HPC Complete: Reproducible, Sustainable, Productive

Michael A. Heroux
Center for Computing Research
Sandia National Laboratories

Sandia National Laboratories is a multi-program laboratory operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin company, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Outline

- Context, definitions.
- How the future will be different (IMO).
- Completeness philosophy.
- Reproducibility.
- Sustainability.
- Productivity.
- Some Practical Strategies.

From the US NSCI Announcement (Fact sheet):

Improve HPC application developer productivity.

Current HPC systems are very difficult to program, requiring careful measurement and tuning to get maximum performance on the targeted machine. Shifting a program to a new machine can require repeating much of this process, and it also requires making sure the new code gets the same results as the old code. The level of expertise and effort required to develop HPC applications poses a major barrier to their widespread use.

Government agencies will support research on new approaches to building and programming HPC systems that make it possible to express programs at more abstract levels and then automatically map them onto specific machines. In working with vendors, agencies will emphasize the importance of programmer productivity as a design objective. Agencies will foster the transition of improved programming tools into actual practice, making the development of applications for HPC systems no more difficult than it is for other classes of large-scale systems.

https://www.whitehouse.gov/sites/default/files/microsites/ostp/nsci_fact_sheet.pdf

Reproducible

Real result, not coincidence or mistake.

Productive

Better, Faster, Cheaper: Pick all three

Sustainable

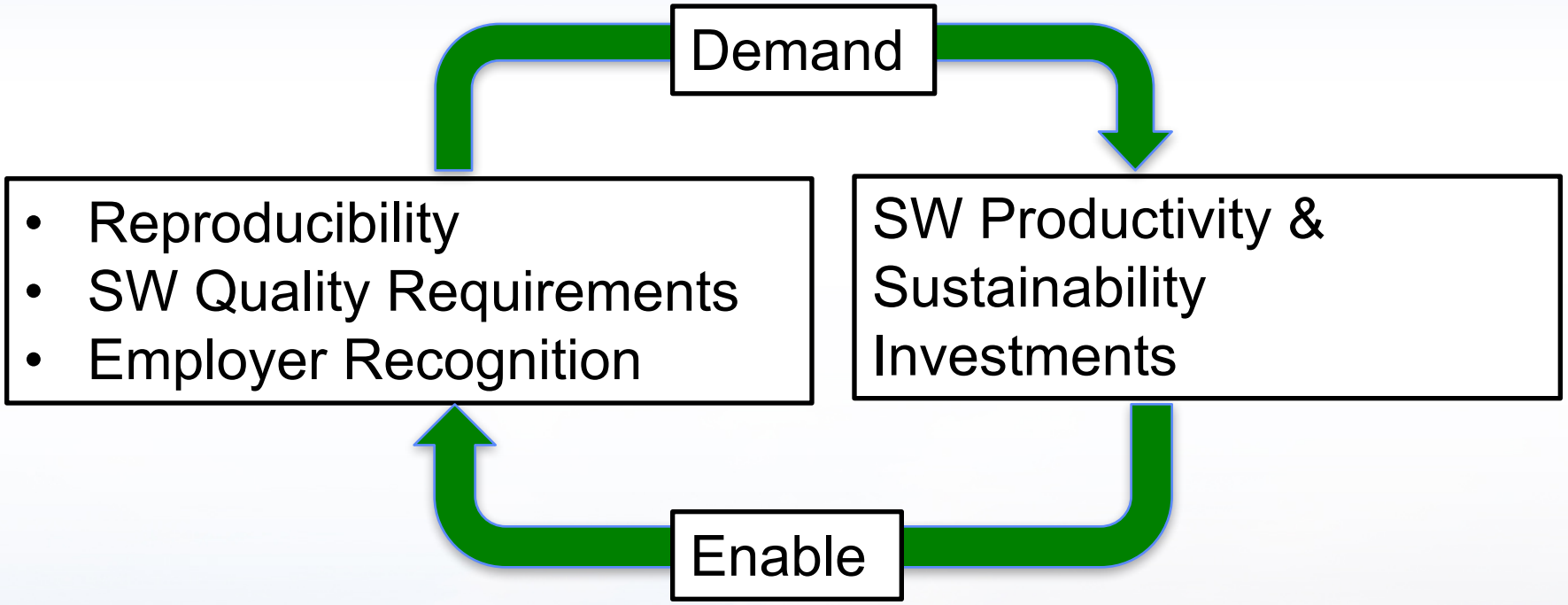
Code usable for expected SW lifetime

How the Future will be Different

- Publishers:
  - Will expect reproducible computational results.
- Funding agencies:
  - Will expect improved productivity, sustainable software.
- Employers:
  - Will reward staff, faculty producing good software.

