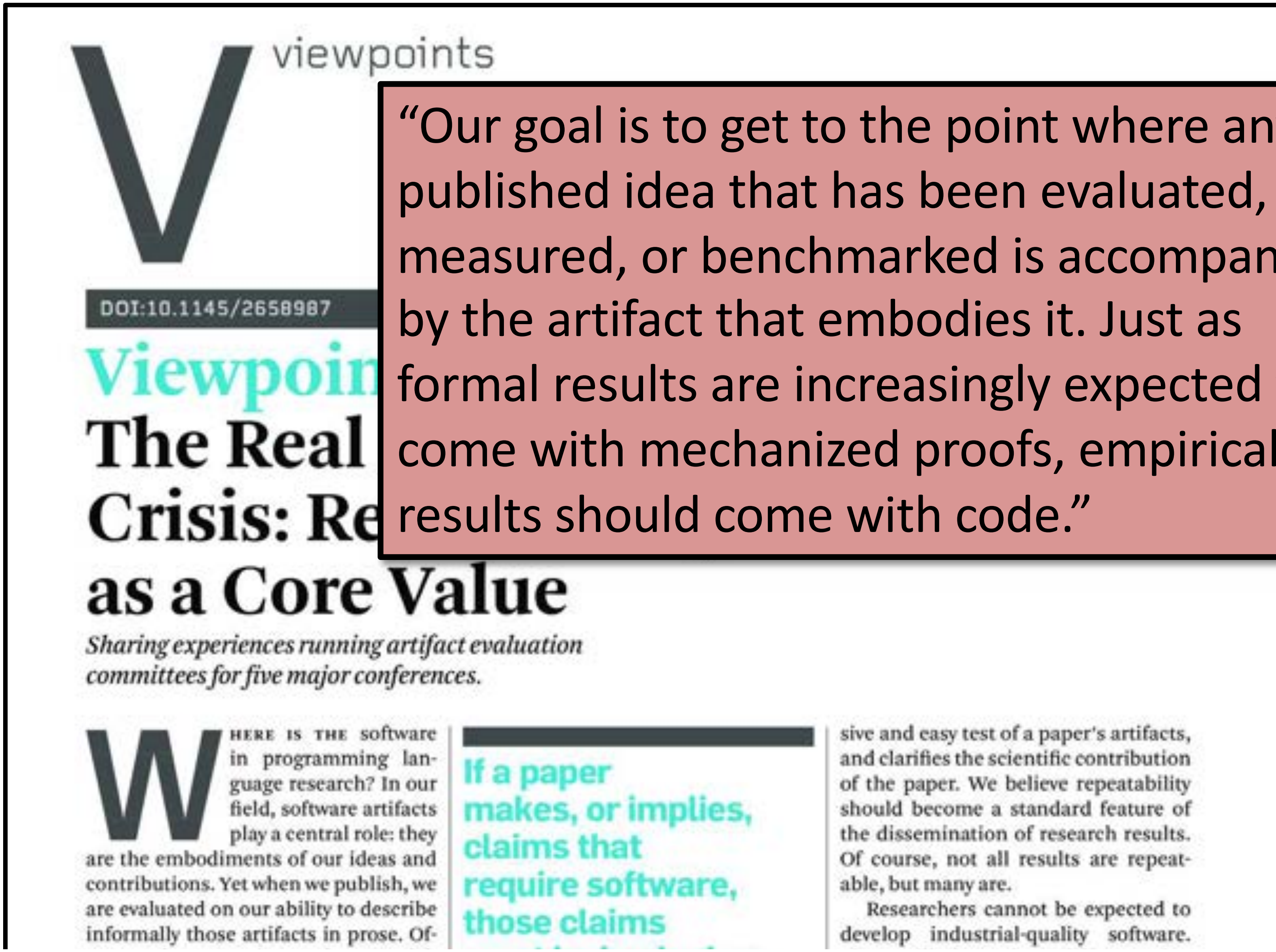
sive and easy test of a paper's artifacts, and clarifies the scientific contribution of the paper. We believe repeatability should become a standard feature of the dissemination of research results. Of course, not all results are repeatable, but many are.

Researchers cannot be expected to develop industrial-quality software.

Artifact evaluation

doi> 10.1145/2658987

"Our goal is to get to the point where any published idea that has been evaluated, measured, or benchmarked is accompanied by the artifact that embodies it. Just as formal results are increasingly expected to come with mechanized proofs, empirical results should come with code."



V viewpoints

DOI:10.1145/2658987

Viewpoin

The Real
Crisis: Re
as a Core Value

Sharing experiences running artifact evaluation committees for five major conferences.

WHERE IS THE software in programming language research? In our field, software artifacts play a central role: they are the embodiments of our ideas and contributions. Yet when we publish, we are evaluated on our ability to describe informally those artifacts in prose. Of-

If a paper makes, or implies, claims that require software, those claims

sive and easy test of a paper's artifacts, and clarifies the scientific contribution of the paper. We believe repeatability should become a standard feature of the dissemination of research results. Of course, not all results are repeatable, but many are.

Researchers cannot be expected to develop industrial-quality software.

Why did we choose that goal?

Why reproduce?

Producing Wrong Data Without Doing Anything Obviously Wrong!

Todd Mytkowicz Amer Diwan
Department of Computer Science
University of Colorado
Boulder, CO, USA
{mytkowit,diwan}@colorado.edu

Matthias Hauswirth
Faculty of Informatics
University of Lugano
Lugano, CH
Matthias.Hauswirth@unisi.ch

Peter F. Sweeney
IBM Research
Hawthorne, NY, USA
pfs@us.ibm.com

Abstract

This paper presents a surprising result: changing a seemingly innocuous aspect of an experimental setup can cause a systems researcher to draw wrong conclusions from an experiment. What appears to be an innocuous aspect in the experimental setup may in fact introduce a significant bias in an evaluation. This phenomenon is called *measurement bias* in the natural and social sciences.

Our results demonstrate that measurement bias is significant and commonplace in computer system evaluation. By *significant* we mean that measurement bias can lead to a performance analysis that either over-states an effect or even yields an incorrect conclusion. By *commonplace* we mean that measurement bias occurs in all architectures that we tried (Pentium 4, Core 2, and m5 O3CPU), both compilers that we tried (gcc and Intel's C compiler), and most of the SPEC CPU2006 C programs. Thus, we cannot ignore measurement bias. Nevertheless, in a literature survey of 133 recent papers from ASPLOS, PACT, PLDI, and CGO, we de-

1. Introduction

Systems researchers often use experiments to drive their work: they use experiments to identify bottlenecks and then again to determine if their optimizations for addressing the bottlenecks are effective. If the experiment is biased then a researcher may draw an incorrect conclusion: she may end up wasting time on something that is not really a problem and may conclude that her optimization is beneficial even when it is not.

We show that experimental setups are often biased. For example, consider a researcher who wants to determine if optimization O is beneficial for system S. If she measures S and $S + O$ in an experimental setup that favors $S + O$, she may overstate the effect of O or even conclude that O is beneficial even when it is not. This phenomenon is called *measurement bias* in the natural and social sciences. This paper shows that measurement bias is commonplace and significant: it can easily lead to a performance analysis that yields incorrect conclusions.

doi> 10.1145/1508244.1508275

Why reproduce?

Producing Wrong Data Without Doing Anything Obviously Wrong!

"This paper presents a surprising result: changing a seemingly innocuous aspect of an experimental setup can cause a systems researcher to draw wrong conclusions from an experiment. What appears to be an innocuous aspect in the experimental setup may in fact introduce a significant bias in an evaluation."

Our results demonstrate that measurement bias is significant and commonplace in computer system evaluation. By *significant* we mean that measurement bias can lead to a performance analysis that either over-states an effect or even yields an incorrect conclusion. By *commonplace* we mean that measurement bias occurs in all architectures that we tried (Pentium 4, Core 2, and m5 O3CPU), both compilers that we tried (gcc and Intel's C compiler), and most of the SPEC CPU2006 C programs. Thus, we cannot ignore measurement bias. Nevertheless, in a literature survey of 133 recent papers from ASPLOS, PACT, PLDI, and CGO, we de-

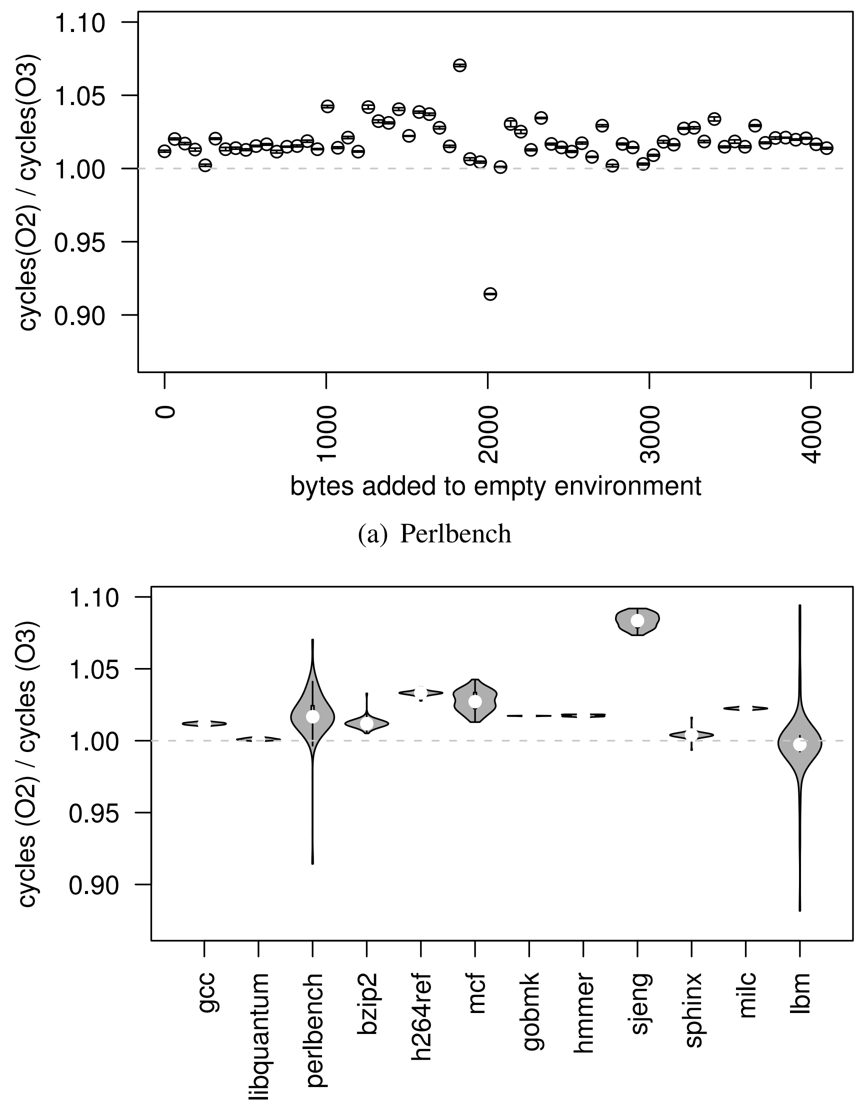
...when it is not.

We show that experimental setups are often biased. For example, consider a researcher who wants to determine if optimization O is beneficial for system S. If she measures S and $S + O$ in an experimental setup that favors $S + O$, she may overstate the effect of O or even conclude that O is beneficial even when it is not. This phenomenon is called *measurement bias* in the natural and social sciences. This paper shows that measurement bias is commonplace and significant: it can easily lead to a performance analysis that yields incorrect conclusions.

doi> 10.1145/1508244.1508275

Myktowicz et al.

- is –O3 optimization beneficial over –O2?
- compile and run SPEC 2006 benchmarks
  - vary size of environment
  - vary link order
- result: performance of benchmarks varied widely



(a) Perlbench

(b) All Benchmarks

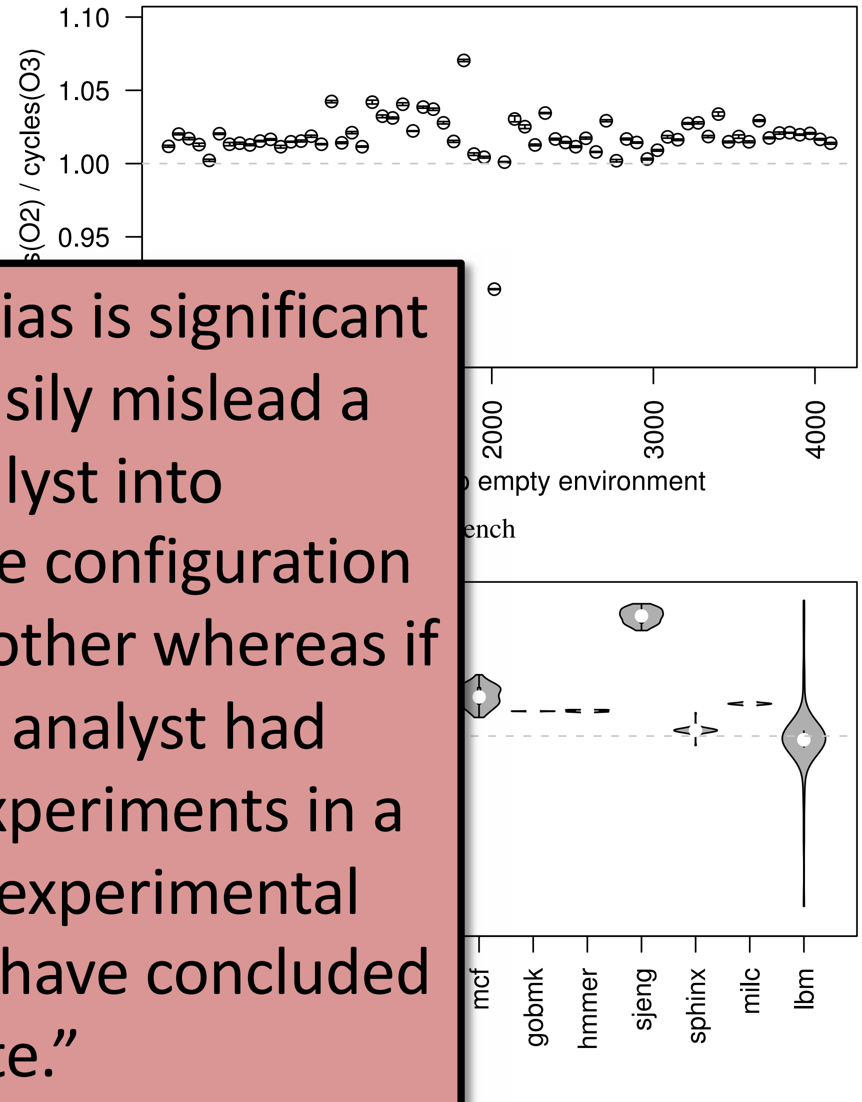
Figure 3. The effect of UNIX environment size on the speedup of *O3* on Core 2.

Figure credit: Myktowicz et al., doi> 10.1145/1508244.1508275

Myktowicz et al.

- is –O3 optim
  beneficial ov
- compile and
  2006 bench
  - vary size of
  - vary link or
- result: perfo
  benchmarks varied widely

“Measurement bias is significant because it can easily mislead a performance analyst into believing that one configuration is better than another whereas if the performance analyst had conducted the experiments in a slightly different experimental setup she would have concluded the exact opposite.”



(b) All Benchmarks

Figure 3. The effect of UNIX environment size on the speedup of *O3* on Core 2.

Figure credit: Myktowicz et al., doi> 10.1145/1508244.1508275

Improving CS research

R^3 – Repeatability, Reproducibility and Rigor

Jan Vitek
Purdue University, USA

Tomas Kalibera
University of Kent, UK

Abstract

Computer systems research spans sub-disciplines that include embedded systems, programming languages and compilers, networking, and operating systems. Our contention is that a number of structural factors inhibit quality systems research. We highlight some of the factors we have encountered in our own work and observed in published papers and propose solutions that could both increase the productivity of researchers and the quality of their output.

1. Introduction

> "One of the students told me she wanted to do an experiment that went something like this ... under certain circumstances, X, rats did something, A. She was curious as to whether, if she changed the circumstances to Y, they would still do, A. So her proposal was to do the experiment under circumstances Y and see if they still did A. I explained to her that it was necessary first to repeat in her laboratory the experiment of the other person — to do it under condition X to see if she could also get result A — and then change to Y and see if A changed. Then she would know that the real difference was the thing she thought she had under control. She was very delighted with this new idea, and went to her professor. And his reply was, no, you cannot do that, because the experiment has already been done and

The essence of the scientific process consists of (*a*) positing a hypothesis or model, (*b*) engineering a concrete implementation, and (*c*) designing and conducting an experimental evaluation. What is the value of an unevaluated claim? How much work is needed to truly validate a claim? What is reasonable to expect in a paper? Given the death march of our field towards publication it is not realistic to expect much. Evaluating a non-trivial idea is beyond the time budget of any single paper as this requires running many benchmarks on multiple implementations with different hardware and software platforms. Often a careful comparison to the state of the art means implementing competing solutions. The result of this state of affairs is that papers presenting potentially useful novel ideas regularly appear without a comparison to the state of the art, without appropriate benchmarks, without any mention of limitations, and without sufficient detail to reproduce the experiments. This hampers scientific progress and perpetuates the cycle.

