Productive & Sustainable: More Effective CSE

better scientific software

IDEAS productivity

Sandia National Laboratories

Exceptional service in the national interest

Mike Heroux
Senior Scientist
Center for Computing Research
Sandia National Laboratories

Scientist in Residence
St. John's University, MN

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

16th SIAM Conference on Parallel Processing for Scientific Computing (PP18)

March 7-10, 2018, Tokyo, Japan

- Venue
  - Nishi-Waseda Campus, Waseda University
- Organizing Committee Co-Chairs's
  - Satoshi Matsuoka (Tokyo Institute of Technology, Japan)
  - Kengo Nakajima (The University of Tokyo, Japan)
  - Olaf Schenk (Universita della Svizzera italiana, Switzerland)
- Contact
  - Kengo Nakajima, nakajima(at)cc.u-tokyo.ac.jp
  - http://nkl.cc.u-tokyo.ac.jp/SIAMPP18/
  - http://www.siam.org/meetings/pp18/

TEAMS, COLLABORATORS, INSPIRATIONS

DOE ASCR IDEAS Productivity Project

Sandia National Laboratories

DOE Program Managers

ASCR: Thomas Ndousse-Fetter
BER: Paul Bayer, David Lesmes

IDEAS: Interoperable Design of Extreme-scale Application Software

ASCR Co-Leads: Mike Heroux (SNL) and Lois Curfman McInnes (ANL)
BER Lead: J. David Moulton (LANL)

Executive Advisory Board

John Cary (Tech-X)
Mike Glass (SNL)
Susan Hubbard (LBNL)
Doug Kothe (ORNL)
Sandy Landsberg (DOD)

BER Use Cases

Lead: J. David Moulton (LANL)
Carl Steefel (LBNL) *1
Scott Painter (ORNL) *2
Reed Maxwell (CSM) *3
Glenn Hammond (SNL)
Tim Scheibe (PNNL)
Laura Condon (CSM)
Ethan Coon (LANL)
Dipankar Dwivedi (LBNL)
Jeff Johnson (LBNL)
Eugene Kikinzon (LANL)
Sergi Molins (LBNL)
Steve Smith (LLNL)
Erica Woodburn (LBNL)
Carol Woodward (LLNL)
Xiaofan Yang (PNNL)

Methodologies for Software Productivity

Lead: Mike Heroux (SNL)
Roscoe Bartlett (SNL)
Mark Berrill (ORNL)
Anshu Dubey (ANL)
Todd Gamblin* (LLNL)
Osni Marques (LBNL)
Pat McCormick* (LANL)
Nicholas Moss (LANL)
Andrew Salinger* (SNL)
Jason Sarich (ANL)
Dali Wang (ORNL)

Extreme-scale Scientific Software Development Kit (xSDK)

Lead: Lois Curfman McInnes (ANL)
Alicia Klinvex (SNL)
Sherry Li (LBNL)
Vijay Mahadevan (ANL)
Daniel Osei-Kuffuor (LLNL)
Barry Smith (ANL)
Mathew Thomas (PNNL)
Jim Willenbring (SNL)
Ulrike Yang (LLNL)

Outreach and Community

Lead: David Bernholdt (ORNL)
Lisa Childers* (ALCF)
Rebecca Hartman-Baker* (NERSC)
Judy Hill* (OLCF)
Bill Spotz* (SNL)

IDEAS productivity

* Liaison
*1 *2 *3: Leads: Use Cases 1, 2, 3

BER Terrestrial Programs		**DOE Extreme-scale Programs**			**DOE Computing Facilities**
SFAs	**NGEE**	**Exascale Co-Design**	**ASCR Math & CS**		**ALCF**
CLM	**ACME**	**Exascale Roadmap**	**SciDAC**	**NERSC**	**OLCF**

IDEAS-ECP Productivity Project

Michael Heroux (SNL), Co-Lead PI
Lois Curfman McInnes (ANL), Co-Lead PI
David Bernholdt (ORNL), Co-PI, Outreach Lead
Todd Gamblin (LLNL), Co-PI
Osni Marques (LBNL), Co-PI
David Moulton (LANL), Co-PI
Boyana Norris (Univ of Oregon), Co-PI
Satish Balay (ANL)
Roscoe Bartlett (SNL)
Anshu Dubey (ANL)
Rinku Gupta (ANL)
Christoph Junghans (LANL)
Alicia Klinvex (SNL)

Reed Milewicz (SNL)
Mark Miller (LLNL)
Elaine Raybourn (SNL)
Barry Smith (ANL)
Louis Vernon (LANL)
Greg Watson (ORNL)
James Willenbring (SNL)

Facilities Liaisons:
Lisa Childers (ALCF)
Rebecca Hartman-Baker (NERSC)
Judy Hill (OLCF)
Hai Ah Nam (LANL)
Jean Shuler (LLNL)

+ xSDK-ECP Project

Inspirations and Collaborations

Sandia National Laboratories

Software Sustainability Institute

software carpentry

WSSSPE

WORKING TOWARDS SUSTAINABLE SOFTWARE FOR SCIENCE: PRACTICE AND EXPERIENCES

Trilinos

Outline

- Concepts & Definitions.
- Why I care about productivity & sustainability.
- Incentives: Better productivity & sustainability.
- Completeness philosophy.
- Some Practical Strategies.
- Productivity++ and BetterScientificSoftware

My Roles

- Regarding observations on opportunities to improve:
  - *More like a psychologist than expert.*
- Regarding software tools, processes, practices improvements:
  - *More like a carpenter than expert.*

CONCEPTS & DEFINITIONS

Productivity & Sustainability

- Productivity – Output per unit input.
  - Specifically: *Developer Productivity.*

- Sustainability – The future cost of usability.
  - Specifically: *Software Sustainability.*

- Strategy: Focus on *improvement.*
  - Optometrist approach.