- Impact: High-quality HPC software will matter a lot more.

Incentives To Change



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

Get this paper submitted.

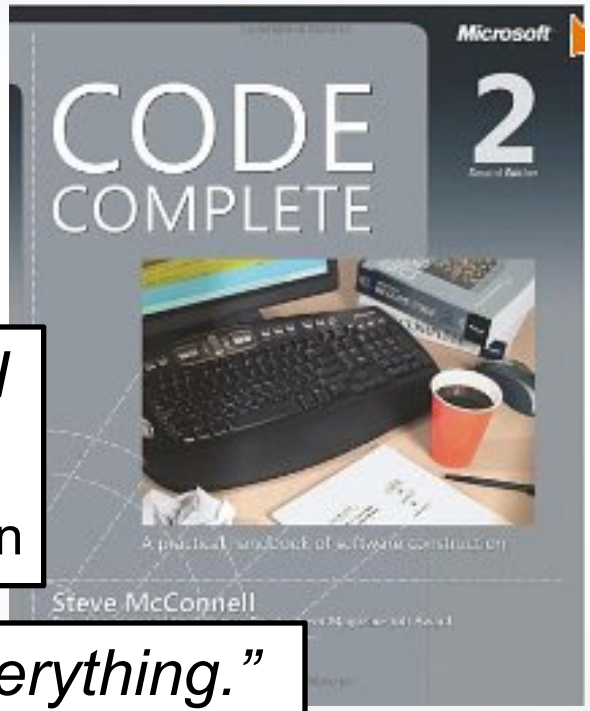
Complete this project task.

Do something my employer values more.

Need to change incentives to include value of better software.

HPC Complete: Useful "Overhead"

- Code Complete: Ultimate value is code.
  - Should we only write code?
  - Some non-coding activities improve code.



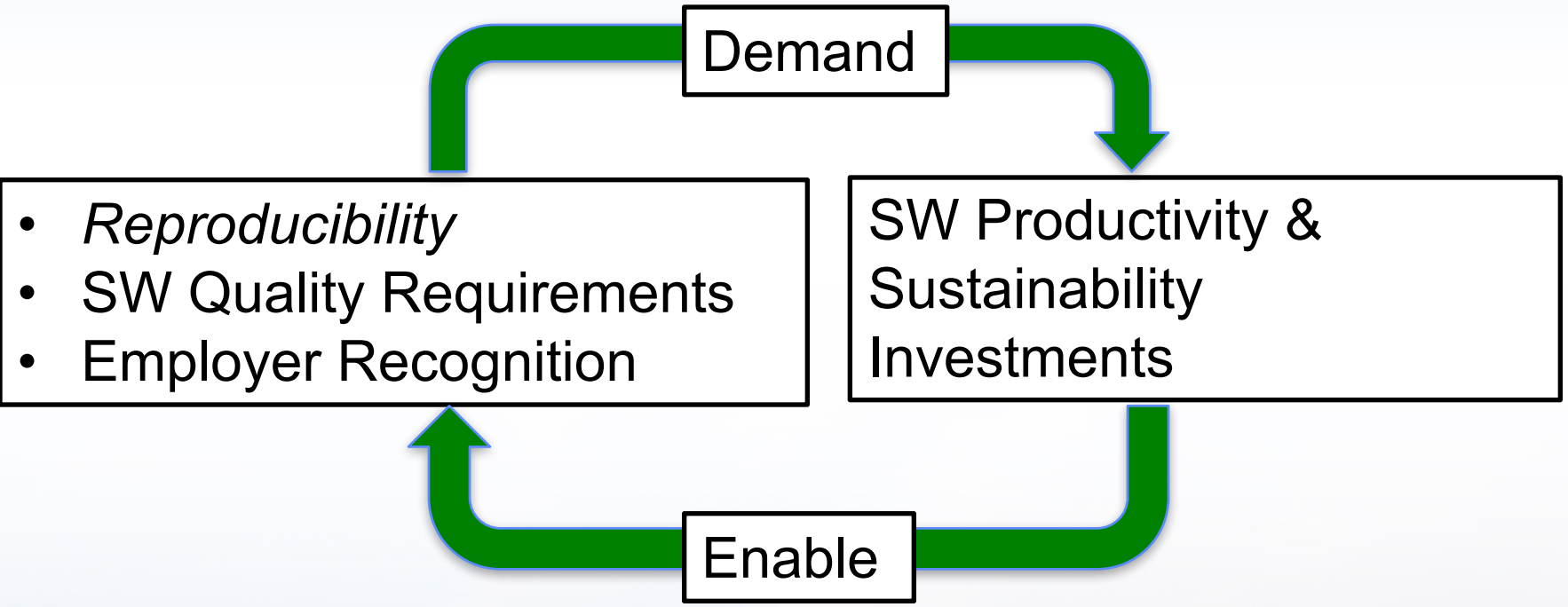
> *"Give me six hours to chop down a tree and I will spend the first four sharpening the axe."*
>
> Abraham Lincoln

