### 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.



Sandia National Laboratories

## 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



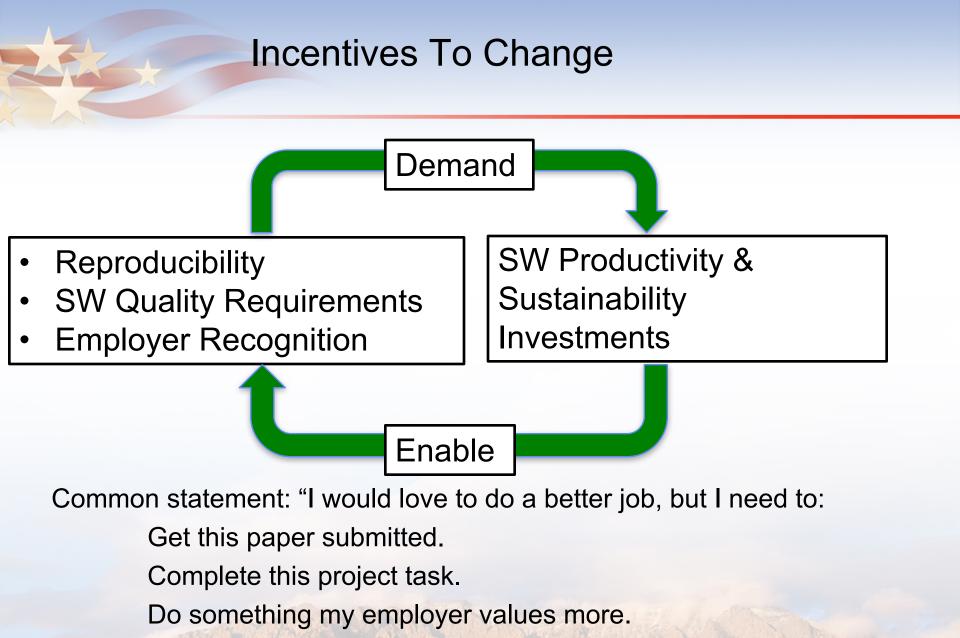
### How the Future will be Different

- 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.





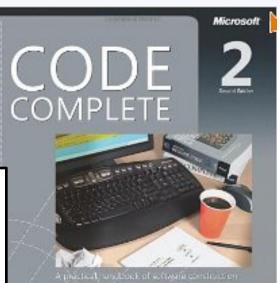
Need to change incentives to include value of better software.



# 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

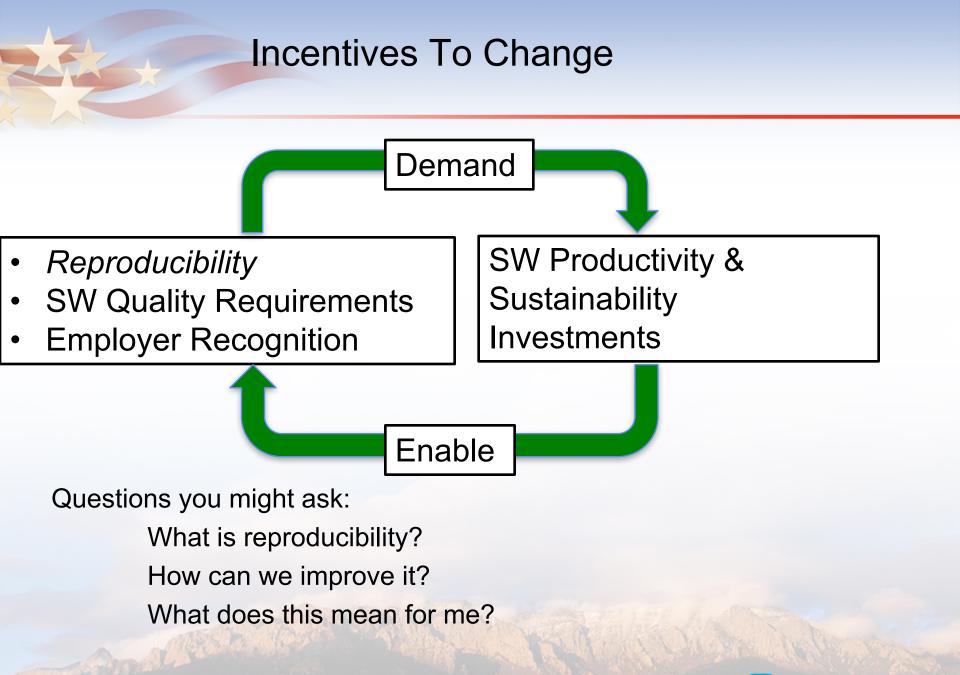


Steve McConnell

*"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...