Better, Faster, Cheaper: Pick two of the three.

Scenario: *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.

PRODUCTIVITY & SUSTAINABILITY: WHY NOW?

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

Sandia National Laboratories

Reproducibility

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

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

Future High Performance Computing:

- IS NOT primarily about HW.
- IS primarily about Algorithms.
- IS primarily about software:
  - ΔA, ΔS >> ΔH
  - Algs: Adequate R&D focus.
  - SW: Opportunities for improvement.

“Easy” Work in Progress

- New parallel algorithms:
  - Turning out to be feasible.
  - Clever ideas: David Keyes talk from Monday.
  - Lots to do, but steady progress
  - Much evidence in this conference.
- Current Thread Programming Environments:
  - C++, OpenMP, others: Working.
  - Runtimes: Still a lot of work, but progress.
- Lots to do, but community is focused.

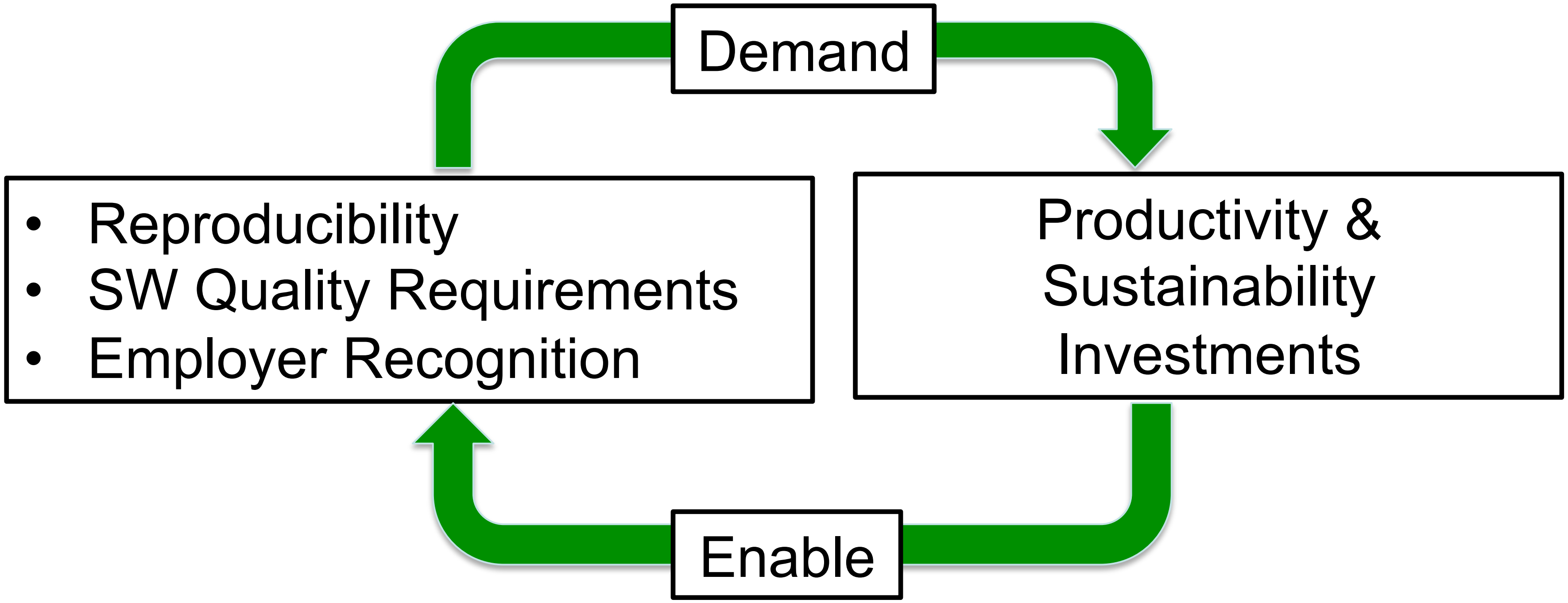
“Hard” Work

- Billions SLOC of encoded science & engineering.

- Challenge:
  - Transfer, refactor, rewrite for modern systems.
  - Do so with modest investment bump up.
  - Work across distinct and diverse teams.
  - Deliver science at the same time.
  - Make the next disruption easier to address.

TOWARD REPRODUCIBLE SCIENCE: CREATING INCENTIVES

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.

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/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.

ACM TOMS
Replicated Computational Results (RCR)

- Standard reviewer assignment:
  - Nothing changes.
- RCR reviewer assignment:
  - Concurrent with standard reviews.
- RCR process:
  - Multi-faceted approach, Bottom line: Trust the reviewer.
- Publication:
  - Replicated Computational Results Designation.
  - Review report appears with published manuscript.

RCR Process: Two Basic Approaches

1. Independent replication:
   - Open review.
2. Review of computational results artifacts:
   - ”Boutique” computing system.
   - In this situation:
     - Careful documentation of the process.
     - Software should have its own substantial V&V process.

Big Picture of TOMS RCR

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).***

- Improve science.
- Not good now:
  - Trust comes from "cloud" of papers with similar results.
  - Which could still be wrong.
  - 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.