> *"Plans are worthless, but planning is everything."*
>
> Dwight D. Eisenhower

- HPC Complete: Ultimate value is HPC.
  - What non-HPC activities improve HPC?

- Let's talk about a few of these…

Incentives To Change



Questions you might ask:

What is reproducibility?

How can we improve it?

What does this mean for me?

Reproducibility

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

- NY Times highlights "problems".
- Only one of many cited examples.
- HPC has been spared this "spotlight" (so far).

- Lots of activity:
  - AAAS, ACM initiatives.
  - PPoPP, Supercomputing 2016.
- 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). http://icerm.brown.edu/html/programs/topical/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.

ACM Transactions on
Mathematical Software

ACM TOMS

- TOMS RCR Initiative: Referee Data.
- Why TOMS? Tradition of real software that others use.
- Two categories: Algorithms, Research.
- TOMS Algorithms Category:
  - Software Submitted with manuscript.
  - Both are thoroughly reviewed.
- TOMS Research Category:
  - Stronger: Previous implicit “real software” requirement is explicit.
  - New: Special designation for replicated results.

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.
- Others ID'ed.

TOMACS

- Initial article complete.
- 4 in the system.

Big Picture of TOMS RCR

- Improve science.
  - Quality of prose in publications: Good.
  - Quality of data: (Very!) 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.

Reproducibility Status & (Some) Futures

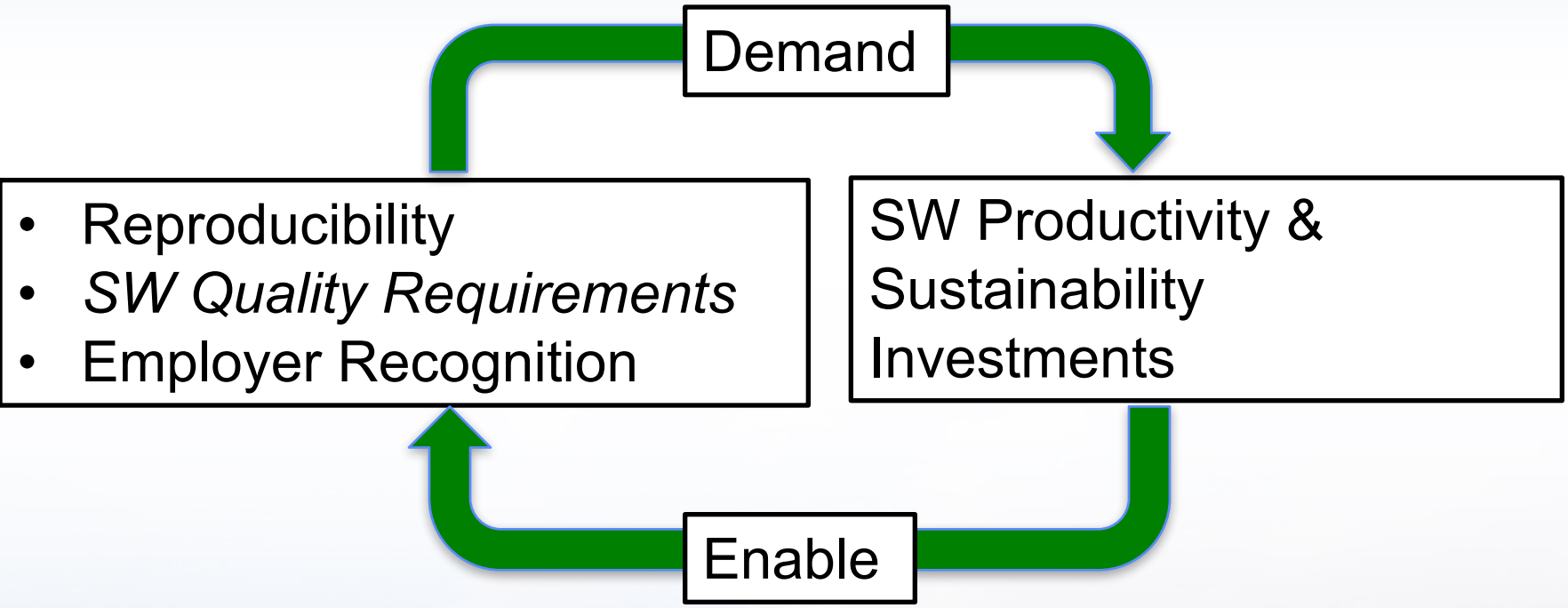
- TOMACS: Adopted TOMS RCR.
- ACM: Completed electronic workflow support.
- Conference proceedings:
  - PPoPP, other conferences, reviewing artifacts.
  - SC16, 17 and beyond. Progressive increase of results review.
- AAAS:
  - *Science* paper (from 3rd Arnold Workshop on Reproducibile Science):
    - *The "REP" Standards for Disclosing Computational Methods*
      Victoria Stodden, Marcia McNutt, David H. Bailey, Ewa Deelman, Yolanda Gil, Brooks Hanson, Michael A. Heroux, John P.A. Ioannidis, Michela Taufer*, submitted, July 2016.*
    - REP = Reproducibility Enhancement Principles