SCIENCE

### Reproducibility

### 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**

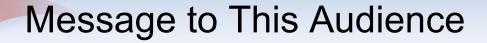
- Improve science.
  - Quality of prose in publications: Good.
  - Quality of data: (Very!) poor.
- So bad now:
  - Trust comes from seeing a "cloud" of similar papers with similar results.
  - Which could still be wrong (built on a common bad piece).
  - Replicability: First step toward improvement.
- 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

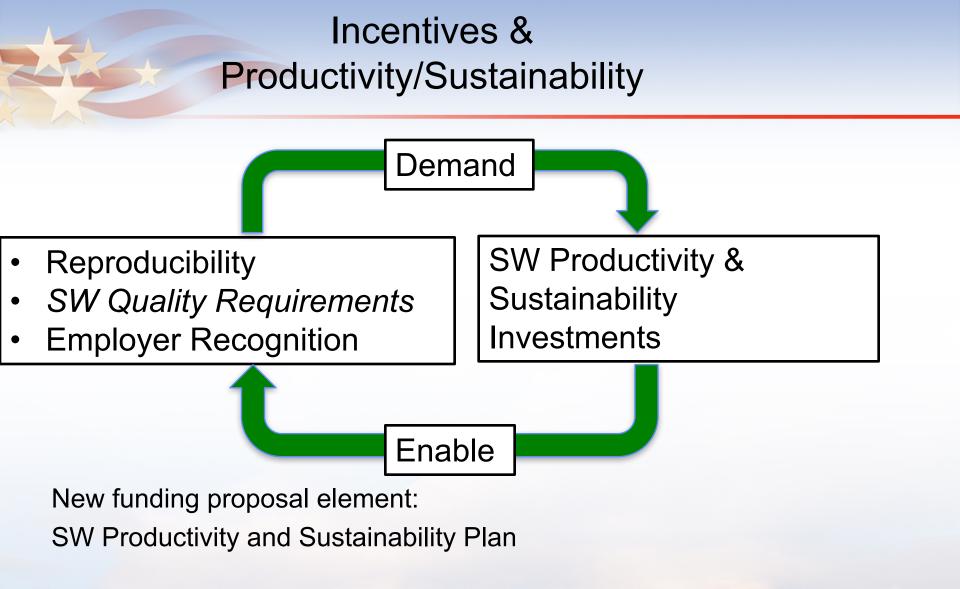
- TOMACS: Adopted TOMS RCR.
- ACM: Completed electronic workflow support.
- Conference proceedings:
  - PPoPP, other conferences, reviewing artifacts.
  - SC16, 17 and beyond. Progressive increase of results review.
- AAAS:
  - Science paper (from 3rd Arnold Workshop on Reproducibile Science):
    - 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





Be prepared to have someone else replicate your results. Create, retain artifacts that establish credible results.





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.



This repository Search		Pull re	Pull requests Issues Gist						
📮 trilinos / <b>Trilinos</b>					O Unwatch →	71			
Code Issues 262	î] Pull requests 25	🗉 Wiki	- Pulse	III Graphs	Settings				
Belos: Expose iteration details through the API. #539									

1) Open KineticTheory wants to merge 1 commit into trilinos:master from KineticTheory:belos\_expose\_iter

🖓 Conversation 🚺 🔸 🗠

- Commits 1 🗄 Files changed 17



When a Belos SolverManager object is created for solving a problem iteratively, and the application of an operator to a vector involves an inner iteration (because the operator itself involves an inverse or because a preconditioner is involved in the application of the operator), then access to the current residual is needed when the action of the operator on a vector is being calculated. This is so that the inner iteration tolerance can be varied for computational efficiency. The residual was not available through existing accessors in the SolverManager classes in Belos. New accessors to the underlying Iteration objects contained in the SolverManagers are made available in the patch because the Iteration objects do have an accessor (getNativeResiduals) that can provide the necessary information.

🔞 Belos: Expose iteration details through the API.							
S 🔊 amklinv added the Belos label 2 days ago							
amklinv commented 2 days ago	Trilinos member	+		×			
Mentioning the relevant person and group :-) @hkthorn @trilinos/belos							
Image: Second state in the second state is a second state in the second state is a second state in the second state is a second state i	mhoemmen 20 h	ours a	qo				

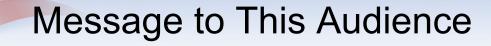
# Sustainable?

