Improving Reproducibility Through Better Software Practices

Presented at
ATPESC 2018

David E. Bernholdt
Distinguished R&D Staff Member and Group Leader
Oak Ridge National Laboratory

License, citation, and acknowledgments

License and Citation

- Requested citation: Michael A. Heroux and David E. Bernholdt, Improving Reproducibility Through Better Software Practices, in Argonne Training Program on Extreme-Scale Computing (ATPESC) 2018. DOI: 10.6084/m9.figshare.6936050.

Acknowledgements

- This work was supported by the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research (ASCR), and by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.
- Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.
- This work was performed in part at the Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC for the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

IDEAS productivity

ECP EXASCALE COMPUTING PROJECT

Outline

- Increasing focus on reproducibility.
- Publication requirements.
- Trustworthiness at Scale.
- Role of better software practices.
- Personal Productivity Commitment.

Reproducibility is essential

Fundamental to the scientific process, and to the credibility of computational results

SCIENCE

Many Psychology Findings Not as Strong as Claimed

By BENEDICT CAREY AUG. 27, 2015

Staff of the the Reproducibility Project at the Center for Open Science in Charlottesville, Va., from left: Mallory Kidwell, Courtney Soderberg, Johanna Cohoon and Brian Nosek. Dr. Nosek and his team led an attempt to replicate the findings of 100 social science studies. Andrew Shurtleff for The New York Times

Reproducibility

- NY Times highlights "problems".
- Only one of many cited examples.
- Feeds public distrust of "science"
- HPC has been spared this "spotlight" (so far).
- Lots of activity:
  - AAAS, ACM initiatives.
  - PPoPP, Supercomputing 2017.
- But what is reproducibility?

http://www.nytimes.com/2015/08/28/science/many-social-science-findings-not-as-strong-as-claimed-study-says.html?_r=0

Reproducibility Terminology

V. Stodden, D. H. Bailey, J. Borwein, R. J. LeVeque, W. Rider, and W. Stein. 2013. Setting the Default to Reproducible: Reproducibility in Computational and Experimental Mathematics. (2013). https://icerm.brown.edu/topical_workshops/tw12-5-rcem/icerm_report.pdf

- **Reviewable Research.** The descriptions of the research methods can be independently assessed and the results judged credible. (This includes both traditional peer review and community review, and does not necessarily imply reproducibility.)
- **Replicable Research.** Tools are made available that would allow one to duplicate the results of the research, for example by running the authors' code to produce the plots shown in the publication. (Here tools might be limited in scope, e.g., only essential data or executables, and might only be made available to referees or only upon request.)
- **Confirmable Research.** The main conclusions of the research can be attained independently without the use of software provided by the author. (But using the complete description of algorithms and methodology provided in the publication and any supplementary materials.)
- **Auditable Research.** Sufficient records (including data and software) have been archived so that the research can be defended later if necessary or differences between independent confirmations resolved. The archive might be private, as with traditional laboratory notebooks.
- **Open or Reproducible Research.** Auditable research made openly available. This comprised well-documented and fully open code and data that are publicly available that would allow one to (a) fully audit the computational procedure, (b) replicate and also independently reproduce the results of the research, and (c) extend the results or apply the method to new problems.

Note: These are the definitions used by ACM. Other organizations/communities have different (conflicting) definitions.

ACM TOMS Replicated Computational Results (RCR)

- Submission: Optional RCR option.
- Standard reviewer assignment: Nothing changes.
- RCR reviewer assignment:
  - Concurrent with standard reviews.
  - As early as possible in review process.
  - Known to and works with authors during the RCR process.
- RCR process:
  - Multi-faceted approach, Bottom line: Trust the reviewer.
- Publication:
  - Replicated Computational Results Designation.
  - The RCR referee acknowledged.
  - Review report appears with published manuscript.

RCR Process: Two Basic Approaches

1. Independent replication (3 options):
   A. Transfer of, or pointer to, author's software.
   B. Guest account, access to author's software.
   C. Observation of authors replicating results.

Or (Untested, rare)

2. Review of computational results artifacts:
   – Results may be from an unavailable system.
   – Leadership class computing system.
   – In this situation:
     • Careful documentation of the process.
     • Software should have its own substantial V&V process.

TOMS:

- First RCR paper in TOMS issue 41:3
  - Editorial introduction.
  - van Zee & van de Geijn, BLIS paper.
  - Referee report.
- Second: TOMS 42:1
  - Hogg & Scott.
- Third: TOMS 42:4.
- More in the meantime.

TOMACS

- Similar.