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

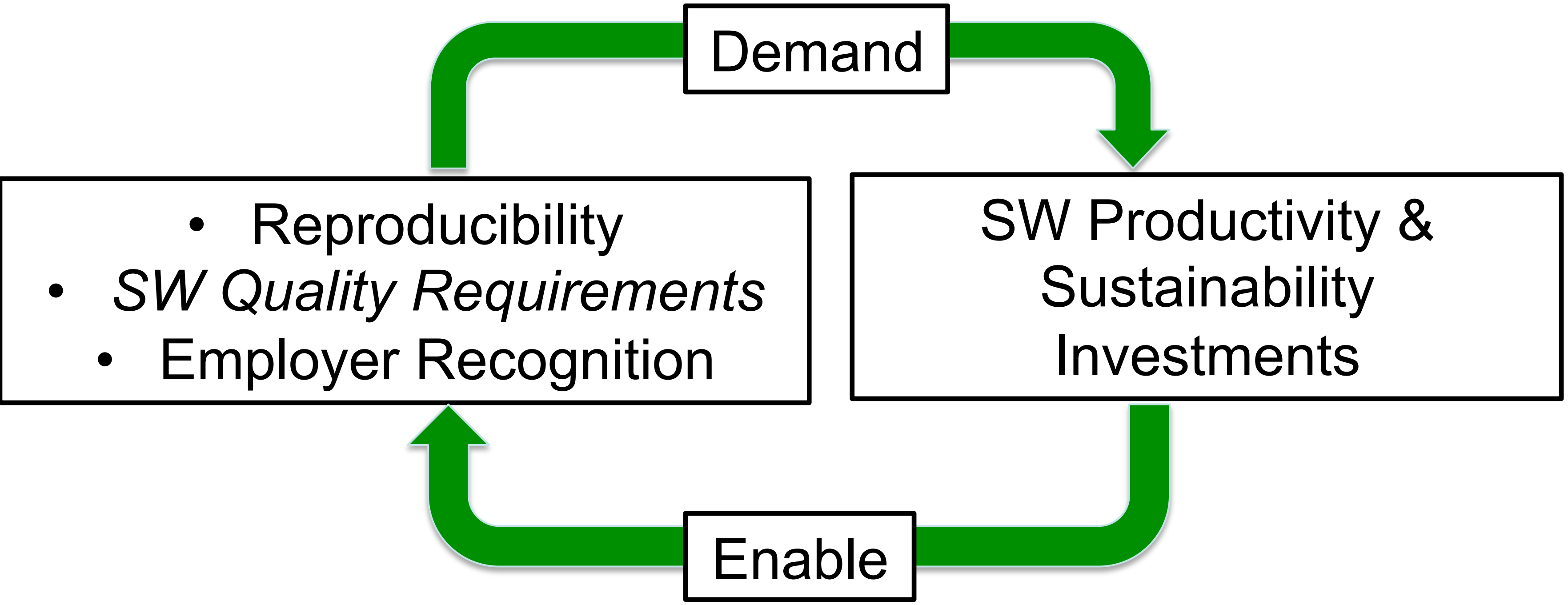
SC17 Reproducibility Initiative

- Two appendices:
  - Artifact description (AD).
    - Schematic of your computing environment.
    - Makes it easier to rerun computations in future.
    - **AD required for best paper & student paper.**
  - Computational Results Analysis (CRA).
    - Targets ”boutique” environments.
    - Improves trustworthiness.
- Details:
  http://sc17.supercomputing.org/submitters/technical-papers/reproducibility-initiatives-for-technical-papers/

Sources for CRA 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.

Incentives & Productivity/Sustainability


Demand
• Reproducibility
• *SW Quality Requirements*
• Employer Recognition
SW Productivity & Sustainability Investments
Enable

New funding proposal element:
SW Productivity and Sustainability Plan

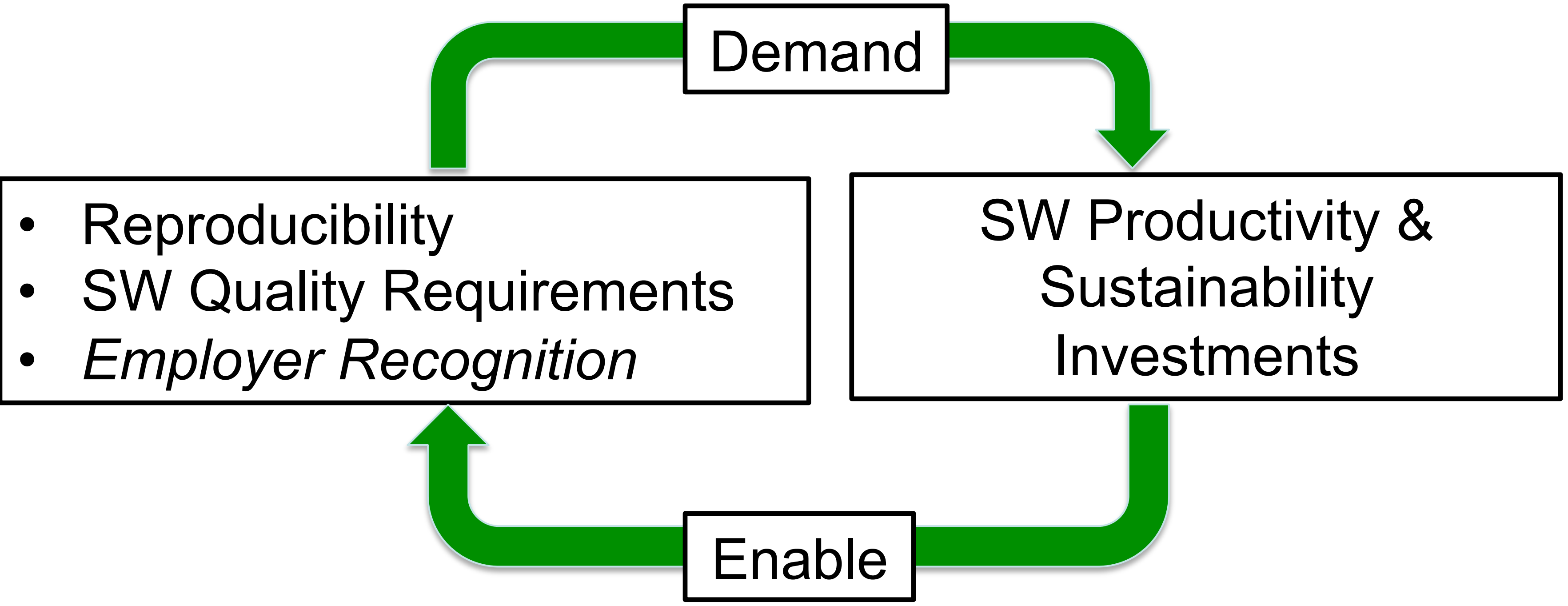
DOE SW Productivity and Sustainability Plan (SW PSP).