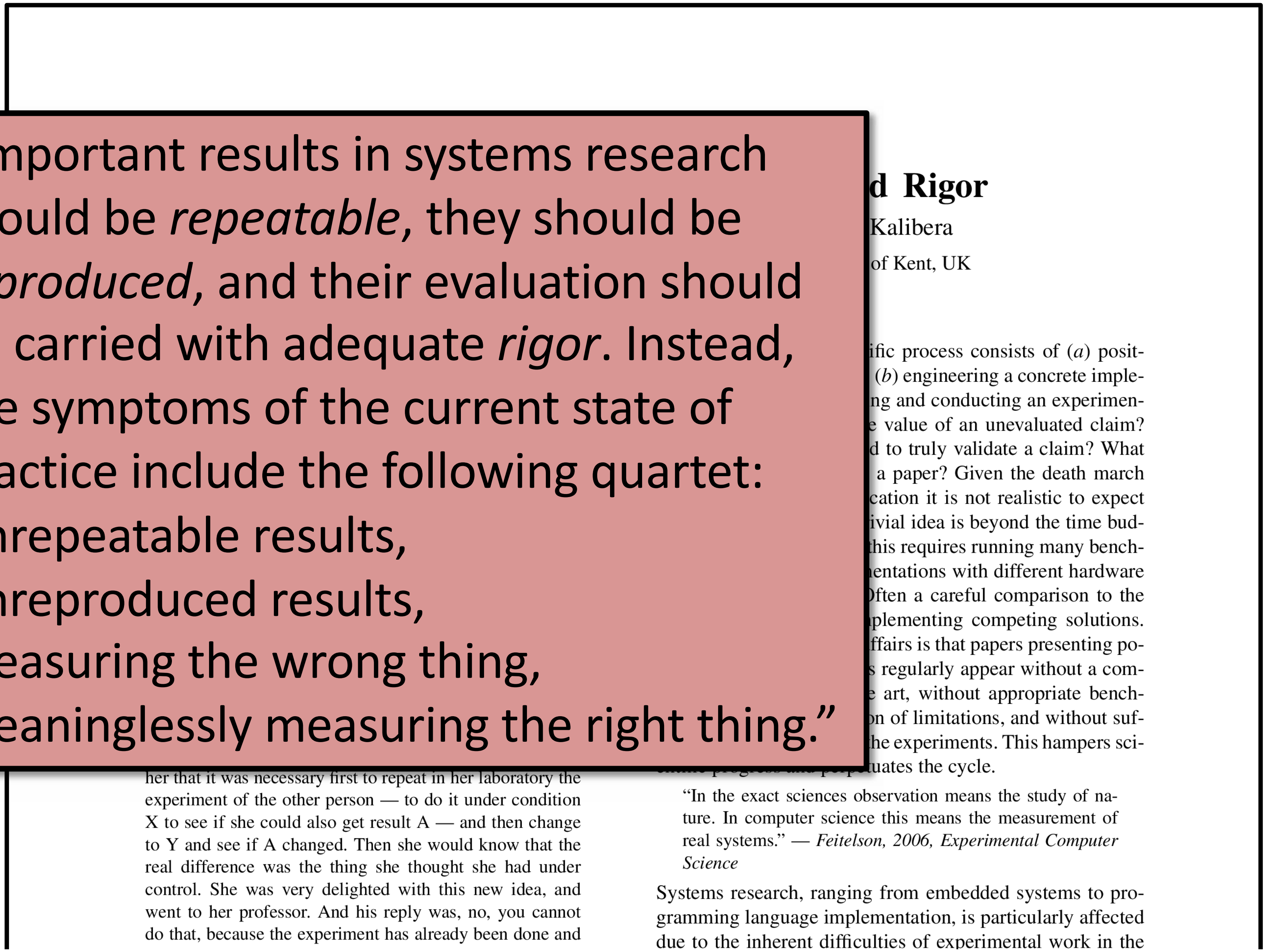
> "In the exact sciences observation means the study of nature. In computer science this means the measurement of real systems." — *Feitelson, 2006, Experimental Computer Science*

Systems research, ranging from embedded systems to programming language implementation, is particularly affected due to the inherent difficulties of experimental work in the

doi> 10.1145/2442776.2442781

Improving CS research

“Important results in systems research should be *repeatable*, they should be *reproduced*, and their evaluation should be carried with adequate *rigor*. Instead, the symptoms of the current state of practice include the following quartet:
Unrepeatable results,
Unreproduced results,
Measuring the wrong thing,
Meaninglessly measuring the right thing.”


d Rigor

Kalibera

of Kent, UK

ific process consists of (*a*) posit-
(*b*) engineering a concrete imple-
ng and conducting an experimen-
e value of an unevaluated claim?
d to truly validate a claim? What
a paper? Given the death march
cation it is not realistic to expect
ivial idea is beyond the time bud-
his requires running many bench-
entations with different hardware
Often a careful comparison to the
plementing competing solutions.
ffairs is that papers presenting po-
s regularly appear without a com-
e art, without appropriate bench-
on of limitations, and without suf-
he experiments. This hampers sci-
uates the cycle.

her that it was necessary first to repeat in her laboratory the experiment of the other person — to do it under condition X to see if she could also get result A — and then change to Y and see if A changed. Then she would know that the real difference was the thing she thought she had under control. She was very delighted with this new idea, and went to her professor. And his reply was, no, you cannot do that, because the experiment has already been done and

“In the exact sciences observation means the study of nature. In computer science this means the measurement of real systems.” — *Feitelson, 2006, Experimental Computer Science*

Systems research, ranging from embedded systems to programming language implementation, is particularly affected due to the inherent difficulties of experimental work in the

doi> 10.1145/2442776.2442781

Vitek and Kalibera

“Deadly Sins”

- unclear goals
- implicit assumptions
- proprietary data
- weak statistics
- meaningless measurements
- no baseline
- unrepresentative workloads

Recommendations

- develop open-source benchmarks
- codify best-practice documentation, methodologies, and reporting standards
- require repeatability of published results
- encourage reproduction studies

Reexamining previous results

2016 IEEE/ACM 38th IEEE International Conference on Software Engineering

On the Techniques We Create, the Tools We Build, and Their Misalignments: a Study of KLEE

Eric F. Rizzi,
Grammatech Inc., Ithaca, NY, USA
{erizzi}@grammatech.com

Sebastian Elbaum, Matthew B. Dwyer
University of Nebraska - Lincoln, USA
{elbaum,dwyer}@cse.unl.edu

ABSTRACT

Our community constantly pushes the state-of-the-art by introducing "new" techniques. These techniques often build on top of, and are compared against, existing systems that realize previously published techniques. The underlying assumption is that existing systems correctly represent the techniques they implement. This paper examines that assumption through a study of KLEE, a popular and well-cited tool in our community. We briefly describe six improvements we made to KLEE, none of which can be considered "new" techniques, that provide order-of-magnitude performance gains. Given these improvements, we then investigate how the results and conclusions of a sample of papers that cite KLEE are affected. Our findings indicate that the strong emphasis on introducing "new" techniques may lead to wasted effort, missed opportunities for progress, an accretion of artifact complexity, and questionable research conclusions (in our study, 27% of the papers that depend on KLEE can be questioned). We conclude by revisiting initiatives that may help to realign the incentives to better support the foundations on which we build.

CCS Concepts

•General and reference → Empirical studies; •Software and its engineering → Software libraries and repositories: Software

been lost as a priority. We contend that the software engineering research community is worse for this.

The focus on discovery leads much of the research published in software engineering to make claims of the form "technique A is the new state-of-the-art". To support such claims it is very common to manifest a technique in the implementation of a software system. Every year there are papers in major conferences and journals reporting evaluations using, for example, test generators, program analyzers, refactoring tools, program comprehension systems, fault localizers, recommendation systems, and user interfaces. As a community, we rely on the fidelity and quality of these implementations to support conclusions we draw about the techniques that they realize, but it is notoriously difficult to distinguish discovery from mistaken or sub-optimal implementation [73].

Demonstrating the value of technique A may involve direct comparison with, or building on top of, technique B. In either case, the implementations of A and B play a crucial role in the validity of the conclusions that can be drawn about A. Inadequacies in those implementations can lead to different kinds of problems. A faulty implementation of B may lead to *invalid conclusions* about the value of A. Researchers may *waste effort* in creating a new technique, A, because of a perceived inadequacy in B, but that inadequacy may simply be a fault in the implementation of B. Faults or limitations in

doi> 10.1145/2884781.2884835

Reexamining previous results

2016 IEEE/ACM 38th IEEE International Conference on Software Engineering

On the Techniques We Create, the Tools We Build, and Their Misalignments: a Study of KLEE

"We briefly describe six improvements we made to KLEE… Given these improvements, we then investigate how the results and conclusions of a sample of papers that cite KLEE are affected. Our findings indicate that the strong emphasis on introducing 'new' techniques may lead to… questionable research conclusions (in our study, 27% of the papers that depend on KLEE can be questioned)."

CCS Concepts

•General and reference → Empirical studies; •Software and its engineering → Software libraries and repositories; Software

of A. Researchers may *waste effort* in creating a new technique, A, because of a perceived inadequacy in B, but that inadequacy may simply be a fault in the implementation of B. Faults or limitations in

doi> 10.1145/2884781.2884835

Rizzi et al.

"Of the 25 papers whose results could be affected by our KLEE improvements, our analysis identified 11... that required a deeper examination because we deemed that their conclusions could be significantly affected. This deeper examination consisted not just in analyzing the papers in more detail, but also attempting to replicate studies. **In spite of our efforts, we were able to replicate the studies in only two of these papers.**"

(Emphasis mine.)

Rizzi et al.

"Of the 25 papers whose results could be affected by our KLEE improvements, our analysis identified 11... that required a deeper examination because
we de
signif
consi
detail
spite
studi

"We were able to replicate HMP-19 which contained a reference to an **online-repository, with all of the necessary code and data.** The other paper we were eventually able to replicate with the authors' assistance was IA-20, although as we shall see even in this instance the result of the replication did not quite match those in the paper."

(Emphasis mine.)

Rizzi et al.

"Of the 25 papers whose results could be affected by our KLEE improvements, our analysis identified 11... that required a deeper examination because we de
signif
consi
detail
spite
studi

"We were able to replicate HMP-19 which contained a reference to an **online-repository, with all of the necessary code and data.** The other paper we were eventua
assistan
in this in
quite ma

"We did not receive a response for two papers, while for the remainder we were informed that **pending patents, work with industrial bodies, or unrecoverable code and data prevented the authors from being able to help us replicate their experiments.**"

(Emphasis mine.)

Surveying a field

doi> 10.1145/2996358

*droid: Assessment and Evaluation of Android Application Analysis Tools

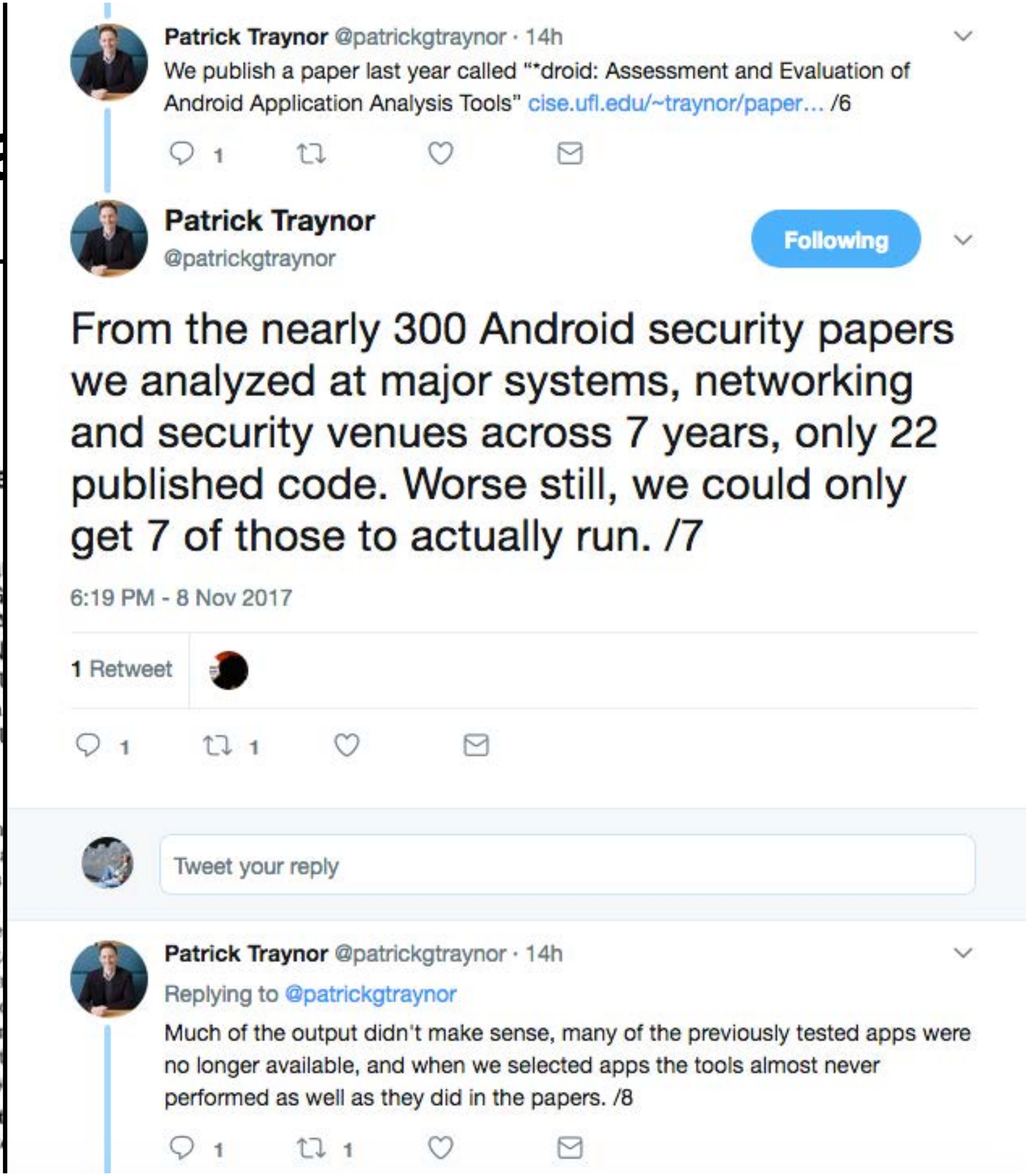
BRADLEY REAVES and JASMINE BOWERS, University of Florida
SIGMUND ALBERT GORSKI III, North Carolina State University
OLABODE ANISE, RAHUL BOBHATE, RAYMOND CHO, HIRANAVA DAS, SHARIQUE HUSSAIN, HAMZA KARACHIWALA, NOLEN SCAIFE, BYRON WRIGHT, and KEVIN BUTLER, University of Florida
WILLIAM ENCK, North Carolina State University
PATRICK TRAYNOR, University of Florida

The security research community has invested significant effort in improving the security of Android applications over the past half decade. This effort has addressed a wide range of problems and resulted in the creation of many tools for application analysis. In this article, we perform the first systematization of Android security research that analyzes applications, characterizing the work published in more than 17 top venues since 2010. We categorize each paper by the types of problems they solve, highlight areas that have received the most attention, and note whether tools were ever publicly released for each effort. Of the released tools, we then evaluate a representative sample to determine how well application developers can apply the results of our community's efforts to improve their products. We find not only that significant work remains to be done in terms of research coverage but also that the tools suffer from significant issues ranging from lack of maintenance to the inability to produce functional output for applications with known vulnerabilities. We close by offering suggestions on how the community can more successfully move forward.

CCS Concepts: • **Security and privacy → Software and application security;** • **Software and its engineering → Automated static analysis; Dynamic analysis;**

Surveying a

doi> 10.1145/2996358



Improving the situation: Artifact evaluation

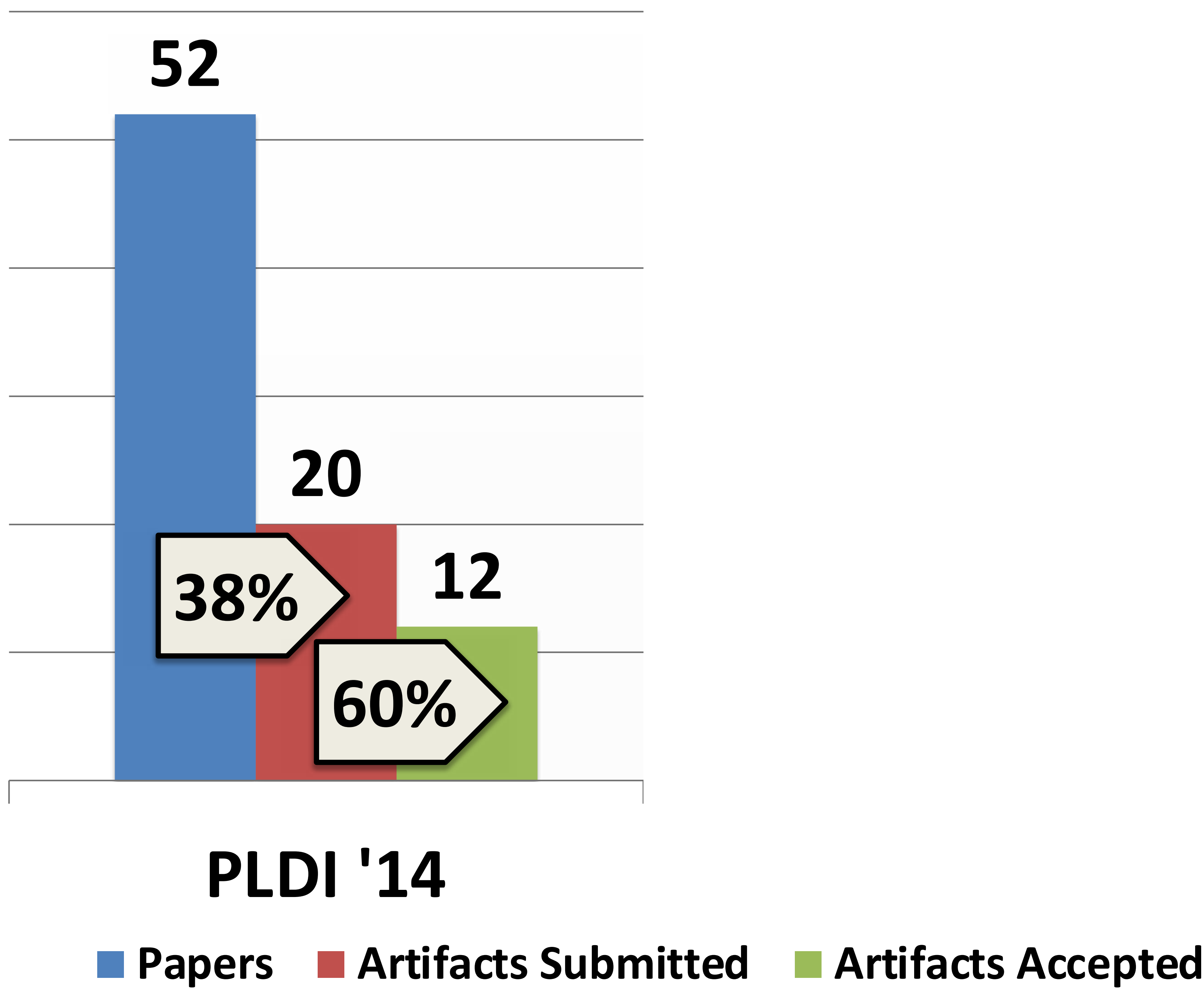
Artifact evaluation: why? how?