Big Picture of ACM RCR

- Improve science.
  - Quality of prose: Good.
  - Quality of data: Poor.
- So bad now:
  - Trust comes from seeing a “cloud” of similar papers with similar results.
  - Which could still be wrong (built on a common bad piece).
  - Replicability: First step toward improvement.
- Engage a “dark portion” of the R&D community.
  - Reviewers not among typical reviewer pool.
  - Practitioners, users. Expert at use of Math SW.

Thank you for taking the time to consider our paper for your journal.

XXX has agreed to undergo the RCR process should the paper proceed far enough in the review process to qualify. ***To make this easier we have preserved the exact copy of the code used for the results (including additional code for generating detailed statistics that is not in the library version of the code).***

SC18 Reproducibility Initiative

- Two appendices:
  - Artifact description (AD).
    - Blue print for setting up your computational experiment.
    - Makes it easier to rerun computations in future.
    - AD appendix will be mandatory for SC19 paper submissions.
  - Artifact Evaluation (AE).
    - Targets ”boutique” environments.
    - Improves trustworthiness when re-running hard, impossible.
- Details:
  - https://collegeville.github.io/sc-reproducibility/

Coming to Your World Soon: Reproducibility Requirements

- These conferences expect artifact evaluation appendices (most optionally):
  - CGO, PPoPP, PACT, RTSS and SC.
  - http://fursin.net/reproducibility.html
- ACM Replicated Computational Results (RCR).
  - ACM TOMS, TOMACS.
  - http://toms.acm.org/replicated-computational-results.cfm
- ACM Badging.
  - https://www.acm.org/publications/policies/artifact-review-badging

How can you prepare?

Improving Trustworthiness at Scale

What if we can’t re-run a computational experiment?

Reproducibility and Supercomputing

Scenario:
You compute a “hero” calculation using 5M core-hours on Mira and submit your results for publication. During the review process, a referee questions the validity of your results.

What options are feasible:

- The reviewer re-runs your code on a laptop or cluster.
- The reviewer re-runs your code on Mira.
- You re-run your code on Mira.
- Your results are rejected.
- Your results are accepted, but with risk.

Sources for Artifact Evaluation metrics

- Synthetic operators with known:
  - Spectrum (Huge diagonals).
  - Rank (by constructions).
- Invariant subspaces:
  - Example: Positional/rotational invariance (structures).
- Conservation principles:
  - Example: Flux through a finite volume.
- General:
  - Pre-conditions, post-conditions, invariants.

Can you think of something for your problems?

Productivity and Sustainability

Synergistic with Reproducibility

Objectives

- Productivity – Output per unit input.
- Sustainability – The future cost of usability.
- Goals for today:
  - Learn how to improve
    - Developer productivity.
    - Software sustainability.
  - For the purposes of better scientific productivity – and reproducibility,
  - Using tools, processes and practices.

Tradeoffs: Better, faster, cheaper

“Better, faster, cheaper: Pick two of the three.”

- Scenario: (Today)
  You are behind in developing a sophisticated new model in your software that you want to use for results in an upcoming paper.
- Which of these could be reasonable choices?
  - Develop a simpler model for the paper.
  - Set other work aside and spend more time on development.
  - Ask for an extension on the paper deadline.
  - Develop sophisticated model, but don’t test its correctness.
  - Develop sophisticated model, but don’t document it or check it in.

Improved developer productivity

"Better, faster, cheaper: Pick all three." – Near term.

- Scenario: (6 months later)
  After investing in developer productivity improvements, you are on time in developing a sophisticated new model in your software that you want to use for results in an upcoming paper.

- Invest in developer tools, processes, practices.

Improved software sustainability

"Better, faster, cheaper: Pick all three." – Long term.