Message to This Audience

Be prepared to have someone else replicate your results.

Create, retain artifacts that establish credible results.

Incentives & Productivity/Sustainability



New funding proposal element:

SW Productivity and Sustainability Plan

Your questions:

What's a SW Productivity and Sustainability Plan?

How do I prepare for productivity and sustainability?

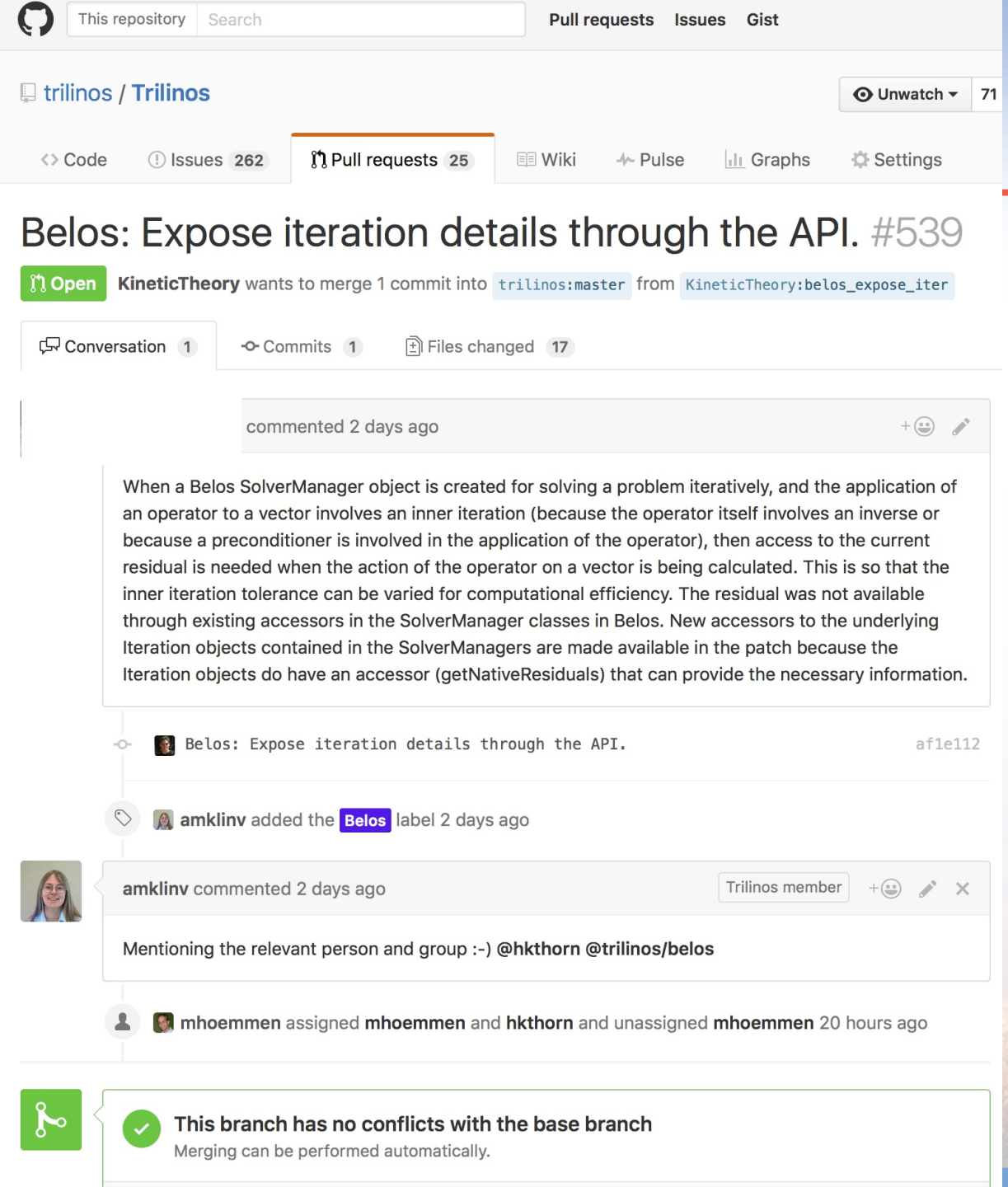
DOE SW Productivity and Sustainability Plan (SW PSP).