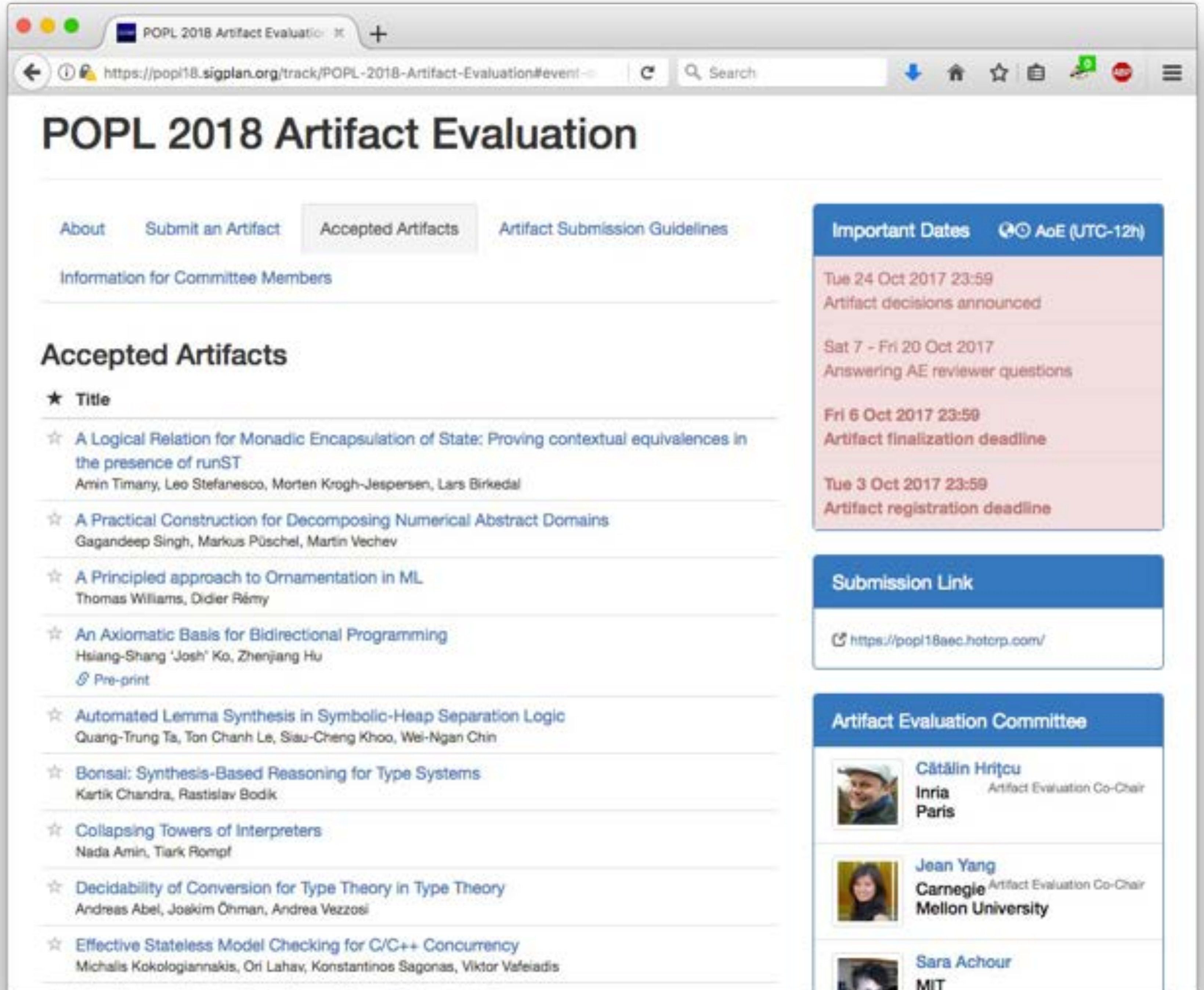
- recognize authors who create useful artifacts
- improve papers through artifact availability & review
- *a first step toward repeatability as a review criterion*

- authors of accepted papers invited to submit artifacts
  - due shortly after paper acceptance

- artifacts reviewed by a separate Artifact Evaluation Committee

- **“Does the artifact meet the expectations set by its paper?”**


52
20
12
38%
60%
PLDI '14
Papers
Artifacts Submitted
Artifacts Accepted


POPL 2018 Artifact Evaluation
https://popl18.sigplan.org/track/POPL-2018-Artifact-Evaluation#event-

POPL 2018 Artifact Evaluation

About | Submit an Artifact | Accepted Artifacts | Artifact Submission Guidelines | Information for Committee Members

Accepted Artifacts

★ Title

☆ A Logical Relation for Monadic Encapsulation of State: Proving contextual equivalences in the presence of runST
Amin Timany, Leo Stefanesco, Morten Krogh-Jespersen, Lars Birkedal

☆ A Practical Construction for Decomposing Numerical Abstract Domains
Gagandeep Singh, Markus Püschel, Martin Vechev

☆ A Principled approach to Ornamentation in ML
Thomas Williams, Didier Rémy

☆ An Axiomatic Basis for Bidirectional Programming
Hsiang-Shang 'Josh' Ko, Zhenjiang Hu
Pre-print

☆ Automated Lemma Synthesis in Symbolic-Heap Separation Logic
Quang-Trung Ta, Ton Chanh Le, Siau-Cheng Khoo, Wei-Ngan Chin

☆ Bonsai: Synthesis-Based Reasoning for Type Systems
Kartik Chandra, Rastislav Bodik

☆ Collapsing Towers of Interpreters
Nada Amin, Tiark Rompf

☆ Decidability of Conversion for Type Theory in Type Theory
Andreas Abel, Joakim Öhman, Andrea Vezzosi

☆ Effective Stateless Model Checking for C/C++ Concurrency
Michalis Kokologiannakis, Ori Lahav, Konstantinos Sagonas, Viktor Vafeiadis

Important Dates AoE (UTC-12h)

Tue 24 Oct 2017 23:59
Artifact decisions announced

Sat 7 - Fri 20 Oct 2017
Answering AE reviewer questions

Fri 6 Oct 2017 23:59
Artifact finalization deadline

Tue 3 Oct 2017 23:59
Artifact registration deadline

Submission Link

https://popl18aec.hotcrp.com/

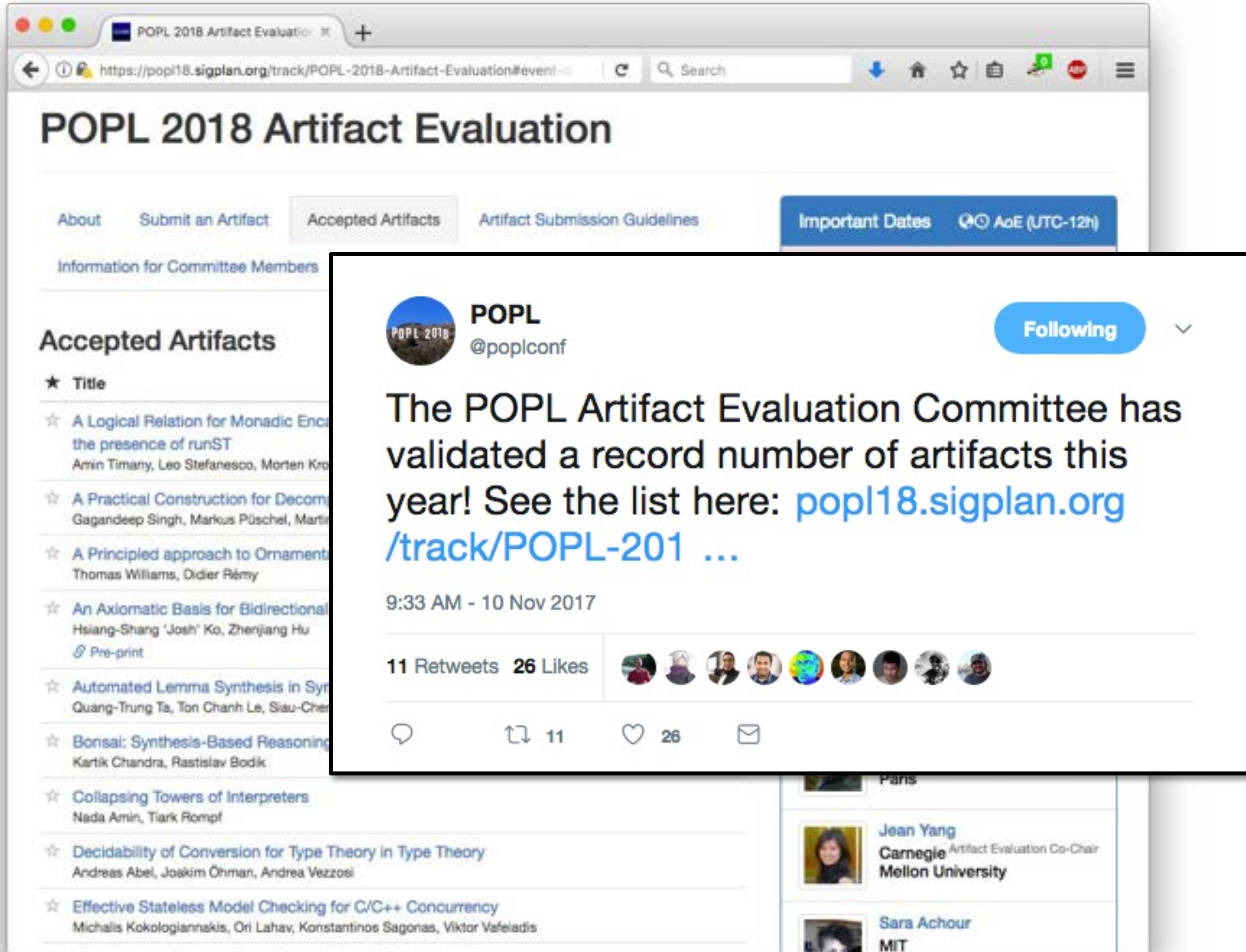
Artifact Evaluation Committee

Cătălin Hriţcu
Inria Paris
Artifact Evaluation Co-Chair

Jean Yang
Carnegie Mellon University
Artifact Evaluation Co-Chair

Sara Achour
MIT

https://popl18.sigplan.org/track/POPL-2018-Artifact-Evaluation https://popl18.sigplan.org/track/POPL-2018-Artifact-Evaluation




ACM SIGMOD 2017 Most Rep

https://sigmod.org/2017-reproducibility-award/

Search

Home About Membership PODS Publications History Software

ACM SIGMOD 2017 Most Reproducible Paper Award Winners

This award recognizes the best papers in terms of reproducibility. The three most reproducible papers are picked every year and the awards are presented during the awards session of the SIGMOD conference (next year). Each award comes with a 750$ honorarium sponsored by IBM.

The criteria are as follows: (i) coverage (ideal: all results can be verified), (ii) ease of reproducibility (ideal: just works), (iii) flexibility (ideal: can change workloads, queries, data and get similar behavior with published results), and (iv) portability (ideal: linux, mac, windows).

Winners of 2017

Awarded to *Most Reproducible Papers* of ACM SIGMOD 2016.

Generating Preview Tables for Entity Graphs

by Ning Yan, Sona Hasani, Abolfazl Asudeh, Chengkai Li

Verified by: Hideaki Kimura

ACM: Result & artifact badging

doi> 10.1145/2994031

from the publications board

DOI:10.1145/2994031

Ronald F. Boisvert

Incentivizing Reproducibility

A SCIENTIFIC RESULT is not truly established until it is independently confirmed. This is one of the tenets of experimental science. Yet, we have seen a rash of recent headlines about experimental results that could not be reproduced. In the biomedical field, efforts to reproduce results of academic research by drug companies have had less than a 50% success rate,[a] resulting in billions of dollars in wasted effort.[b] In most cases the cause is not intentional fraud, but rather sloppy research protocols and faulty statistical analysis. Nevertheless, this has led to both a loss in public confidence in the scientific enterprise and some serious soul searching within certain fields. Publishers have begun to take the lead in insisting on more careful enable audit and reuse when technically and legally possible.

Some communities within ACM have taken action. SIGMOD has been a true pioneer, establishing a reproducibility review of papers at the SIGMOD conference since 2008. The Artifact Evaluation for Software Conferences initiative has led to formal evaluations of artifacts (such as software and data) associated with papers in 11 major conferences since 2011, including OOPSLA, PLDI, and ISSTA. Here the extra evaluations are optional and are performed only after acceptance. In 2015 the *ACM Transactions on Mathematical Software* announced a Replicated Computational Results initiative,[c] also optional, in which the main results of a paper are independently replicated by a third party (who works cooperatively with the author and uses both confidence in results and downstream reproduction are enhanced if a paper's artifacts (that is, code and datasets) have undergone a rigorous auditing process such as those being undertaken by ACM conferences. The new ACM policy provides two badges that can be applied here: *Artifacts Evaluated—Functional*, when the artifacts are found to be documented, consistent, complete, exercisable, and include appropriate evidence of verification and validation, and if, in addition the artifacts facilitate reuse and repurposing at a higher level, then *Artifacts Evaluated—Reusable* can be applied. When artifacts are made publicly available, further enhancing auditing and reuse, we apply an *Artifacts Available* badge. ACM is working to expose these badges in the ACM Digital Library on

ACM: Result & artifact badging

doi> 10.1145/2994031

from the publications board

"This policy is but the first deliverable of the ACM Task Force on Data, Software and Reproducibility. Ongoing efforts are aimed at surfacing software and data as first-class objects in the DL, so it can serve as both a host and a catalog for not just articles, but the full range of research artifacts deserving preservation."

DOI:10.1145/2994031

Incentiviz

A SCIENTIFIC RESULT is n… established until it … dependently conf… This is one of the te… experimental scienc… we have seen a rash of recent hea… about experimental results that… not be reproduced. In the biom… field, efforts to reproduce results of academic research by drug companies have had less than a 50% success rate,[a] resulting in billions of dollars in wasted effort.[b] In most cases the cause is not intentional fraud, but rather sloppy research protocols and faulty statistical analysis. Nevertheless, this has led to both a loss in public confidence in the scientific enterprise and some serious soul searching within certain fields. Publishers have begun to take the lead in insisting on more careful

led to formal evaluations of artifacts (such as software and data) associated with papers in 11 major conferences since 2011, including OOPSLA, PLDI, and ISSTA. Here the extra evaluations are optional and are performed only after acceptance. In 2015 the *ACM Transactions on Mathematical Software* announced a Replicated Computational Results initiative,[c] also optional, in which the main results of a paper are independently replicated by a third party (who works cooperatively with the author and uses

uated—Functional, when the artifacts are found to be documented, consistent, complete, exercisable, and include appropriate evidence of verification and validation, and if, in addition the artifacts facilitate reuse and repurposing at a higher level, then *Artifacts Evaluated—Reusable* can be applied. When artifacts are made publicly available, further enhancing auditing and reuse, we apply an *Artifacts Available* badge. ACM is working to expose these badges in the ACM Digital Library on

ACM badges

- **Artifacts Evaluated—Functional**
  - documented, consistent, complete, exercisable
- **Artifacts Evaluated—Reusable**
  - functional, plus
  - reuse and repurposing is facilitated
- **Artifacts Available**
  - placed on a publicly accessible archival repository
- **Results Reproduced**
  - main results have been obtained in a subsequent study by someone other than the authors, using artifacts provided by the author
- **Results Replicated**
  - main results have been independently obtained in a subsequent study by someone other than the authors, without author-supplied artifacts
- may be awarded post-publication

https://www.acm.org/publications/policies/artifact-review-and-badging-current

Artifact evaluation is now commonplace. Did we win?

Education

doi> 10.1145/3089262.3089266

Learning Networking by Reproducing Research Results

Lisa Yan
Stanford University
yanlisa@stanford.edu

Nick McKeown
Stanford University
nickm@stanford.edu

This article is an editorial note submitted to CCR. It has NOT been peer reviewed.
The authors take full responsibility for this article's technical content. Comments can be posted through CCR Online.

ABSTRACT

In the past five years, the graduate networking course at Stanford has assigned over 200 students the task of reproducing results from over 40 networking papers. We began the project as a means of teaching both engineering rigor and critical thinking, qualities that are necessary for careers in networking research and industry. We have observed that reproducing research can simultaneously be a tool for education and a means for students to contribute to the networking community. Through this editorial we describe our project in reproducing network research and show through anecdotal evidence that this project is important for both the classroom and the networking community at large, and we hope to encourage other institutions to host similar class projects.

CCS Concepts

•**Social and professional topics** → **Computing education;** •**Networks** → *Network performance evaluation;*

Keywords

Reproducible research, Teaching computer networks

our experience, students who experience "building their own Internet" gain a thorough knowledge of how the Internet works, how to read and implement RFCs, and how to build network systems.