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.

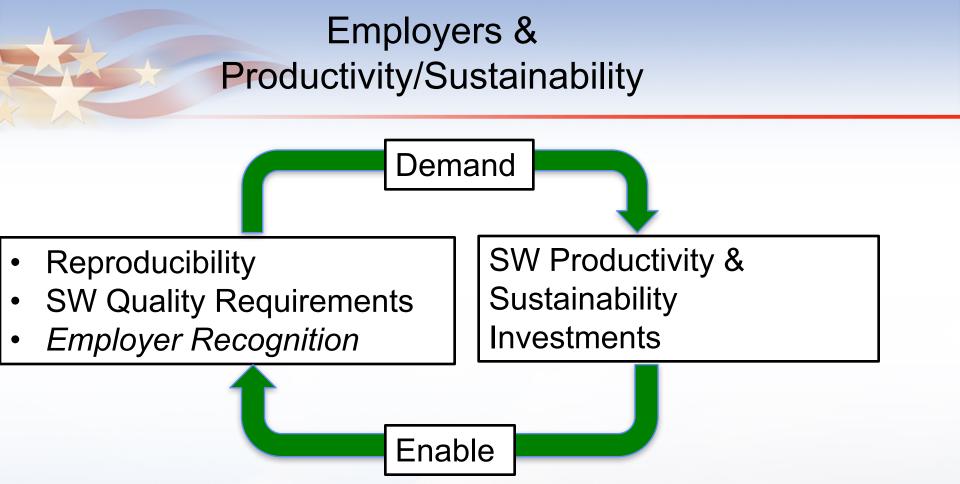


This branch has no conflicts with the base branch Merging can be performed automatically.



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





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	Source: <i>Code Complete</i> Steve McConnell
	Correctness	<b>●</b>	1	<u>⊢</u>	<u>⊢</u>	I	P	<u>→</u>	<b>→</b>	
	Usability	_	↑		_		↑	1	-	
$\langle$	Efficiency	¥		↑	¥	¥	♦	¥		$\mathbf{D}$
	Reliability	↑			1	↑		↑	¥	
	Integrity			¥	1	↑				
	Adaptability					✦	↑		↑	
	Accuracy	↑		¥	↑		¥	↑	¥	Helps it 🕈
	Robustness	¥	↑	¥	¥	¥	↑	¥	↑	Hurts it 븆

25

### Scientific Software Engineering



"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

27

### **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

### What Are Software Testing Practices? **IDE** The IDEAS Scientific Software Productivity Project productivity What is www.ideas-productivity.org Motivation: Software requires regular extensive to · to maintain portability to a wide variety of ( How to Add and Improve Testing IDEAS · to allow refactoring or the addition of new fea in Your CSE Software Project unknowingly introduce new errors, or reintrod to produce correct results for users. The IDEAS Scientific Software Productivity Project www.ideas-productivity.org In this document, we introduce some termino and general approaches to testing. Overview: Adding tests of sufficient coverage and quality improves confidence in software and makes it easier to change and extend. Tests should be added to existing code before the Types and granularities of testing: Software engi code is changed. Tests should be added to new code before (or while) it is being written testing (see Definition and Categorization of Tests for These tests then become the foundation of a regression test suite that helps effectively drive future development and improves long-term sustainability. Verification testing: Tests that verify that the Target Audience: CSE software project leaders and developers who are facing significant relactioning efforts because of hardware architecture changes or increased demands for multiphysics and multiscale coupling, and who want to increase the quality and speed of · No-change (often, perhaps mistakenly, cal code produces the same results (to an appro Having comprehensive no-change unit tests development and reduce development and maintenance costs code (refactoring) but quickly verify that the re Purpose: Show how to add quality testing to a project in order to support efficient modification of existing code or addition of new code. Show how to add tests to support (1) adding a new feature, (2) fixing a bug, (3) improving the design and implementation, or In addition, three granularities of testing are recog · Unit tests: Focus on testing individual softw (4) optimizing resource usage individual classes Prerequisites: First read the document <u>What Are Software Testing Practices?</u> and brows through <u>Definition and Categorization of Tests for CSE Software</u>. · Integration tests: Focus on testing the inter the full system level. · System-level tests: Focus on testing the full Steps: 1. Set up automated builds of the code with high warning levels and eliminate all level. For example, a system-level test of a C input files, running the full simulation code, an warmogi. Select fare harness frameworks Belict a system-level test harness for system-executable tests that report mesults appropriately (or CTest/CDark, Jenkins). Select a unit test harness to effectively define and nn fine-grained retegration and unit tests (e.g. Google Test, PDI-IN). solutions Managing and reporting on testing: The simple runs one or more executables, saving the output into c. Customize or streamline system-level and/or unit test frameworks for use in examine. Once a package becomes too complex, th your particular project. satisfactory and requires various enhancements. Au Add system-level tests to protect major user functionality. a. Select inputs for several important problem classes and run code to produce called test harnesses) are introduced to lower the ( outputs. adding new tests. For example, filters can be autor outputs outputs so change or verification tests with a system-level test harness in order so now important behavior. 4. Add integration and unit tests (as needed for addrug/thangrg code) a. incorporate tests [1, 2] for code to be changed identify change points for thread change or new code. Find test points when code behavior and be sneed. indicates problems (e.g., here and here), and to disp color) which build instantiations generated errors (e requires developers to check a website on a regula Regression -- a return to a former or less de Break dependencies in order to get the targeted code into the unit test return to a less developed state. · Cover targeted code to be changed with sufficient (characterization) This material is based upon work supported by the U.S. Dep Computing Research and Biological and Envi is material is based upon work supported by the U.S. Department of Energy earch and Biological and En DRAFT Version 0.1, April 27, 2015