No viable SW plan, no money (eventually)

- Key Entities:
  - DOE Biological and Environmental Research (BER).
  - DOE Advanced Scientific Computing Research (ASCR)
  - IDEAS Scientific SW Productivity Project
- Milestone:
  - First-of-a-kind SW Productivity and Sustainability Plan.

DOE BER SW PSP Requirements (I)

- Describe overall SW development process.
  - Software lifecycle, testing, documentation and training.
- Development tools and processes:
  - source management, issue tracking, regression testing, SW distribution.
- Training and transition:
  - New and departing team members.
- Continuous process improvement:
  - Getting better at productivity and sustainability.



Sustainable?

Pull request scenario

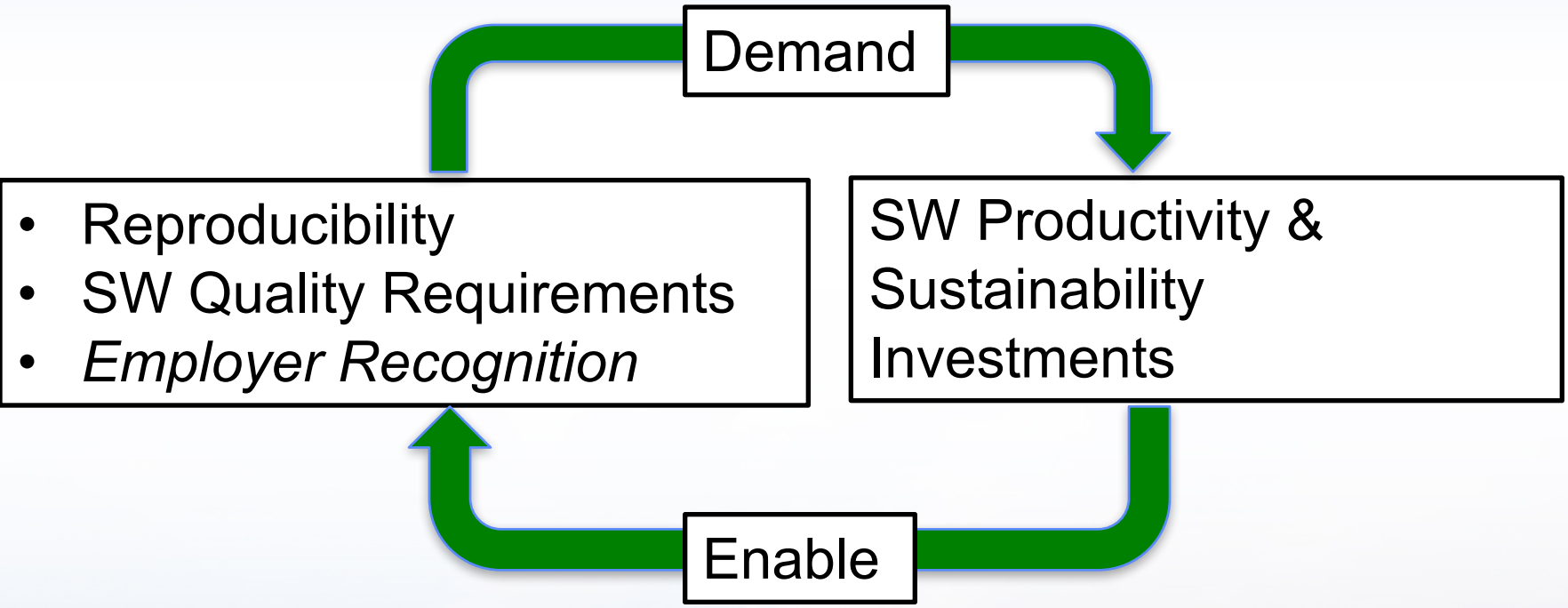
- Pull request.
- Competent external user.
- Useful feature.
- No conflicts.
- Accept: Yes, if OK.
- What must happen?
  - Vet commit:
    - Comes with tests.
    - Comes with docs.
    - Follows style rules.
  - Test:
    - No regressions.
    - Portable.