For a more advanced graduate class in networking, it is less obvious what the most appropriate programming assignments are. Should students build more advanced pieces of the Internet—such as firewalls, load-balancers, and new transport layers? This has the advantage of giving them more experience building network systems, but lacks a research ingenuity component where they can dream up and test their own ideas. And so it is more common in graduate studies for students to do a more creative open-ended project of their own design, perhaps using a simulator, testbed or analytical tools. In our earlier experience with CS244, we opted for the second style, and had students create open-ended projects of their own design. But we kept finding the projects to be lacking—mostly because it is hard to build a meaningful networking system or a persuasive prototype in such a short time. Often, students picked projects that turned out to be too ambitious, and on an incomplete prototype it was hard to collect meaningful experimental results. As a result, the projects tended to be incremental, and the educational experience of the students seemed to be too sus-

Education

doi> 10.1145/3089262.3089266

Learning Net

S
yanli

This articl
The authors take full respor

ABSTRACT

In the past five years, the gra
Stanford has assigned over 200
ducing results from over 40 ne
the project as a means of tea
and critical thinking, qualities
in networking research and indu
reproducing research can simu
cation and a means for studen
working community. Through t
project in reproducing network
anecdotal evidence that this p
the classroom and the network
we hope to encourage other ins
projects.

CCS Concepts

•**Social and professional topics** → **Computing education;** •**Networks** → *Network performance evaluation;*

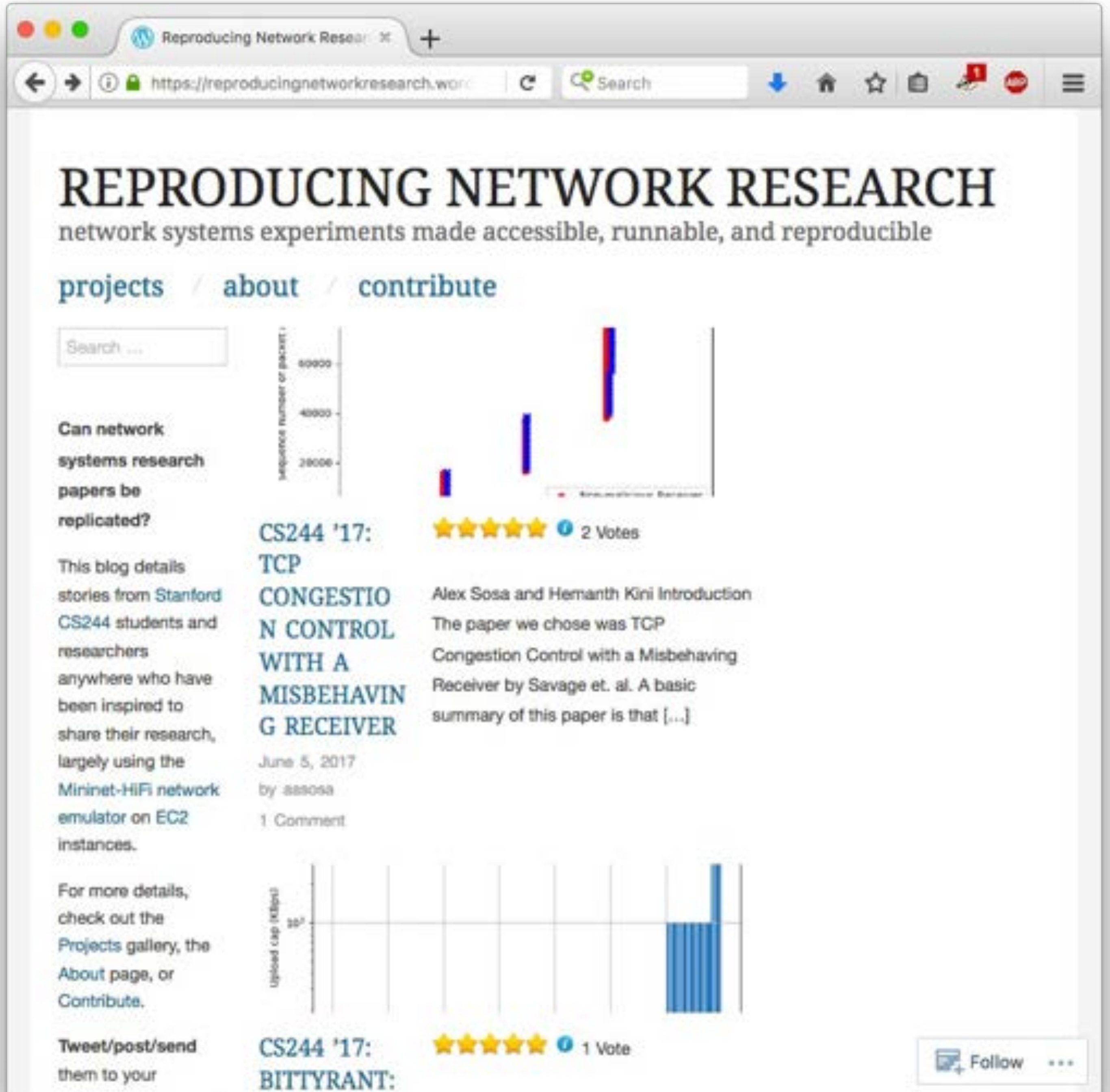
Keywords

Reproducible research, Teaching computer networks

opted for the second style, and had students create open-ended projects of their own design. But we kept finding the projects to be lacking—mostly because it is hard to build a meaningful networking system or a persuasive prototype in such a short time. Often, students picked projects that turned out to be too ambitious, and on an incomplete prototype it was hard to collect meaningful experimental results. As a result, the projects tended to be incremental, and the educational experience of the students seemed to be too sus-

"We have observed that reproducing research can simultaneously be a tool for education and a means for students to contribute to the networking community. Through this editorial we describe our project in reproducing network research and show through anecdotal evidence that this project is important for both the classroom and the networking community at large…"

https://reproducingnetworkresearch.wordpress.com/

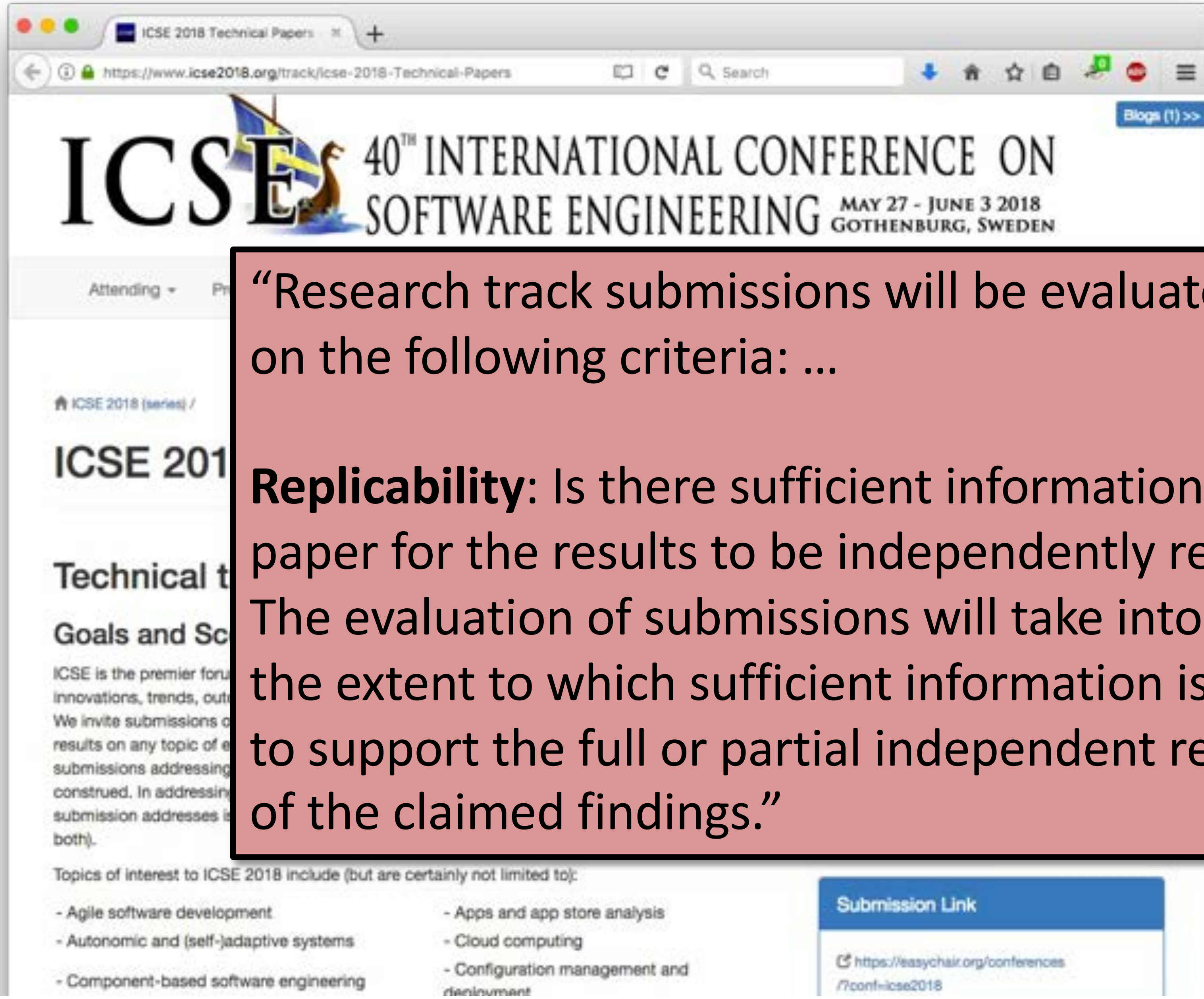


Replicability as a criterion

https://www.icse2018.org/track/icse-2018-Technical-Papers

Replicability as a criterion

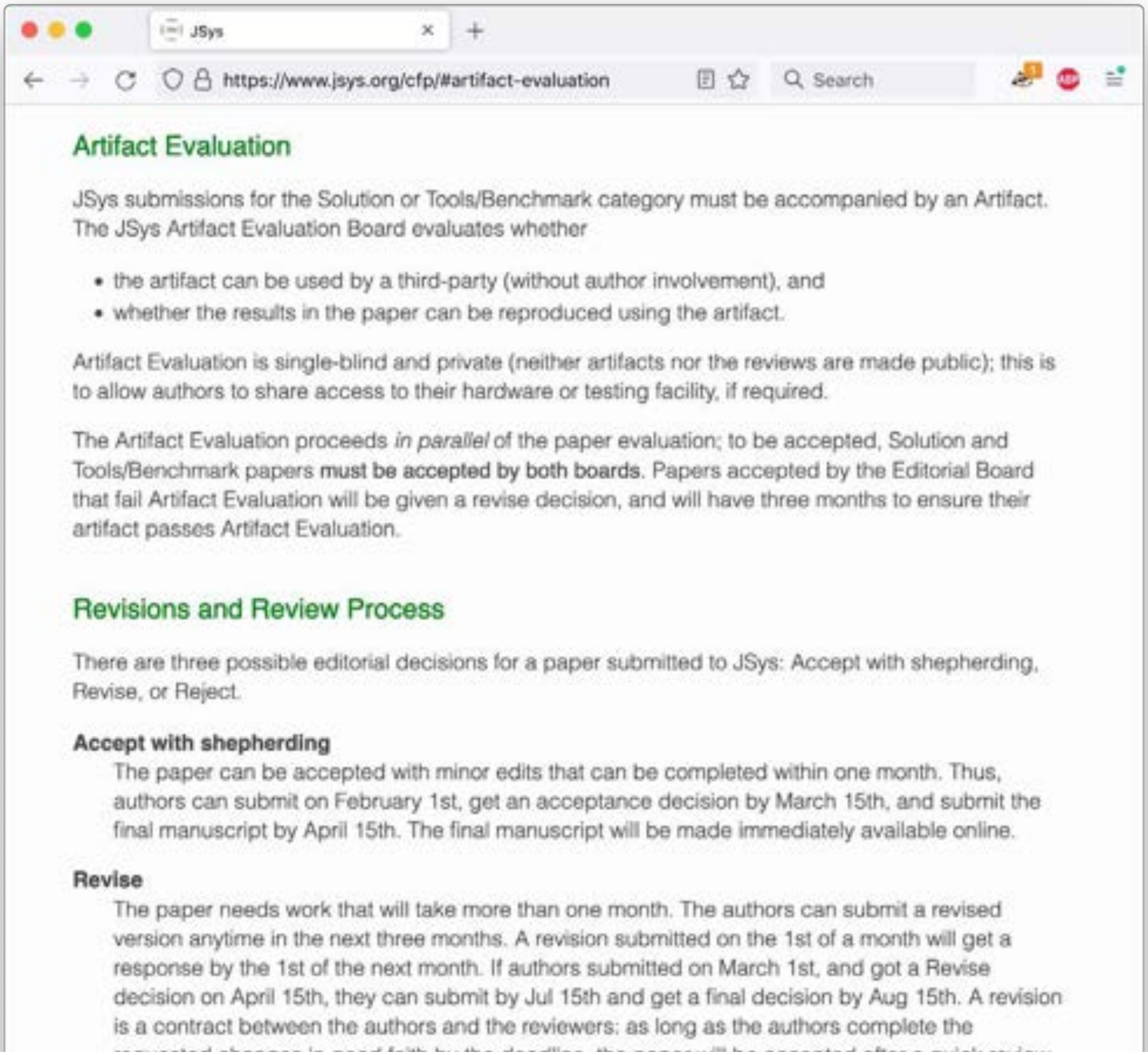
https://www.icse2018.org/track/icse-2018-Technical-Papers



"Research track submissions will be evaluated based on the following criteria: …

Replicability: Is there sufficient information in the paper for the results to be independently replicated? The evaluation of submissions will take into account the extent to which sufficient information is available to support the full or partial independent replication of the claimed findings."

Reproduction as a requirement



Artifact Evaluation

JSys submissions for the Solution or Tools/Benchmark category must be accompanied by an Artifact. The JSys Artifact Evaluation Board evaluates whether

- the artifact can be used by a third-party (without author involvement), and
- whether the results in the paper can be reproduced using the artifact.

Artifact Evaluation is single-blind and private (neither artifacts nor the reviews are made public); this is to allow authors to share access to their hardware or testing facility, if required.

The Artifact Evaluation proceeds *in parallel* of the paper evaluation; to be accepted, Solution and Tools/Benchmark papers **must be accepted by both boards**. Papers accepted by the Editorial Board that fail Artifact Evaluation will be given a revise decision, and will have three months to ensure their artifact passes Artifact Evaluation.

Revisions and Review Process

There are three possible editorial decisions for a paper submitted to JSys: Accept with shepherding, Revise, or Reject.

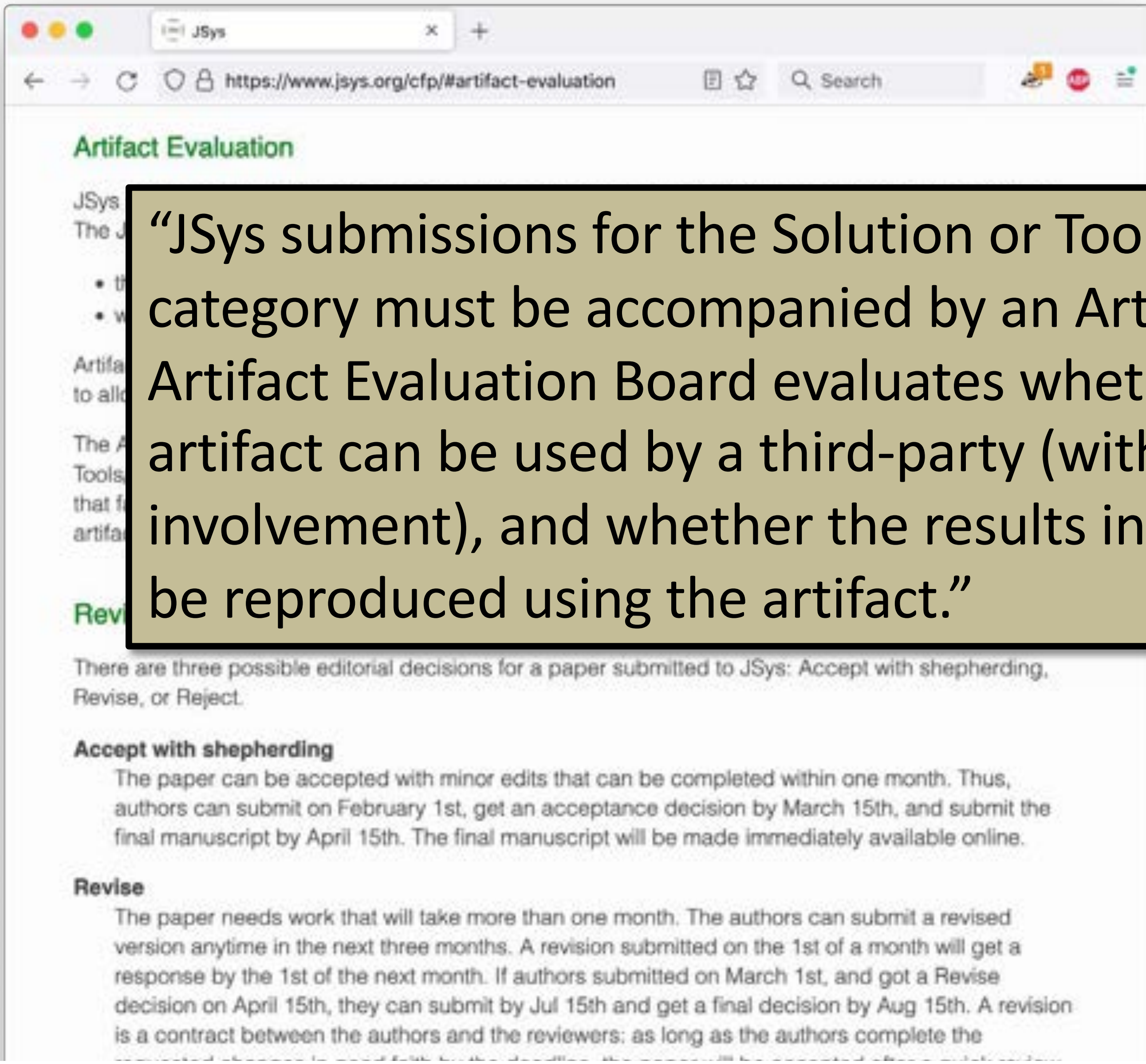
Accept with shepherding
The paper can be accepted with minor edits that can be completed within one month. Thus, authors can submit on February 1st, get an acceptance decision by March 15th, and submit the final manuscript by April 15th. The final manuscript will be made immediately available online.