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

DOE BER SW PSP Requirements

- 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.

Employers & Productivity/Sustainability



- Reproducibility
- SW Quality Requirements
- *Employer Recognition*

SW Productivity & Sustainability Investments

Next focus:

- Work with DOE labs to recognize SW as base research.
- Pressing for Scientific SW Productivity R&D base funding.

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 CSE software will matter a lot more.

PRODUCTIVITY IMPROVEMENT

CSE & Formal Software Methodologies: Troubled History

- Cray (1990):
  - Formal Waterfall Method.
- DOE ASCI (2000):
  - CMMI
- Failed to follow own process: Elicit requirements.

CSE Complete: Useful "Overhead"

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

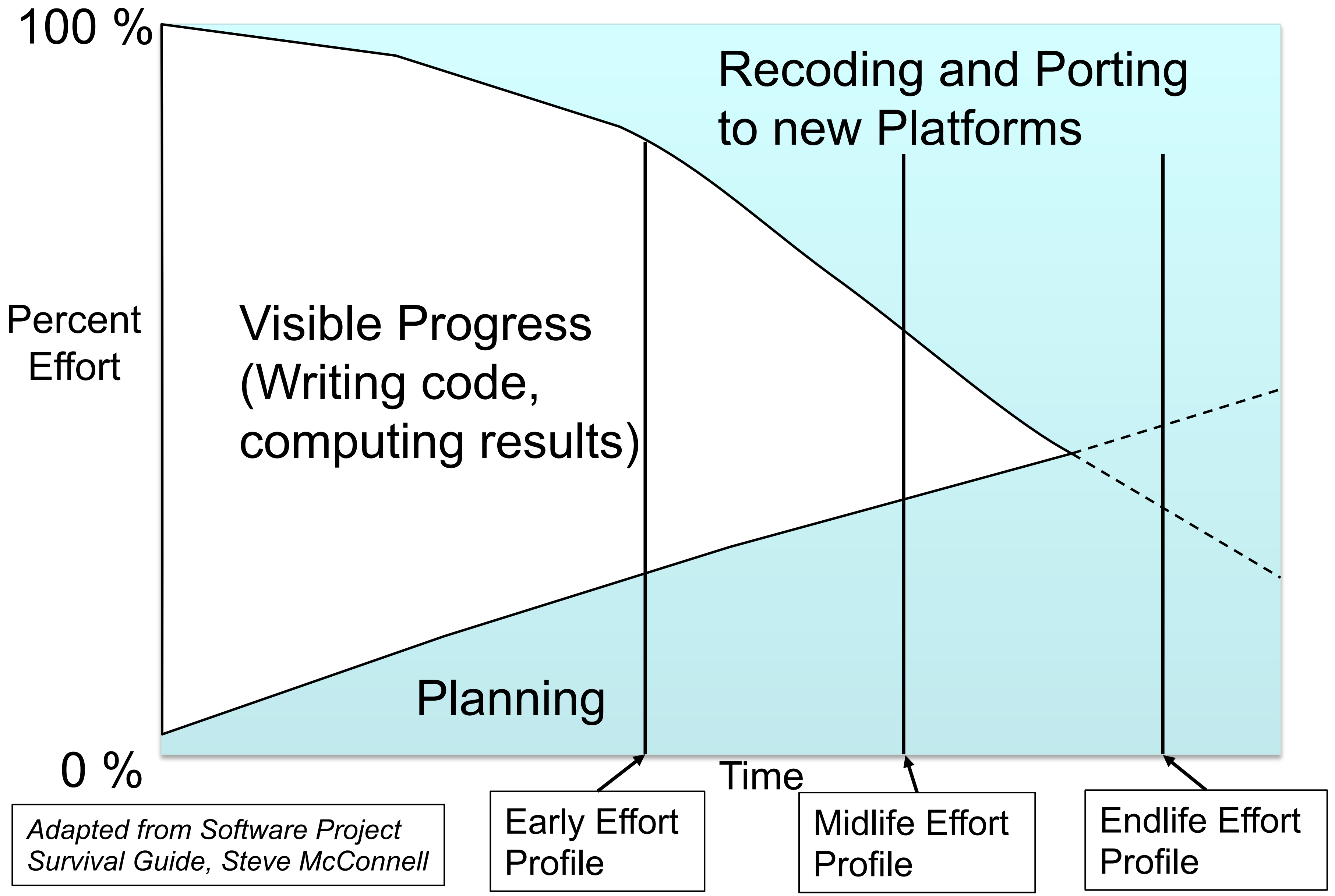
"Give me six hours to chop down a tree and I will spend the first four sharpening the axe."
Abraham Lincoln

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

- CSE Complete: Ultimate value is CSE.
  - What activities improve CSE?