Sandia National Laboratories

Message to This Audience

Write tests now, while (or before) writing your intended production software.

Employers & Productivity/Sustainability



Next focus:

- Work with DOE labs to recognize SW as base research.
- Pressing for SW Methodologies base funding.

Message to This Audience

Advocate for the value of high quality software in your life:

- *Cite software.*
- *Promote with your management. Evaluate SW quality in reviews.*
- *Promote the need for SW methodologies research.*

High-quality Software:
HPC Challenges
A Few Practical Suggestions

Software Engineering and HPC: Efficiency vs Other Quality Metrics

How focusing on the factor below affects the factor to the right	Correctness	Usability	Efficiency	Reliability	Integrity	Adaptability	Accuracy	Robustness
Correctness	↑		↑	↑			↑	↓
Usability		↑				↑	↑	
Efficiency	↓		↑	↓	↓	↓	↓	
Reliability	↑			↑	↑		↑	↓
Integrity			↓	↑	↑			
Adaptability					↓	↑		↑
Accuracy	↑		↓	↑		↓	↑	↓
Robustness	↓	↑	↓	↓	↓	↑	↓	↑

Source:
Code Complete
Steve McConnell

Helps it ↑

Hurts it ↓

Scientific Software Engineering

Teaming

"A scientist builds in order to learn;
an engineer learns in order to build."

- Fred Brooks

Scientist: Barely-sufficient building.

Engineer: Barely-sufficient learning.

Both: Insufficiency leads to poor SW.

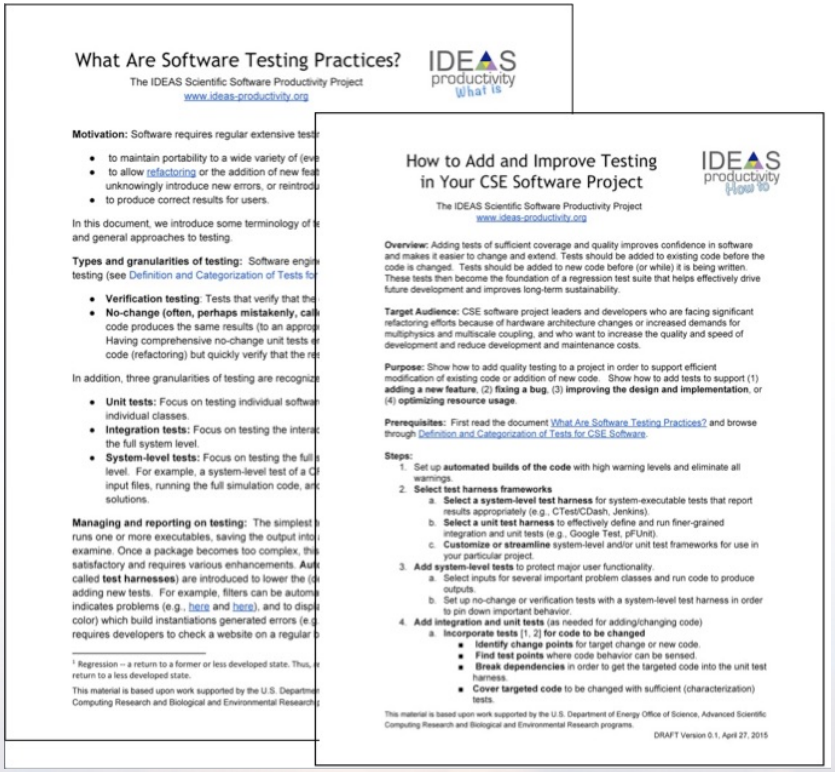
Teams: Can complement each other.