Revise
The paper needs work that will take more than one month. The authors can submit a revised version anytime in the next three months. A revision submitted on the 1st of a month will get a response by the 1st of the next month. If authors submitted on March 1st, and got a Revise decision on April 15th, they can submit by Jul 15th and get a final decision by Aug 15th. A revision is a contract between the authors and the reviewers: as long as the authors complete the

https://www.jsys.org/cfp/#artifact-evaluation

Reproduction as a requirement

https://www.jsys.org/cfp/#artifact-evaluation


JSys

https://www.jsys.org/cfp/#artifact-evaluation

Search

Artifact Evaluation

There are three possible editorial decisions for a paper submitted to JSys: Accept with shepherding, Revise, or Reject.

Accept with shepherding

The paper can be accepted with minor edits that can be completed within one month. Thus, authors can submit on February 1st, get an acceptance decision by March 15th, and submit the final manuscript by April 15th. The final manuscript will be made immediately available online.

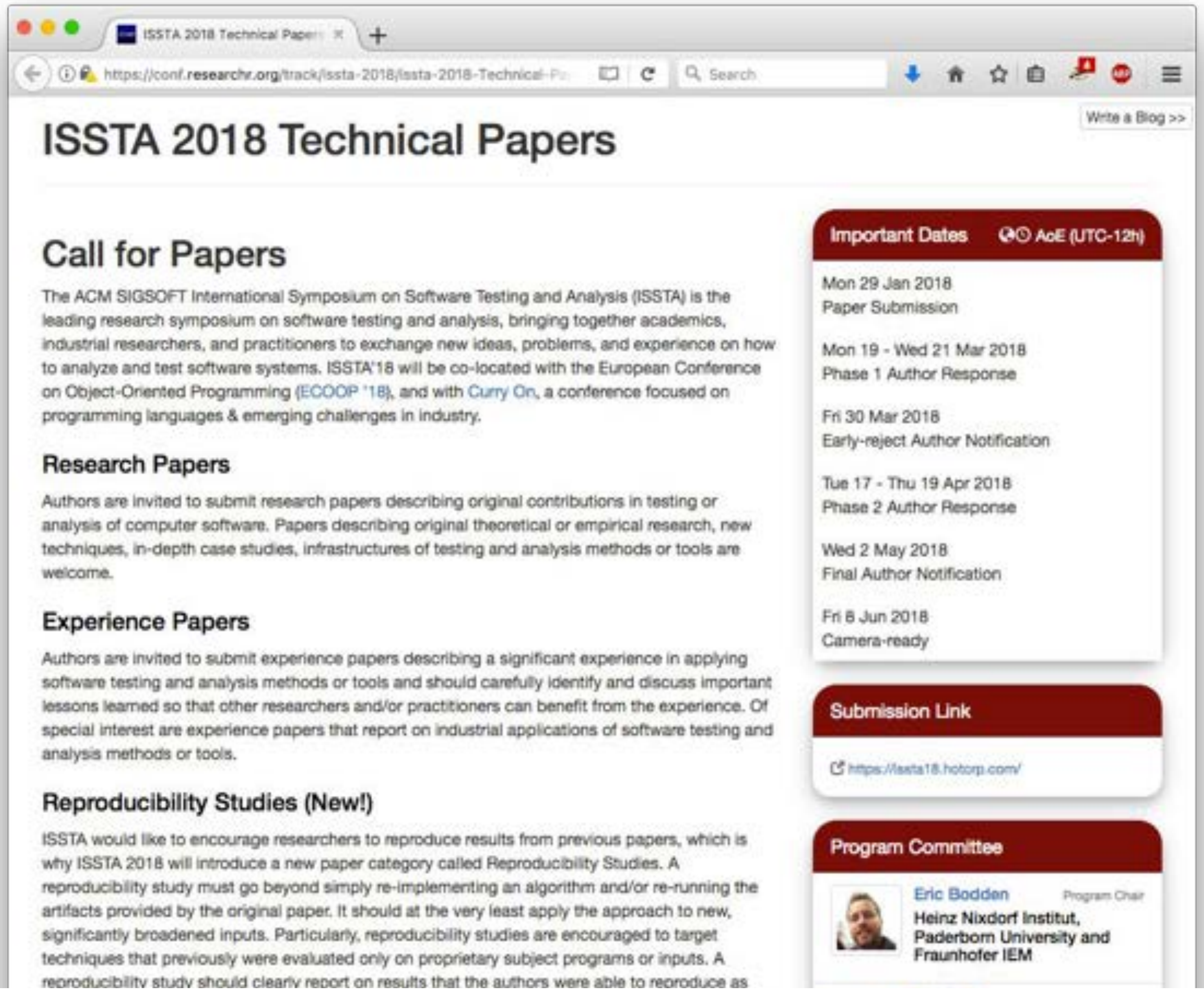
Revise

The paper needs work that will take more than one month. The authors can submit a revised version anytime in the next three months. A revision submitted on the 1st of a month will get a response by the 1st of the next month. If authors submitted on March 1st, and got a Revise decision on April 15th, they can submit by Jul 15th and get a final decision by Aug 15th. A revision is a contract between the authors and the reviewers: as long as the authors complete the

"JSys submissions for the Solution or Tools/Benchmark category must be accompanied by an Artifact. The JSys Artifact Evaluation Board evaluates whether the artifact can be used by a third-party (without author involvement), and whether the results in the paper can be reproduced using the artifact."

Calls for reproducibility studies

https://conf.researchr.org/track/issta-2018/issta-2018-Technical-Papers



ISSTA 2018 Technical Papers

https://conf.researchr.org/track/issta-2018/issta-2018-Technical-Pa...

Write a Blog >>

ISSTA 2018 Technical Papers

Call for Papers

The ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA) is the leading research symposium on software testing and analysis, bringing together academics, industrial researchers, and practitioners to exchange new ideas, problems, and experience on how to analyze and test software systems. ISSTA'18 will be co-located with the European Conference on Object-Oriented Programming (ECOOP '18), and with Curry On, a conference focused on programming languages & emerging challenges in industry.

Research Papers

Authors are invited to submit research papers describing original contributions in testing or analysis of computer software. Papers describing original theoretical or empirical research, new techniques, in-depth case studies, infrastructures of testing and analysis methods or tools are welcome.

Experience Papers

Authors are invited to submit experience papers describing a significant experience in applying software testing and analysis methods or tools and should carefully identify and discuss important lessons learned so that other researchers and/or practitioners can benefit from the experience. Of special interest are experience papers that report on industrial applications of software testing and analysis methods or tools.

Reproducibility Studies (New!)

ISSTA would like to encourage researchers to reproduce results from previous papers, which is why ISSTA 2018 will introduce a new paper category called Reproducibility Studies. A reproducibility study must go beyond simply re-implementing an algorithm and/or re-running the artifacts provided by the original paper. It should at the very least apply the approach to new, significantly broadened inputs. Particularly, reproducibility studies are encouraged to target techniques that previously were evaluated only on proprietary subject programs or inputs. A reproducibility study should clearly report on results that the authors were able to reproduce as

Important Dates AoE (UTC-12h)

Mon 29 Jan 2018
Paper Submission

Mon 19 - Wed 21 Mar 2018
Phase 1 Author Response

Fri 30 Mar 2018
Early-reject Author Notification

Tue 17 - Thu 19 Apr 2018
Phase 2 Author Response

Wed 2 May 2018
Final Author Notification

Fri 8 Jun 2018
Camera-ready

Submission Link

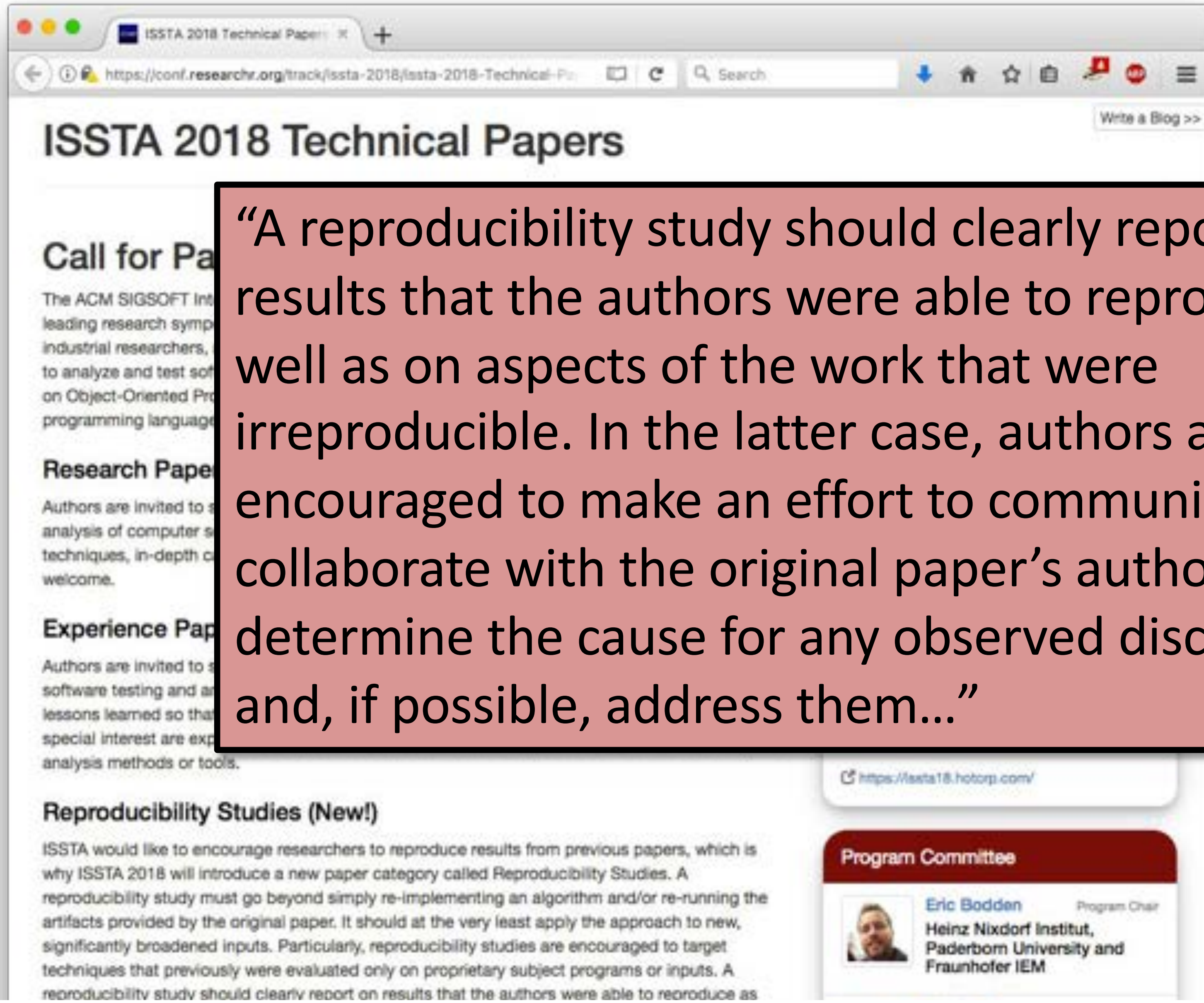
https://issta18.hotcrp.com/

Program Committee

Eric Bodden — Program Chair
Heinz Nixdorf Institut, Paderborn University and Fraunhofer IEM

Calls for reproducibility studies

https://conf.researchr.org/track/issta-2018/issta-2018-Technical-Papers



"A reproducibility study should clearly report on results that the authors were able to reproduce as well as on aspects of the work that were irreproducible. In the latter case, authors are encouraged to make an effort to communicate or collaborate with the original paper's authors to determine the cause for any observed discrepancies and, if possible, address them…"

This paper has badges, but…

Is it *really* reusable?

doi> 10.1145/3402413.3402418

An Artifact Evaluation of NDP

Noa Zilberman
University of Oxford, UK
noa.zilberman@eng.ox.ac.uk

ABSTRACT

Artifact badging aims to rank the quality of submitted research artifacts and promote reproducibility. However, artifact badging may not indicate inherent design and evaluation limitations. This work explores current limits in artifact badging using a performance-based evaluation of the NDP [7] artifact. We evaluate the NDP artifact beyond the *Reusable* badge's level, investigating the effect of aspects such as packet size and random-number seed on throughput and flow completion time. Our evaluation demonstrates that while the NDP artifact is *reusable*, it is not *robust*, and we identify architectural, implementation and evaluation limitations.

CCS CONCEPTS

• **General and reference → Evaluation**; • **Networks → Data center networks**;

KEYWORDS

Reproducibility, Artifact Evaluation, Datacenters, Transport Protocols

1 INTRODUCTION

NDP, a novel data centre transport architecture, was proposed by Handley *et al.* [7], aiming to achieve both low latency and high throughput. NDP offers better short-flow performance than DCTCP [2] or DCQCN [16], achieving more than 95% of the maximum network capacity in a heavily loaded network, near-perfect

window based rather than rate based. We use the simulation environment "as is", except for the minimum amount of changes required to evaluate a specific aspect, e.g., setting the packet size or changing packet size distribution. All the simulations were done on a Xeon E5-2660 v4 server, using 256GB of DDR4-2400 RAM, running at 3.2GHz, and using Ubuntu 14.04, kernel version 3.13.0-32-generic.

Hardware Environment. The Implementation of NDP switch on NetFPGA SUME [19] is based on the NetFPGA Reference Switch design. The NDP switch supports both NDP and non-NDP traffic. We compare the performance of NDP with the NetFPGA Reference Switch, running traffic through both designs. Both designs are synthesized using NetFPGA-SUME release 1.7.1. Our setup is composed of two identical NetFPGA SUME boards, one configured as OSNT [3], an open source network tester (release 1.7.0), and the other as the device under test. The boards are hosted within two identical i7-6700K machines running Ubuntu 14.04; although the host setup has no impact on the test.

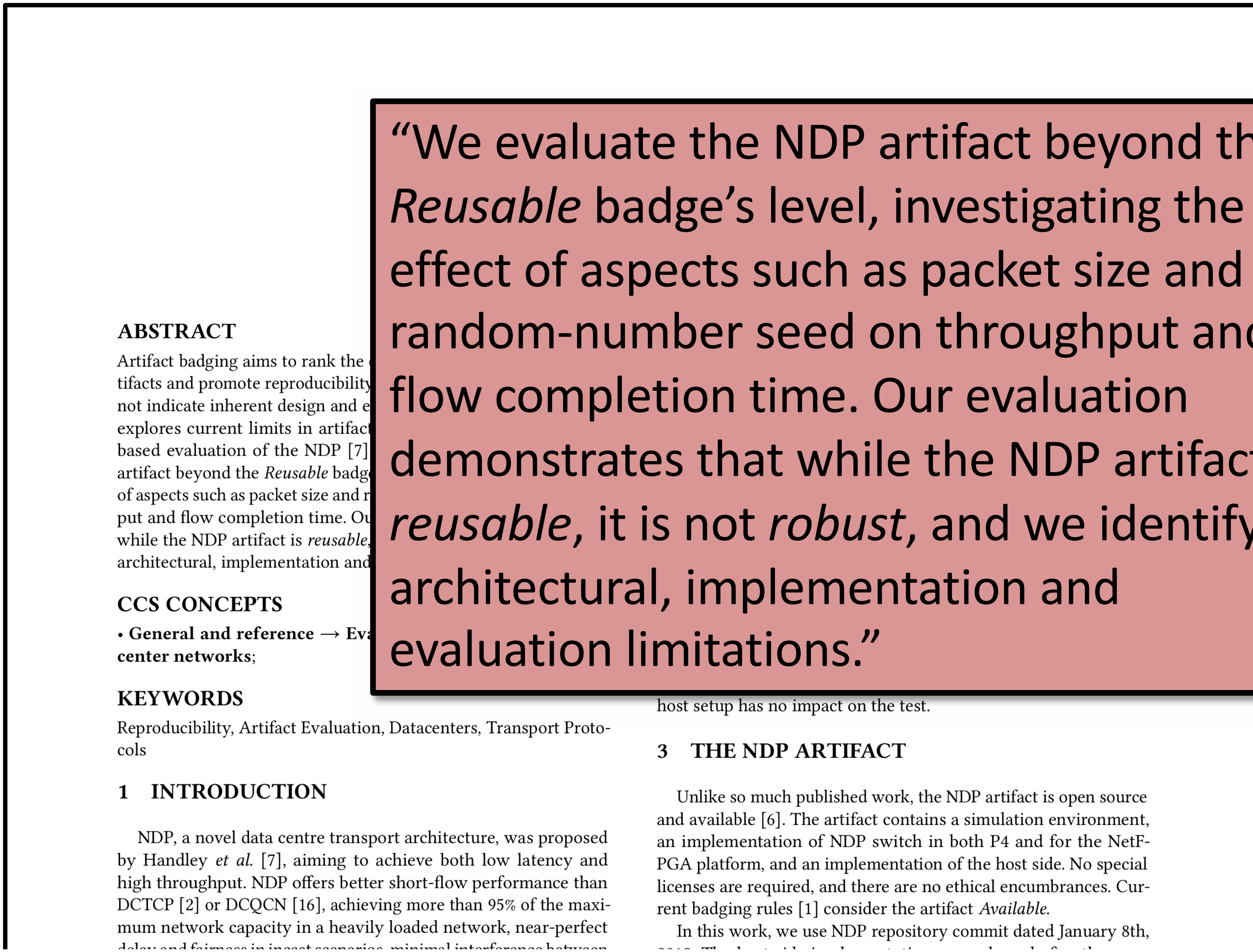
3 THE NDP ARTIFACT

Unlike so much published work, the NDP artifact is open source and available [6]. The artifact contains a simulation environment, an implementation of NDP switch in both P4 and for the NetFPGA platform, and an implementation of the host side. No special licenses are required, and there are no ethical encumbrances. Current badging rules [1] consider the artifact *Available*.