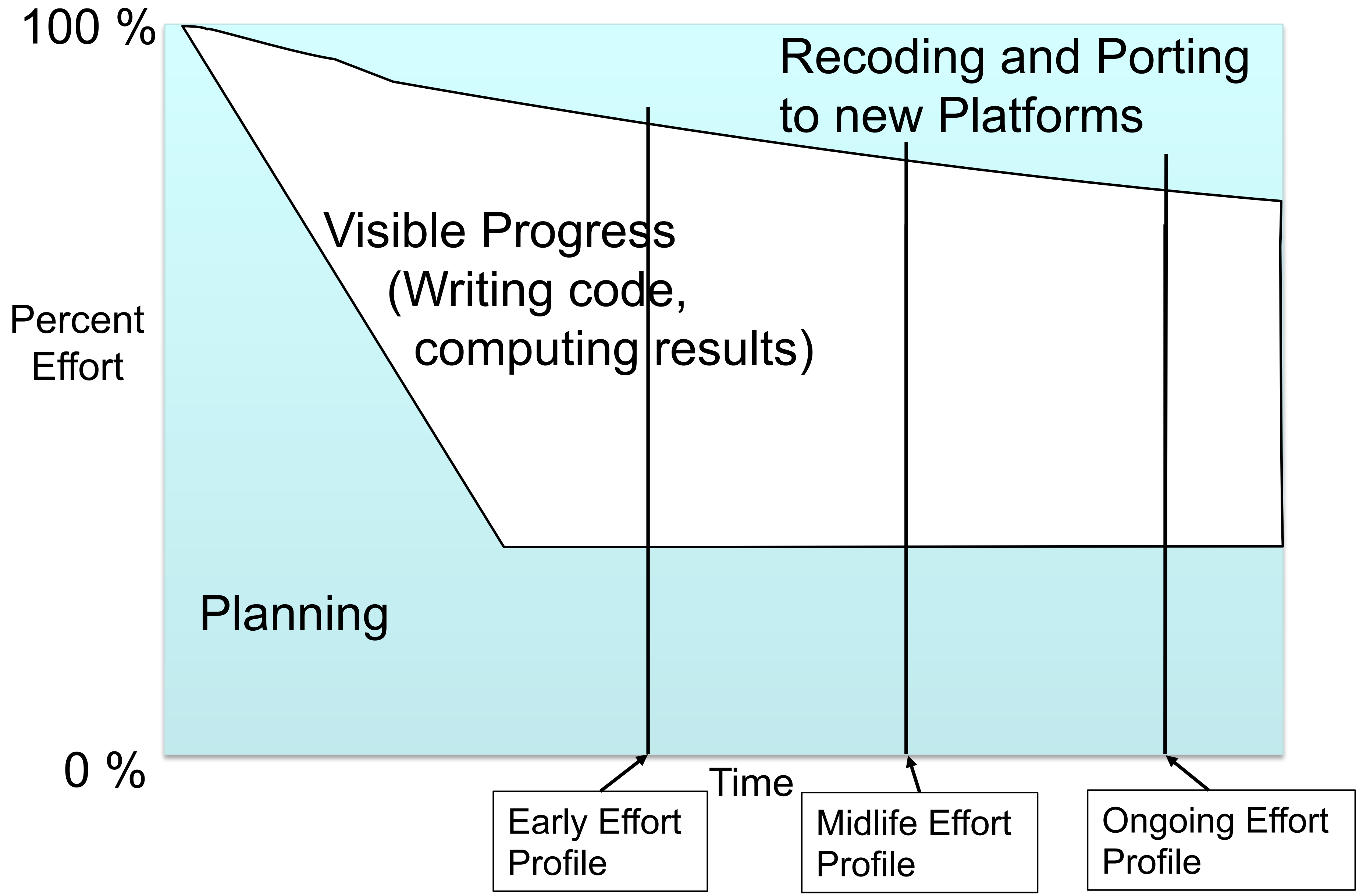
PRODUCTIVITY IMPROVEMENT: PLANNING

Code-and-Fix Development Approach



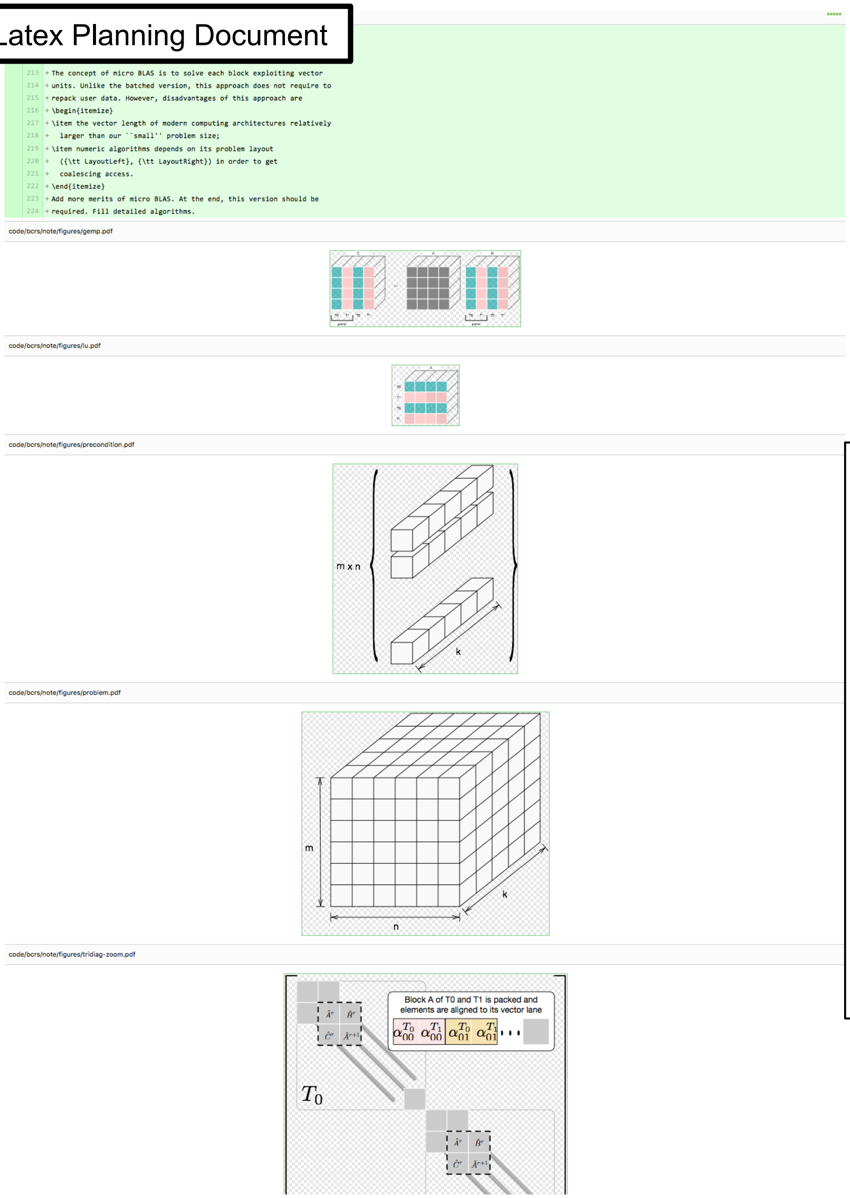
Adapted from Software Project Survival Guide, Steve McConnell