IDEAS ‘What is’ and ‘How to’ docs

Best Practices

- Motivation: Scientific software teams have a wide range of levels of maturity in software engineering practices
  - Baseline survey of *xSDK* and *BER Use Case* teams
- Approach:
  - ‘What Is’ docs: 2-page characterizations of important software project topics
  - ‘How To’ docs: brief sketch of best practices
    - Emphasis on ``bite-sized'' topics enables CSE software teams to consider improvements at a small but impactful scale.
- Initial emphasis:
  - *What is CSE Software Productivity?*
  - *What are Software Testing Practices?*
  - *How to Add and Improve Testing in Your CSE Software Project*
- Topics in progress:
  - Refactoring tools and approaches
  - Best practices for using interoperable libraries
  - Designing for performance portability
  - Etc.

https://ideas-productivity.org/resources/howtos



Impact: Provide baseline nomenclature and foundation for next steps in SW productivity and SW engineering for CSE teams

Managing issues: Fundamental software process

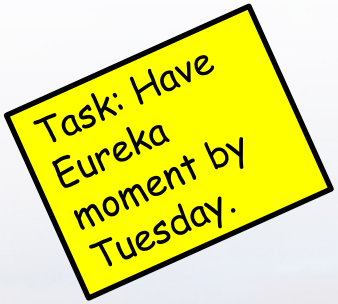
Continual improvement

- Issue: Bug report, feature request
- Approaches:
  - Short-term memory, office notepad
  - ToDo.txt on computer desktop (1 person)
  - Issues.txt in repository root (small co-located team)
  - …
  - Web-based tool + Kanban (distributed, larger team)
  - Web-based tool + Scrum (full-time dev team)

- IDEAS project:
  - Jira Agile + Confluence: Turnkey web platform (ACME too)
  - Kanban: Simplest of widely known Agile SW dev processes

Kanban principles

- Limit number of “In Progress” tasks
- Productivity improvement:
  - Optimize “flexibility vs swap overhead” balance. No overcommitting.
  - Productivity weakness exposed as bottleneck. Team must identify and fix the bottleneck.
  - Effective in R&D setting. Avoids a deadline-based approach. Deadlines are dealt with in a different way.
- Provides a board for viewing and managing issues