In this work, we use NDP repository commit dated January 8th,

Is it *really* reusable?

doi> 10.1145/3402413.3402418


ABSTRACT

Artifact badging aims to rank the
tifacts and promote reproducibility
not indicate inherent design and e
explores current limits in artifact
based evaluation of the NDP [7]
artifact beyond the *Reusable* badg
of aspects such as packet size and r
put and flow completion time. Ou
while the NDP artifact is *reusable*,
architectural, implementation and

CCS CONCEPTS

• **General and reference** → **Ev**
center networks;

KEYWORDS

Reproducibility, Artifact Evaluation, Datacenters, Transport Protocols

1 INTRODUCTION

NDP, a novel data centre transport architecture, was proposed by Handley *et al.* [7], aiming to achieve both low latency and high throughput. NDP offers better short-flow performance than DCTCP [2] or DCQCN [16], achieving more than 95% of the maximum network capacity in a heavily loaded network, near-perfect

host setup has no impact on the test.

3 THE NDP ARTIFACT

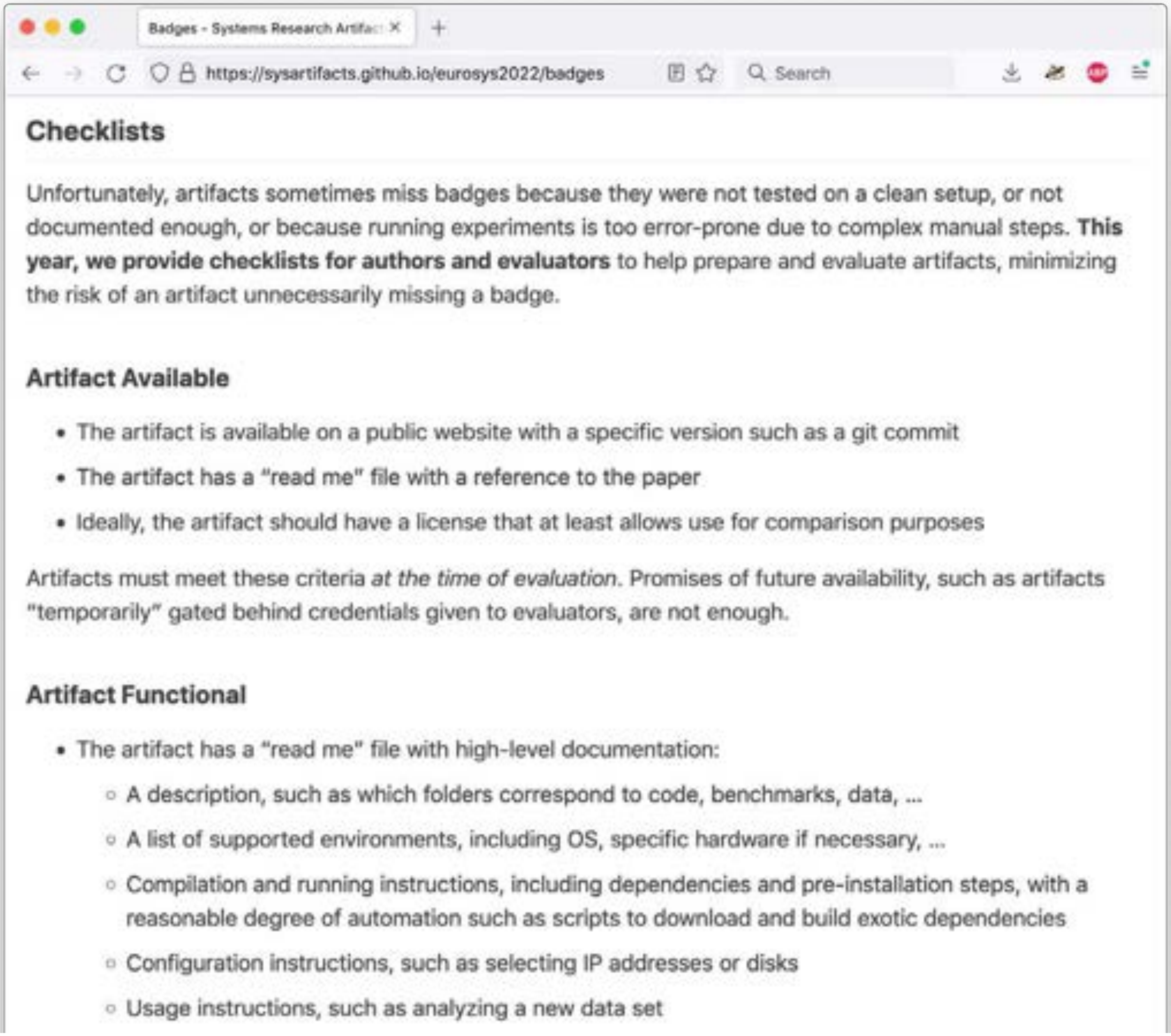
Unlike so much published work, the NDP artifact is open source and available [6]. The artifact contains a simulation environment, an implementation of NDP switch in both P4 and for the NetFPGA platform, and an implementation of the host side. No special licenses are required, and there are no ethical encumbrances. Current badging rules [1] consider the artifact *Available*.

In this work, we use NDP repository commit dated January 8th,

"We evaluate the NDP artifact beyond the *Reusable* badge's level, investigating the effect of aspects such as packet size and random-number seed on throughput and flow completion time. Our evaluation demonstrates that while the NDP artifact is *reusable*, it is not *robust*, and we identify architectural, implementation and evaluation limitations."

What does a badge really mean?

https://sysartifacts.github.io/eurosys2022/badges



Badges - Systems Research Artifac ×

https://sysartifacts.github.io/eurosys2022/badges

Search

Checklists

Unfortunately, artifacts sometimes miss badges because they were not tested on a clean setup, or not documented enough, or because running experiments is too error-prone due to complex manual steps. **This year, we provide checklists for authors and evaluators** to help prepare and evaluate artifacts, minimizing the risk of an artifact unnecessarily missing a badge.

Artifact Available

- The artifact is available on a public website with a specific version such as a git commit
- The artifact has a "read me" file with a reference to the paper
- Ideally, the artifact should have a license that at least allows use for comparison purposes

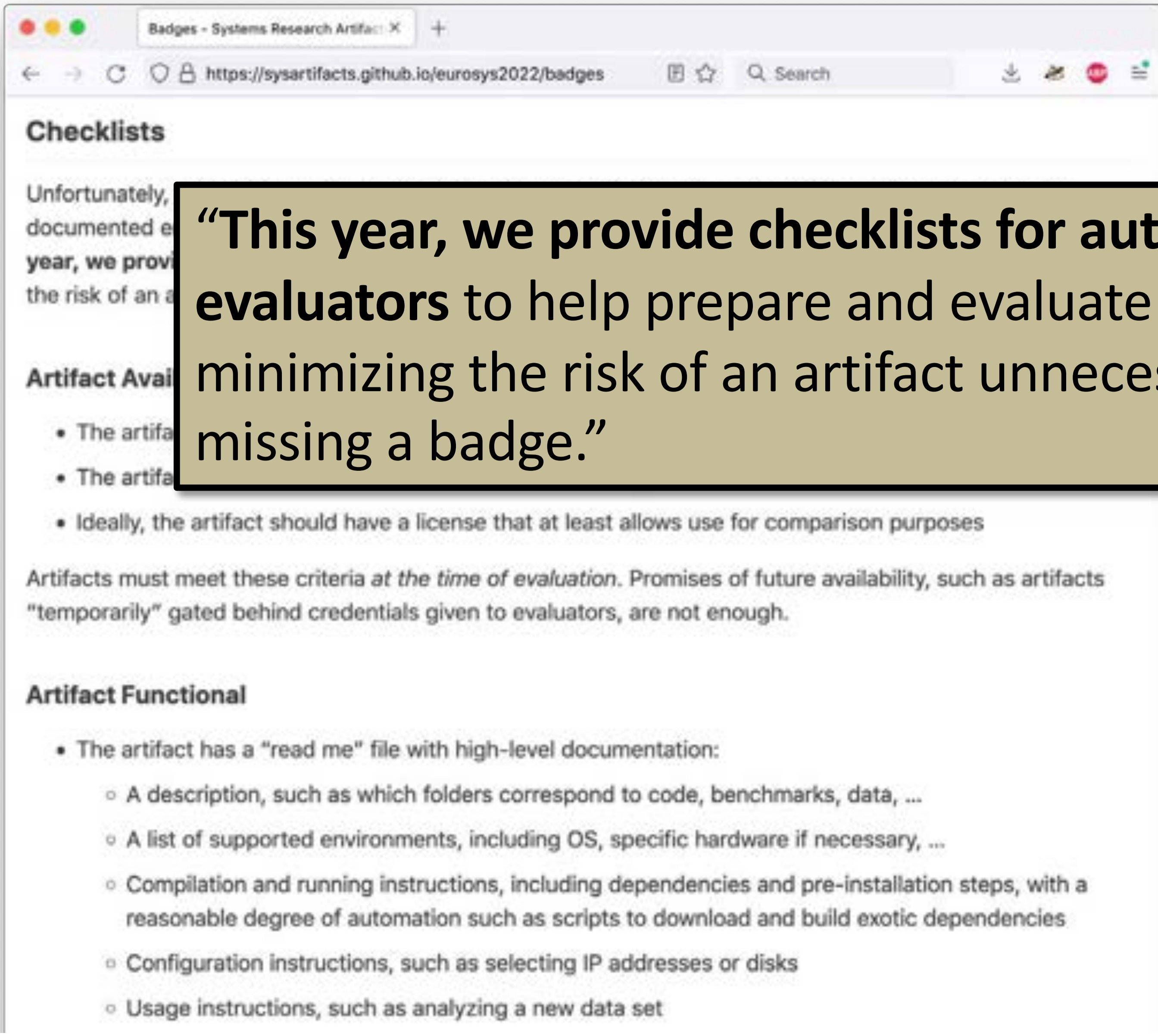
Artifacts must meet these criteria *at the time of evaluation*. Promises of future availability, such as artifacts "temporarily" gated behind credentials given to evaluators, are not enough.

Artifact Functional

- The artifact has a "read me" file with high-level documentation:
  - A description, such as which folders correspond to code, benchmarks, data, ...
  - A list of supported environments, including OS, specific hardware if necessary, ...
  - Compilation and running instructions, including dependencies and pre-installation steps, with a reasonable degree of automation such as scripts to download and build exotic dependencies
  - Configuration instructions, such as selecting IP addresses or disks
  - Usage instructions, such as analyzing a new data set

What does a badge really mean?

https://sysartifacts.github.io/eurosys2022/badges



Checklists

Unfortunately,
documented e
year, we provi
the risk of an a

Artifact Avai

- The artifa
- The artifa
- Ideally, the artifact should have a license that at least allows use for comparison purposes

Artifacts must meet these criteria *at the time of evaluation*. Promises of future availability, such as artifacts "temporarily" gated behind credentials given to evaluators, are not enough.

Artifact Functional

- The artifact has a "read me" file with high-level documentation:
  - A description, such as which folders correspond to code, benchmarks, data, ...
  - A list of supported environments, including OS, specific hardware if necessary, ...
  - Compilation and running instructions, including dependencies and pre-installation steps, with a reasonable degree of automation such as scripts to download and build exotic dependencies
  - Configuration instructions, such as selecting IP addresses or disks
  - Usage instructions, such as analyzing a new data set

“**This year, we provide checklists for authors and evaluators** to help prepare and evaluate artifacts, minimizing the risk of an artifact unnecessarily missing a badge.”

Difficult-to-evaluate artifacts

doi> 10.1145/3402413.3402418

Getting Research Software to Work: A Case Study on Artifact Evaluation for OOPSLA 2019

Erin Dahlgren
Accelerate Publishing[1]

***Abstract*— Due to new peer-review programs, researchers in certain fields can now receive badges on their papers that reward them for writing functional and reusable research code. These badges in turn make their research more attractive for others to cite and build upon. Unfortunately, some submissions to these new programs do not pass the lowest bar, and many submissions are difficult for reviewers to simply setup and test. To understand how to improve submissions and how to help researchers gain badges, we studied the artifact evaluation process of OOPSLA 2019, an ACM conference on the analysis and design of computer programs. Based on reviewer experiences, we highlight best practices and we discuss whether guidelines, tools, or larger cooperative efforts are required to achieve them. To conclude, we present ongoing and future work that helps researchers share and use research code.**

1. Introduction

Many researchers today are frustrated with how difficult it is to reproduce published results [1], [2]. Despite the widespread use of software to conduct research [3], rarely can research software be found, run, and reused, making important research results hard to trust and build upon [4]. In an effort to address this, the Association of Computing Machinery (ACM) created an initiative to award badges

artifact hard to test?" and "What makes an artifact easy to test?". Part 5 summarizes and discusses the data in Parts 3-4, and finally, Part 6 presents ongoing and future work. Henceforth, the terminology and acronyms below will be used interchangeably through this report:

Term	*Description*
artifact	research software artifact
reviewers	members of an artifact evaluation committee
image	contains the files of an artifact
archive	a compressed directory
VM	short for "Virtual Machine"
container	short for "Linux container"
open source	freely readable code
code	software

2. Methodology

To collect data, we participated as a member of the artifact evaluation committee for the OOPSLA 2019 conference [6]. Since OOPSLA accepts research on the analysis and design of computer programs, naturally in many cases research artifacts were in and of themselves the research results,

Difficult-to-evaluate artifacts

doi> 10.1145/3402413.3402418

"…some submissions to these new [artifact evaluation] programs do not pass the lowest bar, and many submissions are difficult for reviewers to simply setup and test."

A Case St

Abstract— **Due to new peer-review programs, researchers in certain fields can now receive badges on their papers that reward them for writing functional and reusable research code. These badges in turn make their research more attractive for others to cite and build upon. Unfortunately, some submissions to these new programs do not pass the lowest bar, and many submissions are difficult for reviewers to simply setup and test. To understand how to improve submissions and how to help researchers gain badges, we studied the artifact evaluation process of OOPSLA 2019, an ACM conference on the analysis and design of computer programs. Based on reviewer experiences, we highlight best practices and we discuss whether guidelines, tools, or larger cooperative efforts are required to achieve them. To conclude, we present ongoing and future work that helps researchers share and use research code.**

1. Introduction

Many researchers today are frustrated with how difficult it is to reproduce published results [1], [2]. Despite the widespread use of software to conduct research [3], rarely can research software be found, run, and reused, making important research results hard to trust and build upon [4]. In an effort to address this, the Association of Computing Machinery (ACM) created an initiative to award badges

artifact hard to test?" and "What makes an artifact easy to test?". Part 5 summarizes and discusses the data in Parts 3-4, and finally, Part 6 presents ongoing and future work. Henceforth, the terminology and acronyms below will be used interchangeably through this report:

Term	*Description*
artifact	research software artifact
reviewers	members of an artifact evaluation committee
image	contains the files of an artifact
archive	a compressed directory
VM	short for "Virtual Machine"
container	short for "Linux container"
open source	freely readable code
code	software

2. Methodology

To collect data, we participated as a member of the artifact evaluation committee for the OOPSLA 2019 conference [6]. Since OOPSLA accepts research on the analysis and design of computer programs, naturally in many cases research artifacts were in and of themselves the research results,

Dahlgren: issues encountered

- long-running tests
- not enough resources
- problems with documentation
- issues compiling or running
- issues with VM or container
- ignored errors
- issues with software dependencies
- works in limited environments
- errors in scripts
- too complicated
- downloads during execution

Artifact evaluation and reproduction: still not so 

software environment
hardware environment
availability of artifacts
incentives

Better practices & tools

SYSTEMS

Standing on the Shoulders of Giants by Managing Scientific Experiments Like Software

IVO JIMENEZ, MICHAEL SEVILLA, NOAH WATKINS, CARLOS MALTZAHN, JAY LOFSTEAD, KATHRYN MOHROR, REMZI ARPACI-DUSSEAU, AND ANDREA ARPACI-DUSSEAU

Ivo Jimenez is a PhD candidate at the UC Santa Cruz Computer Science Department and a member of the Systems Research Lab. His current work focuses on the practical reproducible generation and validation of systems research. Ivo holds a BS in computer science from University of Sonora and a MS from UCSC. ivo@cs.ucsc.edu

Michael Sevilla is a computer science PhD candidate at UC Santa Cruz. As part the Systems Research Lab, he evaluates distributed file system metadata management. At Hewlett Packard Enterprise, he uses open-source tools to benchmark storage solutions. He has a BS in computer

Independently validating experimental results in the field of computer systems research is a challenging task. Recreating an environment that resembles the one where an experiment was originally executed is a time-consuming endeavor. In this article, we present Popper [1], a convention (or protocol) for conducting experiments following a DevOps [2] approach that allows researchers to make all associated artifacts publicly available with the goal of maximizing automation in the re-execution of an experiment and validation of its results.