Simple Planned Development Approach



Planning tools: Use what you know

Latex Planning Document



```
+ The concept of micro BLAS is to solve each block exploiting vector
+ units. Unlike the batched version, this approach does not require to
+ repack user data. However, disadvantages of this approach are
+ \begin{itemize}
+ \item the vector length of modern computing architectures relatively
+   larger than our ``small'' problem size;
+ \item numeric algorithms depends on its problem layout
+   ({\tt LayoutLeft}, {\tt LayoutRight}) in order to get
+   coalescing access.
+ \end{itemize}
+ Add more merits of micro BLAS. At the end, this version should be
+ required. Fill detailed algorithms.
```

code/bcrs/note/figures/gemp.pdf

code/bcrs/note/figures/lu.pdf

code/bcrs/note/figures/precondition.pdf

code/bcrs/note/figures/problem.pdf

code/bcrs/note/figures/tridiag-zoom.pdf

m x n
k
m
n
k
Block A of T0 and T1 is packed and elements are aligned to its vector lane
T_0

Commit log messages

KokkosKernels - add design note for discussion.
This design note is very informal and working note for discussion.
…

For typeset, "make"

KokkosKernels - add more algorithm variants.
In this algorithm design, I have a few assumptions.
…

KokkosKernels:
Micro & Batched BLAS Design Document

- 6 weeks: Design by LaTeX.
  - Review by diverse experts.
  - Significant design changes: In text only.
- 2 weeks: Write code.

Message: *Use the tools you know.*

Courtesy: KokkosKernels Development Team

PRODUCTIVITY IMPROVEMENT: ISSUE TRACKING

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)

Informal, less training

Formal, more training

Kanban principles

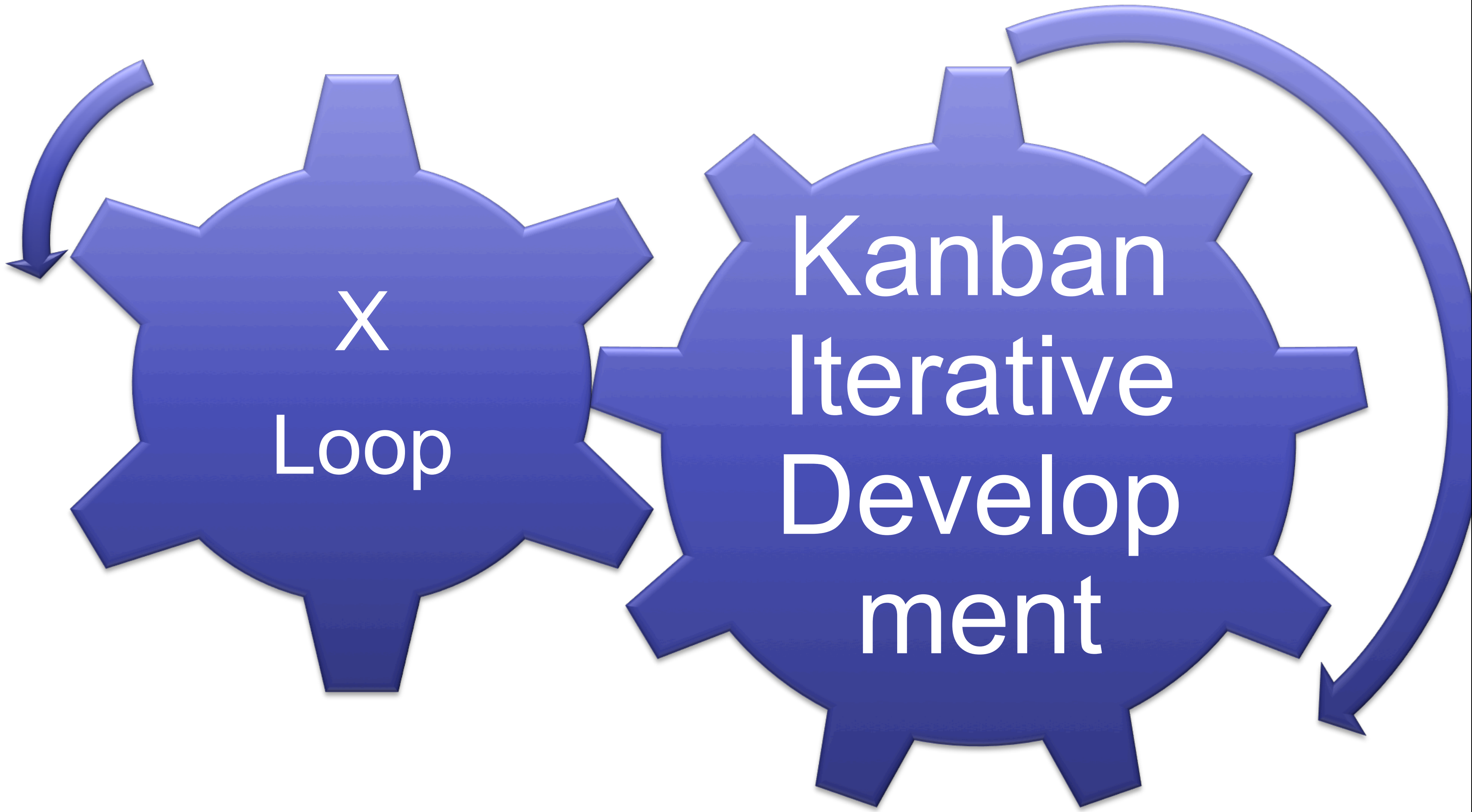
Task: Have Eureka moment by Tuesday.