IDEAS Confluence, Jira Agile, Kanban

Pages / IDEAS Common Home

Managing IDEAS Project Activities Using JIRA Agile and Kanban

Created by Michael Heroux, last modified by Jim Willenbring about an hour ago

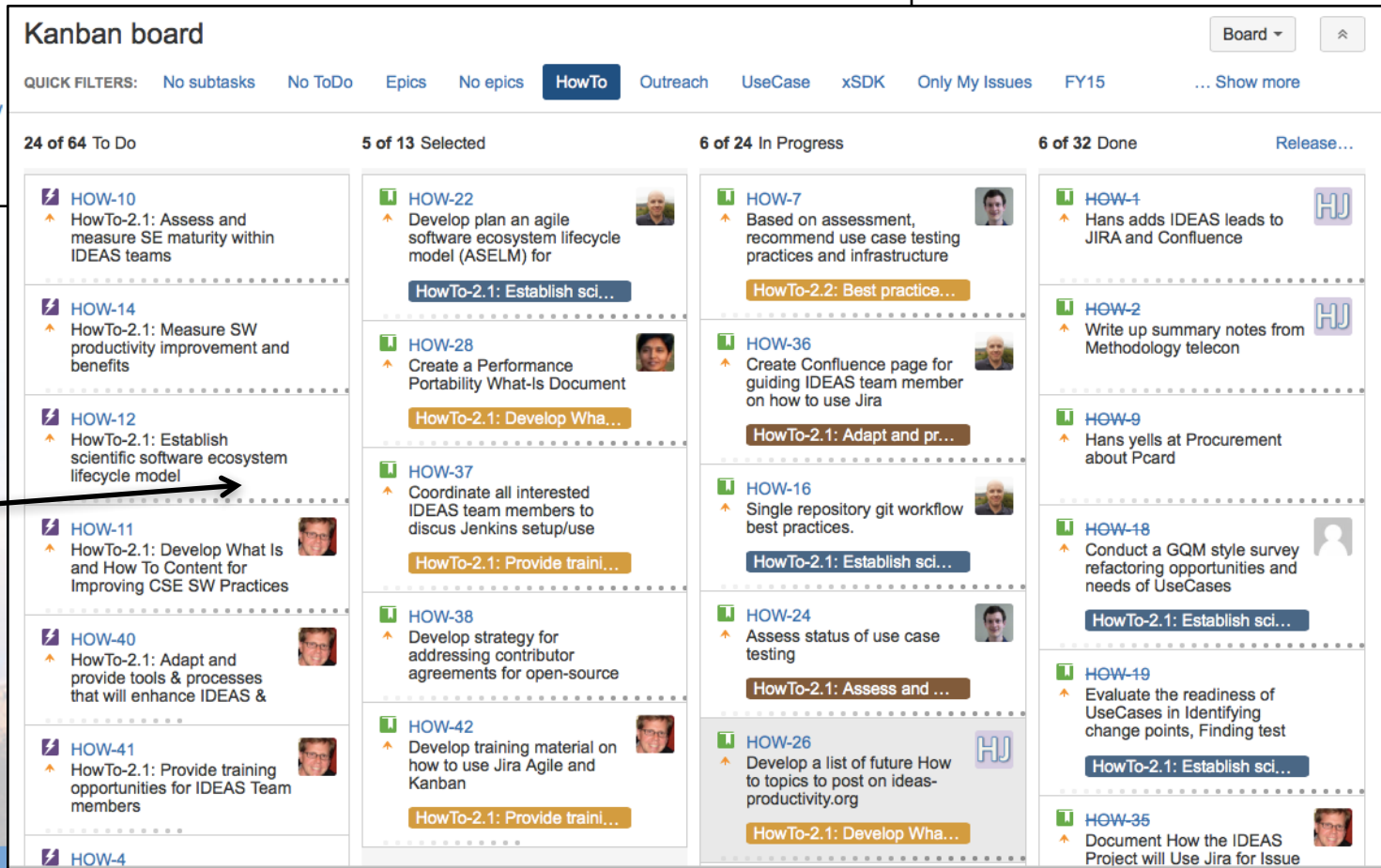
Contents:

Developer Guide, on Confluence site

Kanban Board, on Jira site.

Four columns:

- To Do
- Selected
- In Progress
- Done



Message to This Audience

HPC Software is particularly susceptible to quality challenges.

Scientists and engineers can complement.

Seek best practices: Bite-sized strategy.

Message to This Audience

Embrace continual improvement.
Example:

Improve your issue tracking habits:

- *Nothing -> Desktop/todo.txt*
- *Desktop/todo.txt -> clone/todo.txt*
- *clone/todo.txt -> Git Issues*
- *Git Issues -> Git Issues + Kanban*

or Jira + Kanban

Summary

- HPC is not complete without incorporating better:
  - Reproducibility: Can your results be trusted?
  - Sustainability: Will your ecosystem live long and prosper?
  - Productivity: Will your software development efforts be competitive?
- SW engineering focus is important for HPC:
  - Pursuing efficiency negatively impacts many other quality metrics.
  - Improve incrementally: Teaming, best practices, personal best.
- Focus on productivity cannot take precedence over high value HPC work:
  - High quality, low value software cannot be the outcome.
  - Quality improvements must be measured.
- Changes in publishers, funders, employers expectations:
  - Could be viewed as bothersome: More “overhead” in the way of progress.
  - Could be viewed as opportunity: Be proactive, part of the quality improvement initiative.

Final Thought: Commitment to Quality

Canadian engineers' oath (taken from Rudyard Kipling):

My Time I will not refuse;
my Thought I will not grudge;
my Care I will not deny
toward the honour, use,
stability and perfection of
any works to which I may be
called to set my hand.

http://commons.bcit.ca/update/2010/11/bcit-engineering-graduates-earn-their-iron-rings

Productivity++ Initiative

Ask: *Is My Work _____ ?*

Productivity++

- ✓ Traceable
- ✓ In Progress
- ✓ Sustainable
- ✓ Improved

Version 1.2

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