A basic expectation in the practice of the scientific method is to document, archive, and share all data and the methodologies used so other scientists can reproduce and verify scientific results and students can learn how they were derived. However, in the scientific branches of computation and data exploration the lack of reproducibility has led to a credibility crisis. As more scientific disciplines are relying on computational methods and data-intensive exploration, it has become urgent to develop software tools that help document dependencies on data products, methodologies, and computational environments, that safely archive data products and documentation, and that reliably share data products and documentations so that scientists can rely on their availability.

https://www.usenix.org/publications/login/winter-2016-vol-41-no-4/jimenez

Better practices & tools

"In this article, we present Popper, a convention (or protocol) for conducting experiments following a DevOps approach that allows researchers to make all associated artifacts publicly available with the goal of maximizing automation in the re-execution of an experiment and validation of its results."

ftware

Ivo Jimenez is a PhD candidate at the UC Santa Cruz Computer Science Department and a member of the Systems Research Lab. His current work focuses on the practical reproducible generation and validation of systems research. Ivo holds a BS in computer science from University of Sonora and a MS from UCSC. ivo@cs.ucsc.edu

Michael Sevilla is a computer science PhD candidate at UC Santa Cruz. As part the Systems Research Lab, he evaluates distributed file system metadata management. At Hewlett Packard Enterprise, he uses open-source tools to benchmark storage solutions. He has a BS in computer

Independently validating experimental results in the field of computer systems research is a challenging task. Recreating an environment that resembles the one where an experiment was originally executed is a time-consuming endeavor. In this article, we present Popper [1], a convention (or protocol) for conducting experiments following a DevOps [2] approach that allows researchers to make all associated artifacts publicly available with the goal of maximizing automation in the re-execution of an experiment and validation of its results.

A basic expectation in the practice of the scientific method is to document, archive, and share all data and the methodologies used so other scientists can reproduce and verify scientific results and students can learn how they were derived. However, in the scientific branches of computation and data exploration the lack of reproducibility has led to a credibility crisis. As more scientific disciplines are relying on computational methods and data-intensive exploration, it has become urgent to develop software tools that help document dependencies on data products, methodologies, and computational environments, that safely archive data products and documentation, and that reliably share data products and documentations so that scientists can rely on their availability.

https://www.usenix.org/publications/login/winter-2016-vol-41-no-4/jimenez

Popper

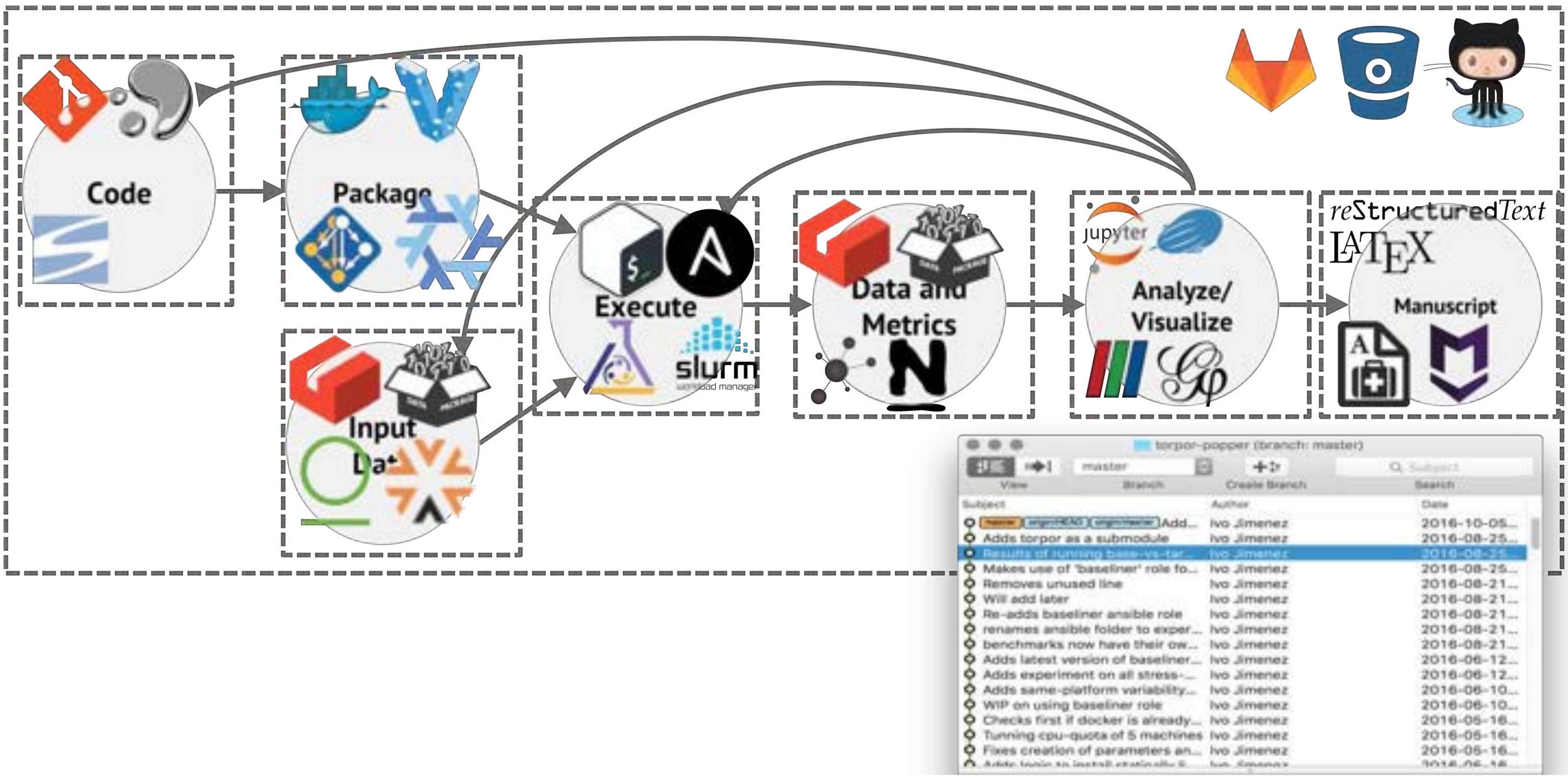


Figure source: Popper web site,
http://popper.readthedocs.io/en/latest/protocol/intro_to_popper.html

Better evaluation platforms

DataMill: Rigorous Performance Evaluation Made Easy

Augusto Born de Oliveira
Electrical and Computer Engineering
University of Waterloo
Waterloo, ON, Canada
a3oliveira@uwaterloo.ca

Jean-Christophe Petkovich
Electrical and Computer Engineering
University of Waterloo
Waterloo, ON, Canada
j2petkov@uwaterloo.ca

Thomas Reidemeister
Electrical and Computer Engineering
University of Waterloo
Waterloo, ON, Canada
treideme@uwaterloo.ca

Sebastian Fischmeister
Electrical and Computer Engineering
University of Waterloo
Waterloo, ON, Canada
sfischme@uwaterloo.ca

Abstract

Empirical systems research is facing a dilemma. Minor aspects of an experimental setup can have a significant impact on its associated performance measurements and potentially invalidate conclusions drawn from them. Examples of such influences, often called hidden factors, include binary link order, process environment size, compiler generated randomized symbol names, or group scheduler assignments. The growth in complexity and size of modern systems will further aggravate this dilemma, especially with the given time pressure of producing results. So how can one trust any reported empirical analysis of a new idea or concept in computer science?

This paper introduces DataMill, a community-based easy-to-use services-oriented open benchmarking infrastructure for performance evaluation. DataMill facilitates producing robust, reliable, and reproducible results. The infrastruc-

Keywords

DataMill; performance; experimentation; infrastructure; robustness; repeatability; reproducibility

1. INTRODUCTION

Empirical computer performance evaluation is essential for computer science and industry alike. The empirical measurement of performance sees widespread use to guide the research of new ideas and the development new technologies. A performance improvement of a few percentage points may mean large savings in dollars, when applied to a large data center with billions of clients. It is also essential, then, that computer practitioners dominate the methodology necessary to evaluate computer performance correctly.

However, the research community [10, 25, 31, 32] has demonstrated that experimental evaluation in computer sci-

doi> 10.1145/2479871.2479892

Better evaluation platforms

DataMill: Rigorous Performance Evaluation Made Easy

"Many aspects of complex performance experimentation are automated by DataMill enabling users to set up performance experiments easily. Due to its support for many different hardware platforms and automated factor variation, DataMill can cover a larger experiment space than typically considered by most researchers."

ized symbol names, or group scheduler assignments. The growth in complexity and size of modern systems will further aggravate this dilemma, especially with the given time pressure of producing results. So how can one trust any reported empirical analysis of a new idea or concept in computer science?

This paper introduces DataMill, a community-based easy-to-use services-oriented open benchmarking infrastructure for performance evaluation. DataMill facilitates producing robust, reliable, and reproducible results. The infrastruc-

...n is essential for computer science and industry alike. The empirical measurement of performance sees widespread use to guide the research of new ideas and the development new technologies. A performance improvement of a few percentage points may mean large savings in dollars, when applied to a large data center with billions of clients. It is also essential, then, that computer practitioners dominate the methodology necessary to evaluate computer performance correctly.

However, the research community [10, 25, 31, 32] has demonstrated that experimental evaluation in computer sci-

doi> 10.1145/2479871.2479892

DataMill

- define an experiment "package"
- auto execute on various hardware platforms…
  - x86/ARM, speed, mem size
- …with various software factors, e.g.
  - compiler flags
  - link orders
  - ASLR
- …and multiple trials

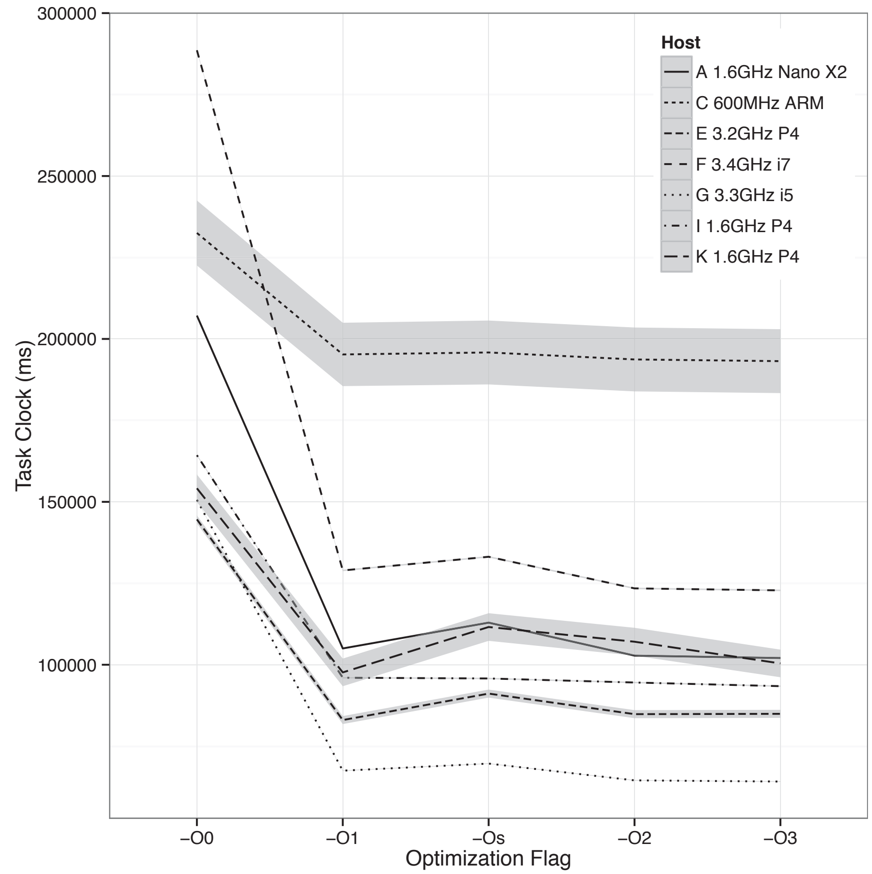
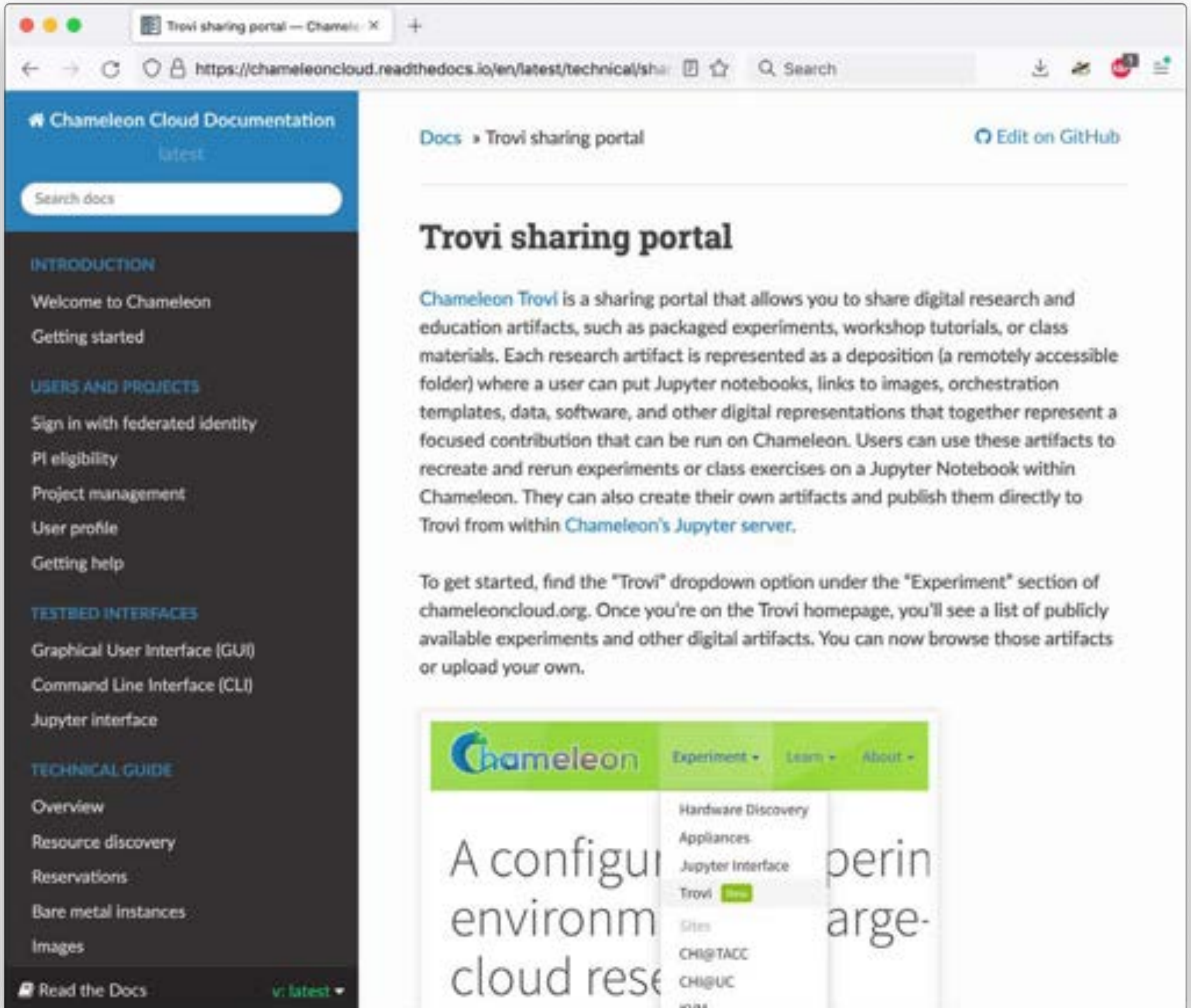