- 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

Basic Kanban

Backlog	Ready	In Progress	Done
• Any task idea	• Task + how to do it. • Could be pulled when slot opens.	• Task you are working on *right now.* • **The only Kanban rule: Can have only so many "In Progress" tasks.** • Limit is based on experience, calibration.	• Completed tasks. • Record of your life activities. • Rate of completion is your "velocity".

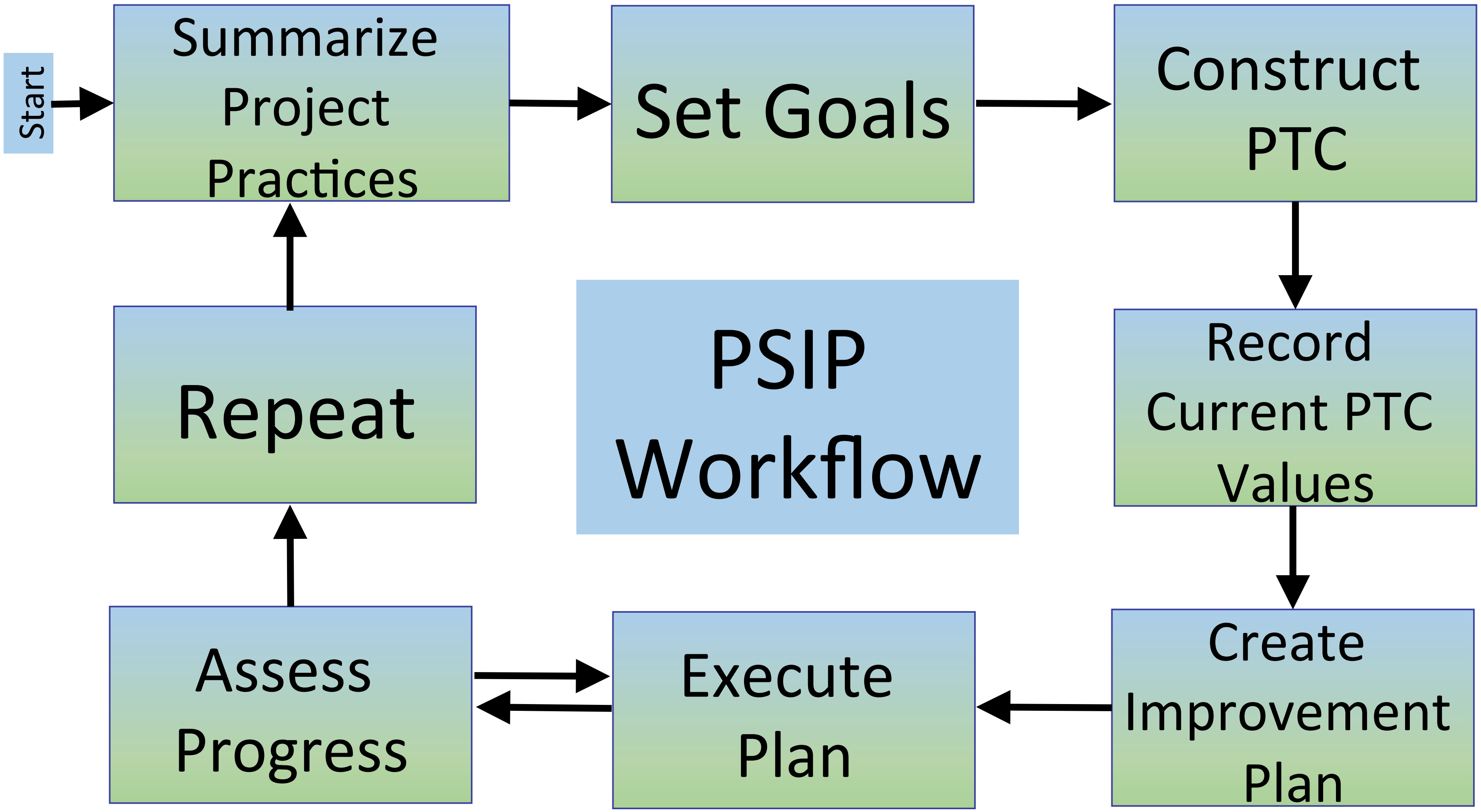
KanbanX



X = Sustainability, Performance

TOWARD A REPEATABLE PROCESS

SW Productivity & Sustainability Improvement Plans



Progress Tracking Card Example

Practice: Test Coverage	Score (0 – 5):	
Score Descriptions		
0	Little or no independent testing. Functional testing via users.	
1	Independent functional testing of primary capabilities.	
2	Primary functional testing, some unit test coverage.	
3	Comprehensive unit testing, primary functional testing.	
4	Comprehensive unit testing, functional testing for documented use cases.	
5	Comprehensive unit, use case functional testing; test coverage commitment.	