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:

29

- 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-
- Effective in R&D setting. Avoids a deadlinebased approach. Deadlines are dealt with in a different way.
- Provides a board for viewing and managing issues



moment Tuesday

### IDEAS Confluence, Jira Agile, Kanban Pages / IDEAS Common Home Developer Guide, on Managing IDEAS Project Activities Using JIRA Agile and Kanban Confluence site Created by Michael Heroux, last modified by Jim Willenbring about an hour ago Contents: Overview Standard JIRA Agile Issue Type Definitions IDEAS Issue Management Structure IDEAS Projects Kanban board Board -☆ Issue Status Viewing Issues QUICK FILTERS: No subtasks No ToDo Epics No epics HowTo Outreach UseCase xSDK Only My Issues **FY15** ... Show more Entering an IDEAS Project Activity Managing Existing IDEAS Issues 6 of 24 In Progress 24 of 64 To Do 5 of 13 Selected 6 of 32 Done Release... HOW-10 HOW-22 L HOW-7 L HOW-1 HowTo-2.1: Assess and Develop plan an agile Based on assessment. Hans adds IDEAS leads to measure SE maturity within software ecosystem lifecycle recommend use case testing JIRA and Confluence model (ASELM) for practices and infrastructure **IDEAS** teams HowTo-2.2: Best practice. HowTo-2.1: Establish sci HOW-14 L HOW-2 Kanban Board, on HowTo-2.1: Measure SW Write up summary notes from HOW-36 **HOW-28** productivity improvement and Methodology telecon Create a Performance Create Confluence page for benefits Jira site. Portability What-Is Document guiding IDEAS team member on how to use Jira HowTo-2.1: Develop Wha. HOW-12 HOW-9 Four columns: HowTo-2.1: Adapt and pr... HowTo-2.1: Establish Hans vells at Procurement scientific software ecosystem about Pcard . **HOW-37** lifecycle model **HOW-16** Το Do Coordinate all interested Single repository ait workflow IDEAS team members to 12 HOW-11 best practices. L discus Jenkins setup/use **HOW-18** HowTo-2.1: Develop What Is Conduct a GQM style survey Selected HowTo-2.1: Establish sci.. HowTo-2.1: Provide traini... and How To Content for refactoring opportunities and Improving CSE SW Practices needs of UseCases In Progress HOW-24 **HOW-38** HowTo-2.1: Establish sci Develop strategy for Assess status of use case HOW-40 De testing HowTo-2.1: Adapt and addressing contributor Done HOW-19 agreements for open-source provide tools & processes HowTo-2.1: Assess and ... that will enhance IDEAS & Evaluate the readiness of . . . . . . . . UseCases in Identifying HOW-42 change points, Finding test **HOW-26** Develop training material on HOW-41 LTT Develop a list of future How how to use Jira Agile and HowTo-2.1: Establish sci HowTo-2.1: Provide training to topics to post on ideas-Kanban opportunities for IDEAS Team productivity.org members HowTo-2.1: Provide traini... HOW-35 HowTo-2.1: Develop Wha... Document How the IDEAS HOW-4 Project will Use Jira for Issue

### 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 \_\_\_\_\_?



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