Figure 6: Effect of GCC Optimization Flags on XZ, by Host

Figure credit: de Oliveira et al., doi> 10.1145/2479871.2479892

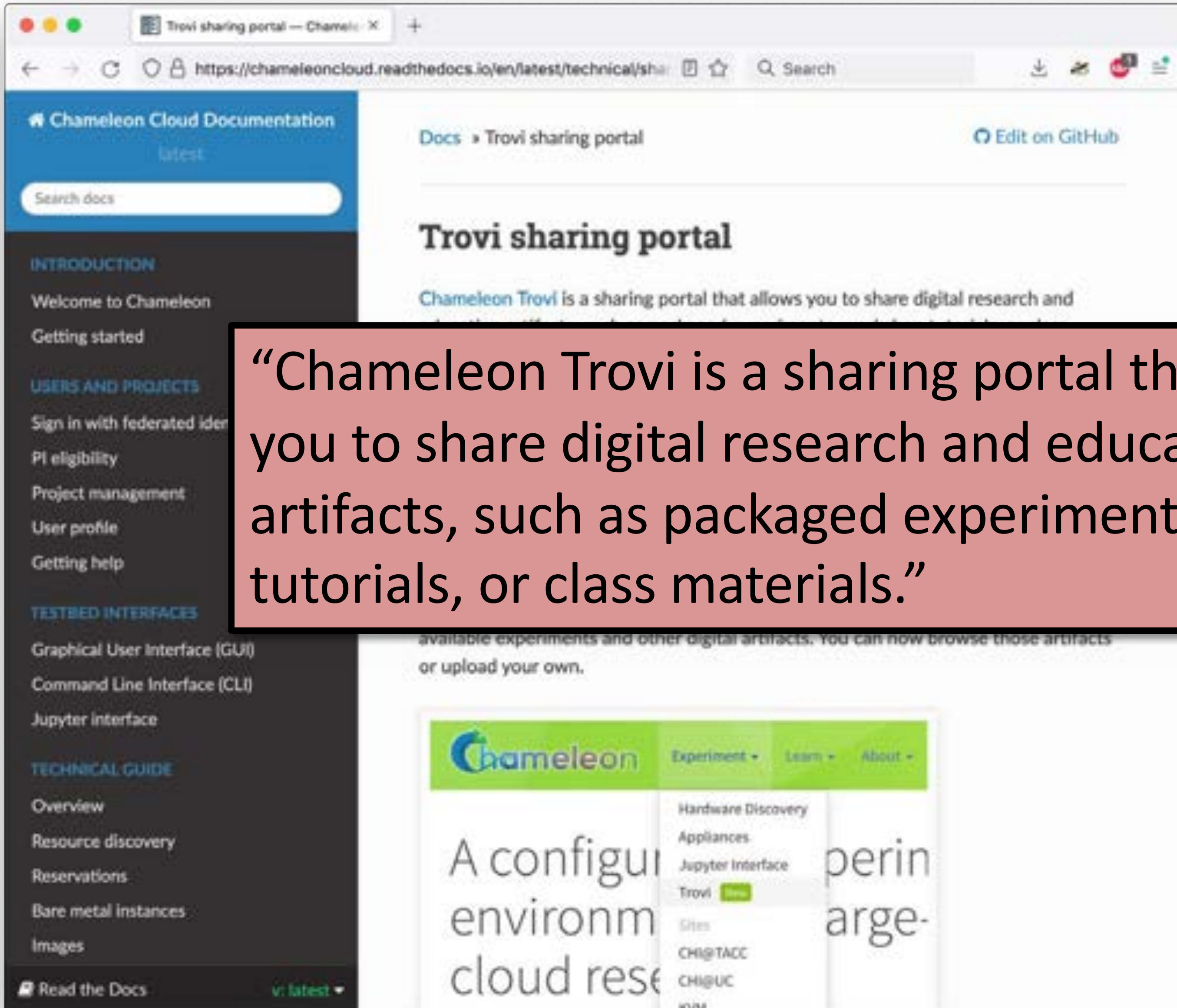
Sharing runnable artifacts



https://chameleoncloud.readthedocs.io/en/latest/technical/sharing.html

Sharing runnable artifacts

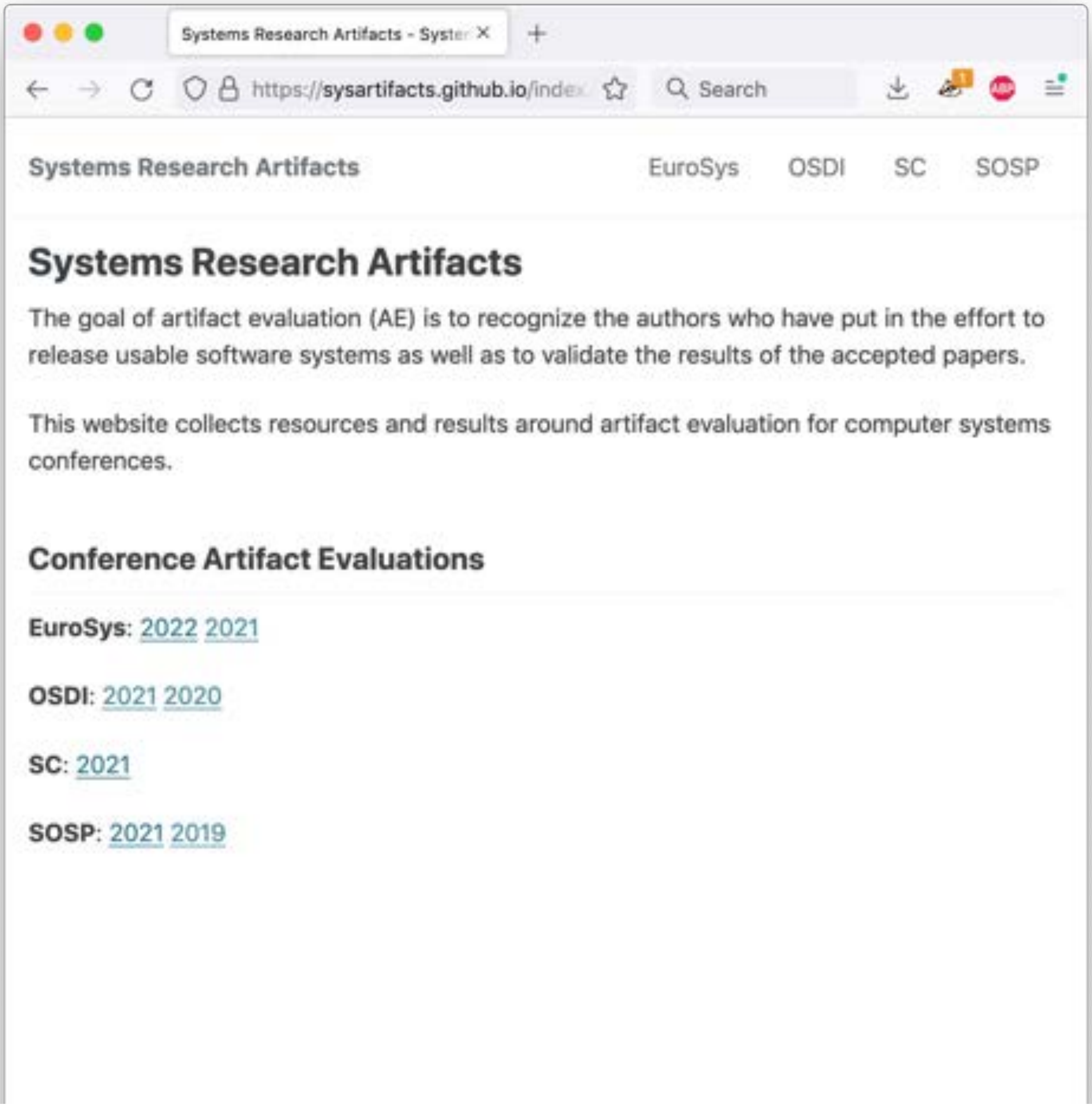
https://chameleoncloud.readthedocs.io/en/latest/technical/sharing.html



"Chameleon Trovi is a sharing portal that allows you to share digital research and education artifacts, such as packaged experiments, workshop tutorials, or class materials."

Artifact evaluation indexes

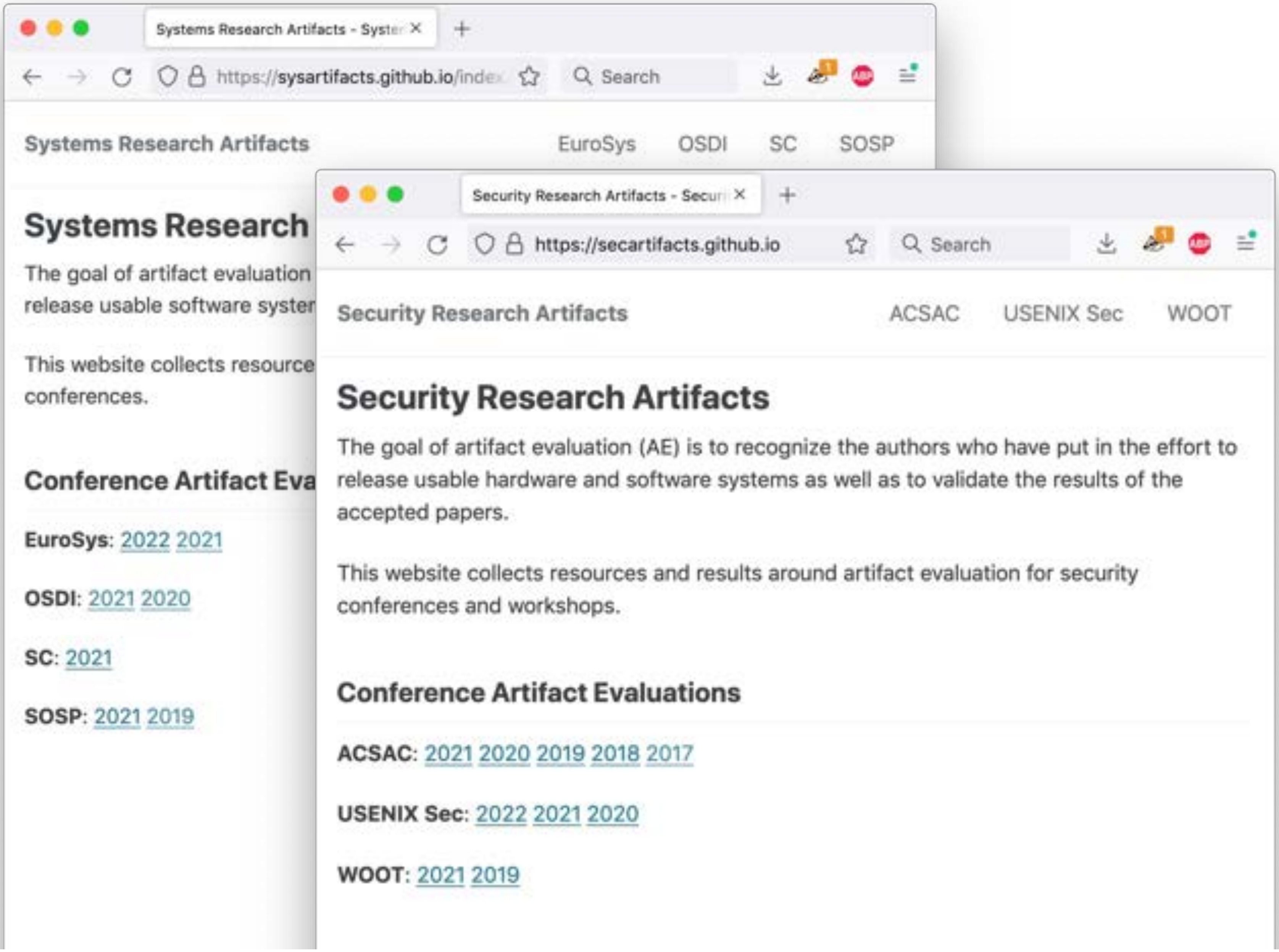
https://sysartifacts.github.io/



https://secartifacts.github.io/

Artifact evaluation indexes

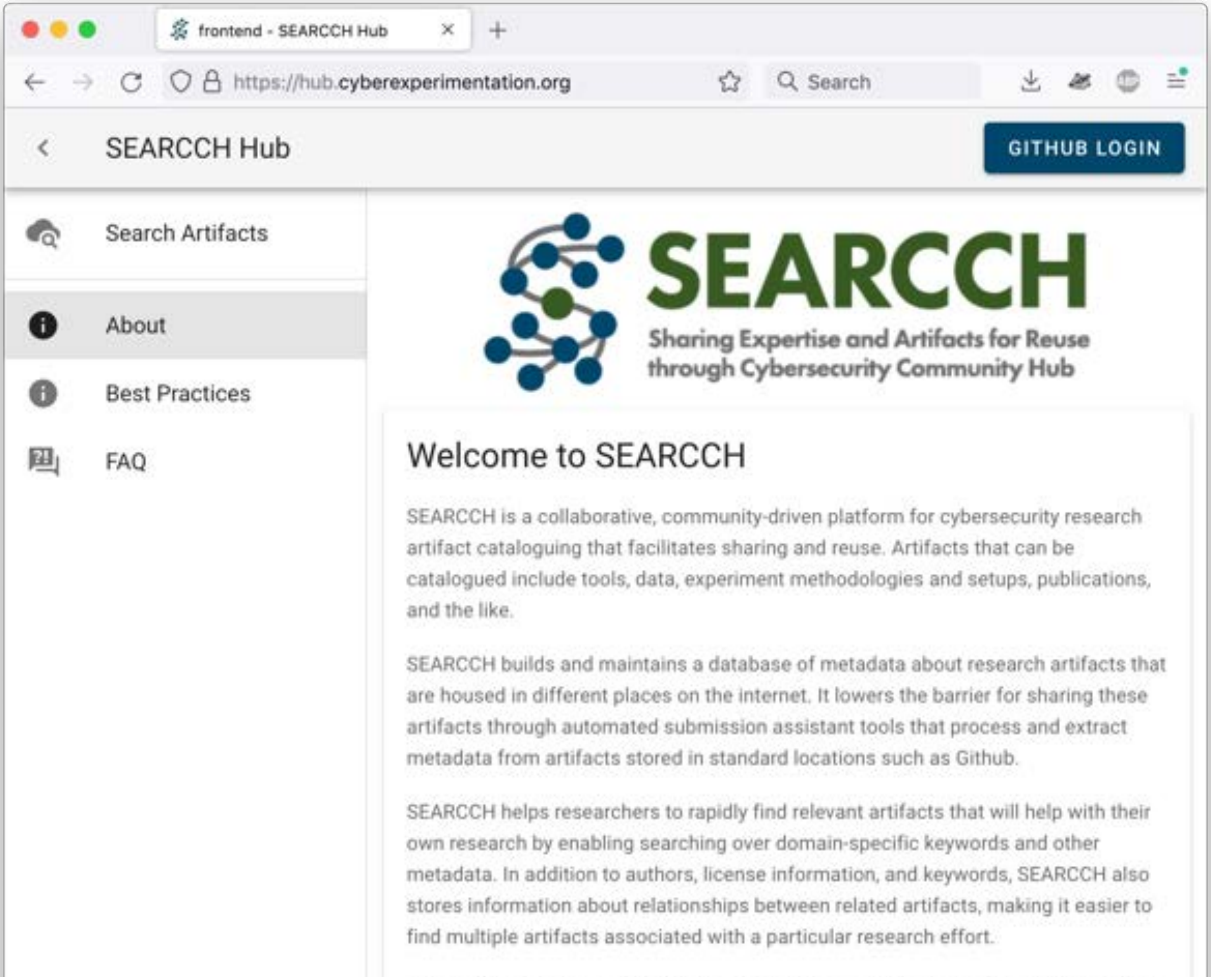
https://sysartifacts.github.io/



https://secartifacts.github.io/

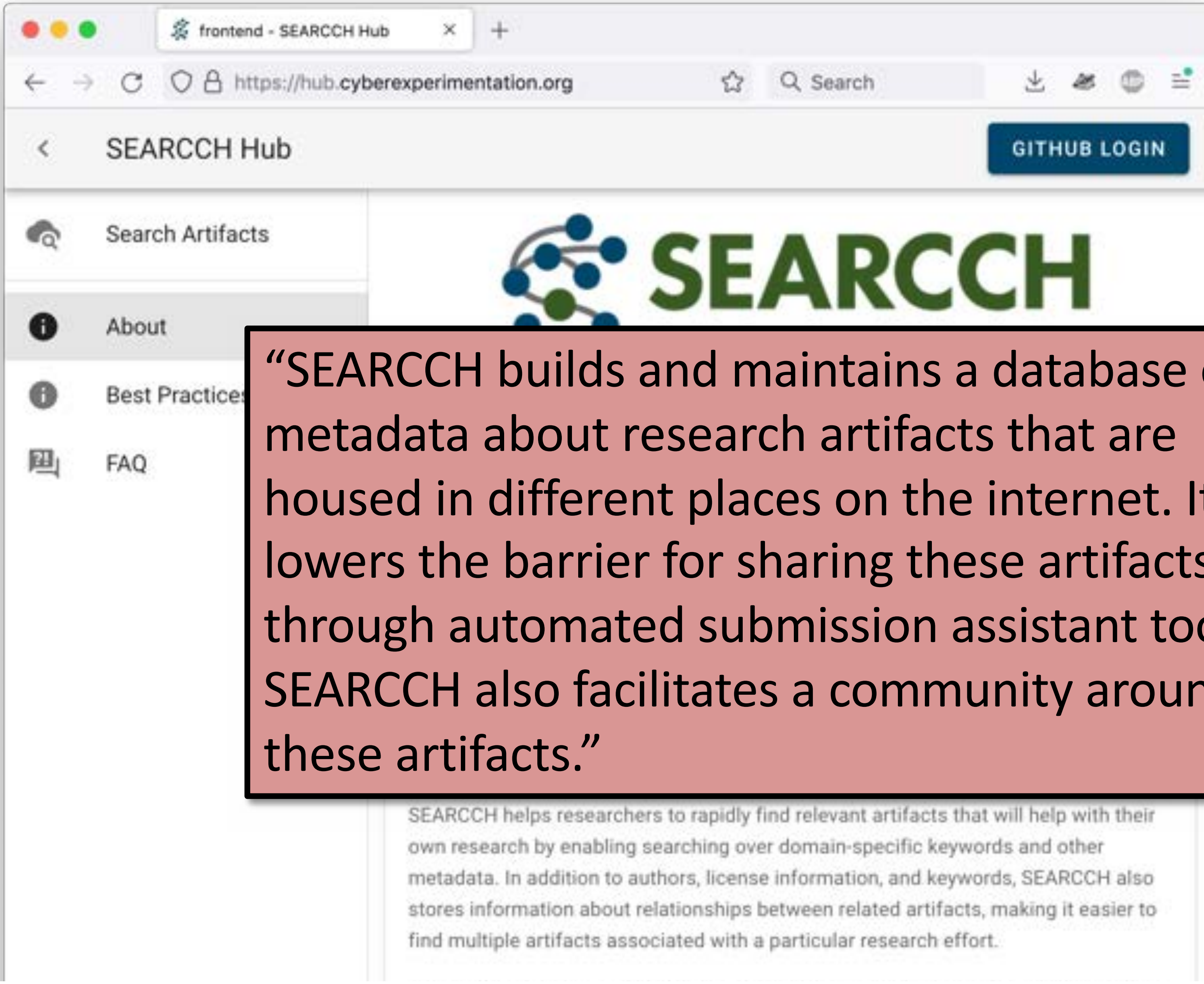
Community artifact hubs

https://hub.cyberexperimentation.org/



Community artifact hubs

https://hub.cyberexperimentation.org/



"SEARCCH builds and maintains a database of metadata about research artifacts that are housed in different places on the internet. It lowers the barrier for sharing these artifacts through automated submission assistant tools… SEARCCH also facilitates a community around these artifacts."

Summary

- artifact evaluation has changed our practices and expectations
- slowly moving toward "standard" practices…
- …but many issues still to be addressed

Eric Eide
www.cs.utah.edu/~eeide/
email: eeide@cs.utah.edu
Twitter: @eeide