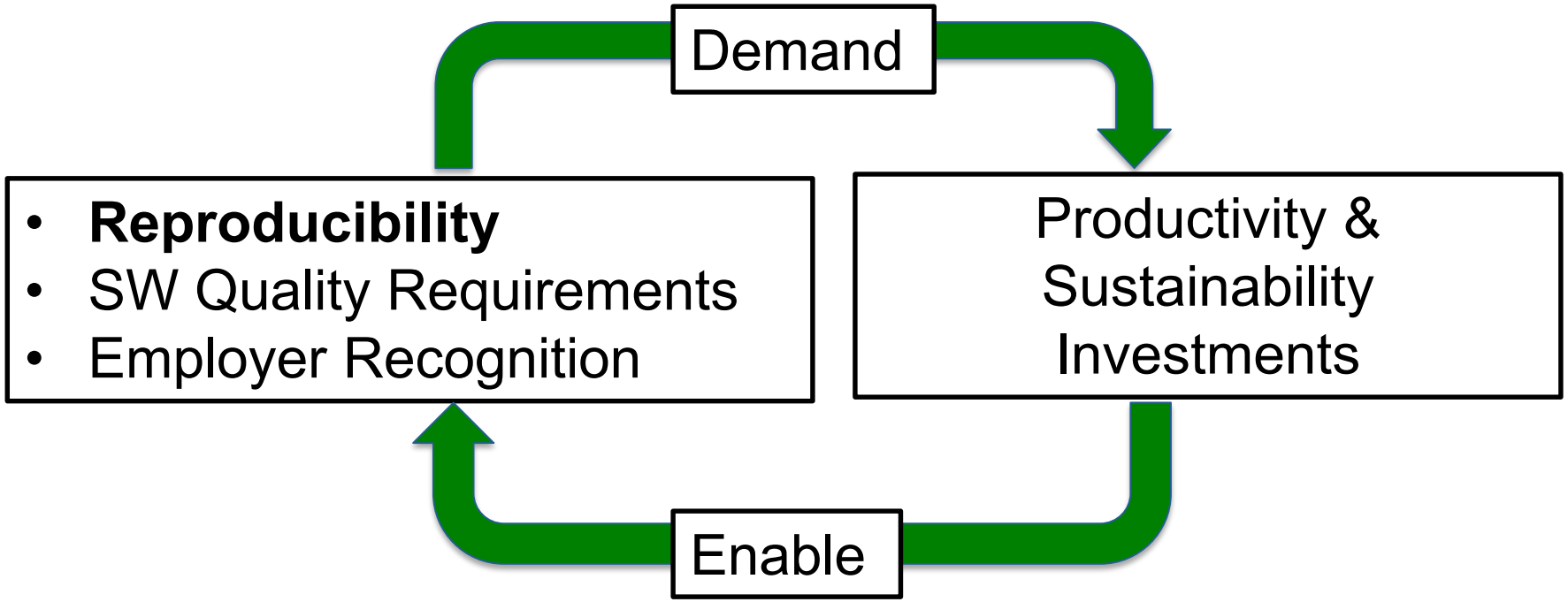
- Scenario: (3 years later)
  After investing in software sustainability improvements, you are on time in developing several sophisticated new models in your software that you want to use for results in upcoming papers.

- Invest in testing, documentation, integration for long-term software usability.

Which of These Enhance Reproducibility?

- Code written by first-year, untrained grad student.
- Tuning for high performance.
- Dynamic parallelism of modern processors.
- Better software testing.
- Source code and versioning management.
- Investing in developer productivity.
- Investing in software sustainability.

Incentives To Change



Common statement: “I would love to do a better job, but I need to:

- Get this paper submitted.
- Complete this project task.
- Do something my employer values more.

Goal: Change incentives to include value of better software.

Personal Expectations

Calling out the best in team members

A Few Concrete Recommendations

> *Show me the person making the most commits on an undisciplined software project and I will show you the person who is injecting the most technical debt.*
>
> *-- Mike Heroux*

- GitHub stats: Easy to find who made the most commits.
  - Some people: Pride in their high ranking.

- Instead, be the person who ranks high in these ways:
  - Writes up requirements, analysis and design, even if simple.
  - Writes good GitHub issues, tracks their progress to completion.
  - Comments on, tests and accepts pull requests.
  - Provide good wiki, gh-pages content, responses to user issues.

(Personal) Productivity++ Initiative

Ask: *Is My Work _____ ?*

https://github.com/trilinos/Trilinos/wiki/Productivity---Initiative

Summary

- Reproducibility demands are coming.
  - Conferences first, journals slower.
- HPC software is particularly challenging:
  - Hardware variation.
  - Code optimization.
  - Dynamic parallelism.
- Better software practices:
  - Improve chances for reproducibility.
  - Lower its cost.
- Many tools emerging to enable reproducibility.

Other Resources

Editorial: ACM TOMS Replicated Computational Results Initiative. Michael A. Heroux. 2015. *ACM Trans. Math. Softw.* 41, 3, Article 13 (June 2015), 5 pages. DOI: http://dx.doi.org/10.1145/2743015

Enhancing Reproducibility for Computational Methods. Victoria Stodden, Marcia McNutt, David H. Bailey, Ewa Deelman, Yolanda Gil, Brooks Hanson, Michael A. Heroux, John P.A. Ioannidis, Michela Taufer Science (09 Dec 2016), pp. 1240-1241