better scientific software

New community portal: Scientific Software Improvement Exchange
`https://betterscientificsoftware.io`

Sandia National Laboratories

May 2017

BSS Software Platform

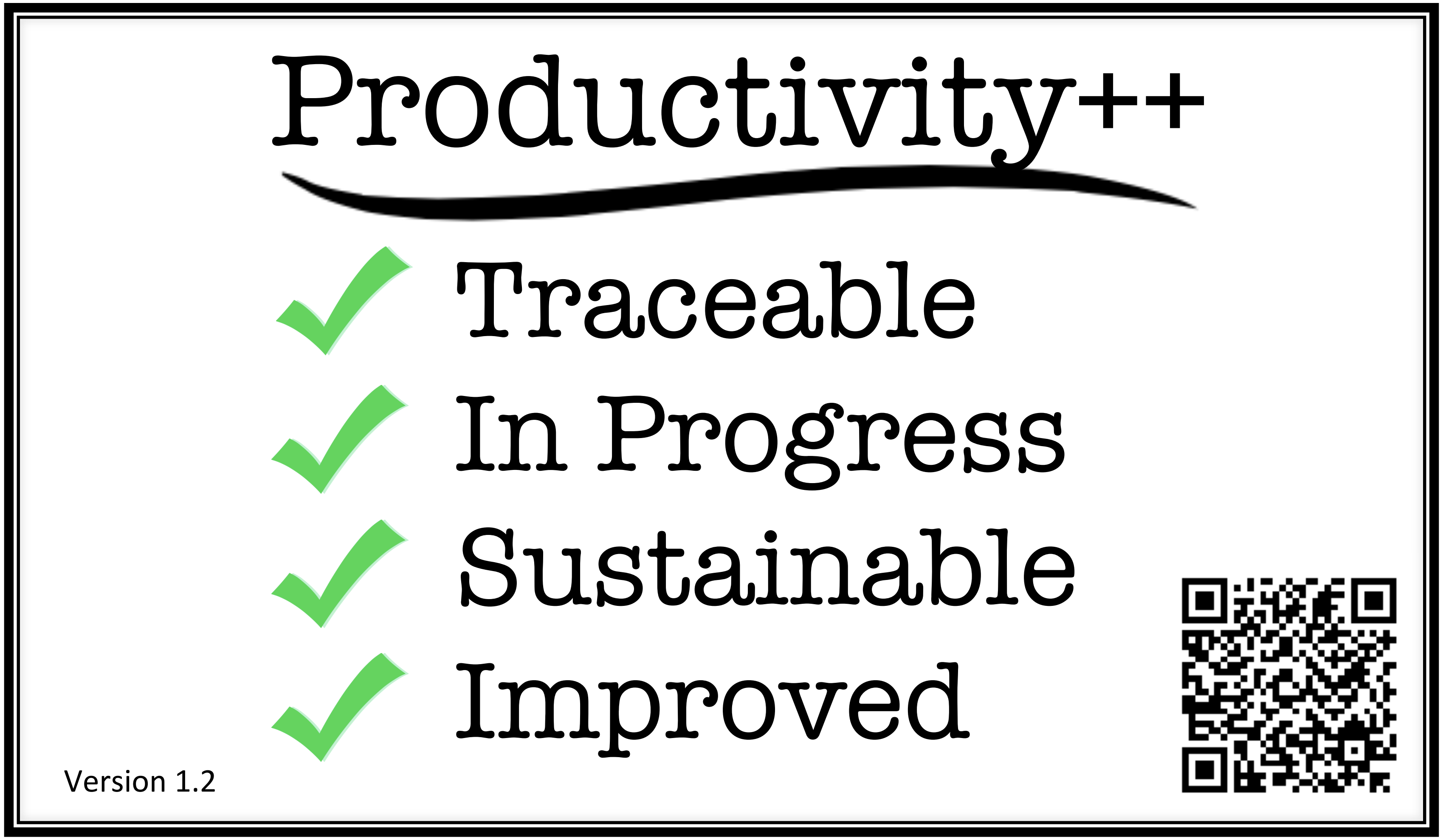
Component Technology	Backend		Frontend
	Google Docs	**GitHub**	**Ruby on Rails**
Location	Google Drive	betterscientificsoftware organization	betterscientificsoftware.info
Purpose	• Rapid collaborative content development. • Multi-user typing, suggested edits, comments.	• Content creation, refinement, management (from Google Drive). • Content packaging for use with BSS.info	• User-facing portal • Polished backend content • Blogs, forums • Mailing lists.
Contributors	Community content experts	Community content experts. BSS staff.	BSS staff. Web dev experts.
Consumers	BSS GitHub Backend	BSS Frontend	CSE community
Content Notes	Content migrates to GitHub after it stabilizes.	Content is managed in git repos, markdown.	Content from backend, community.

Summary

- CSE 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?
- Productivity effort must enhance CSE results:
  - High quality, low value software cannot be the outcome.
  - Quality improvements must be measured.
- Changes in publishers, funders, employers expectations:
  - Could view as bothersome: More “overhead” impeding progress.
  - Could be opportunity: Be proactive, part of the initiative.

Productivity++ Initiative

Ask: *Is My Work _________ ?*



Productivity++

✓ Traceable

✓ In Progress

✓ Sustainable

✓ Improved

Version 1.2

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

Contribute!

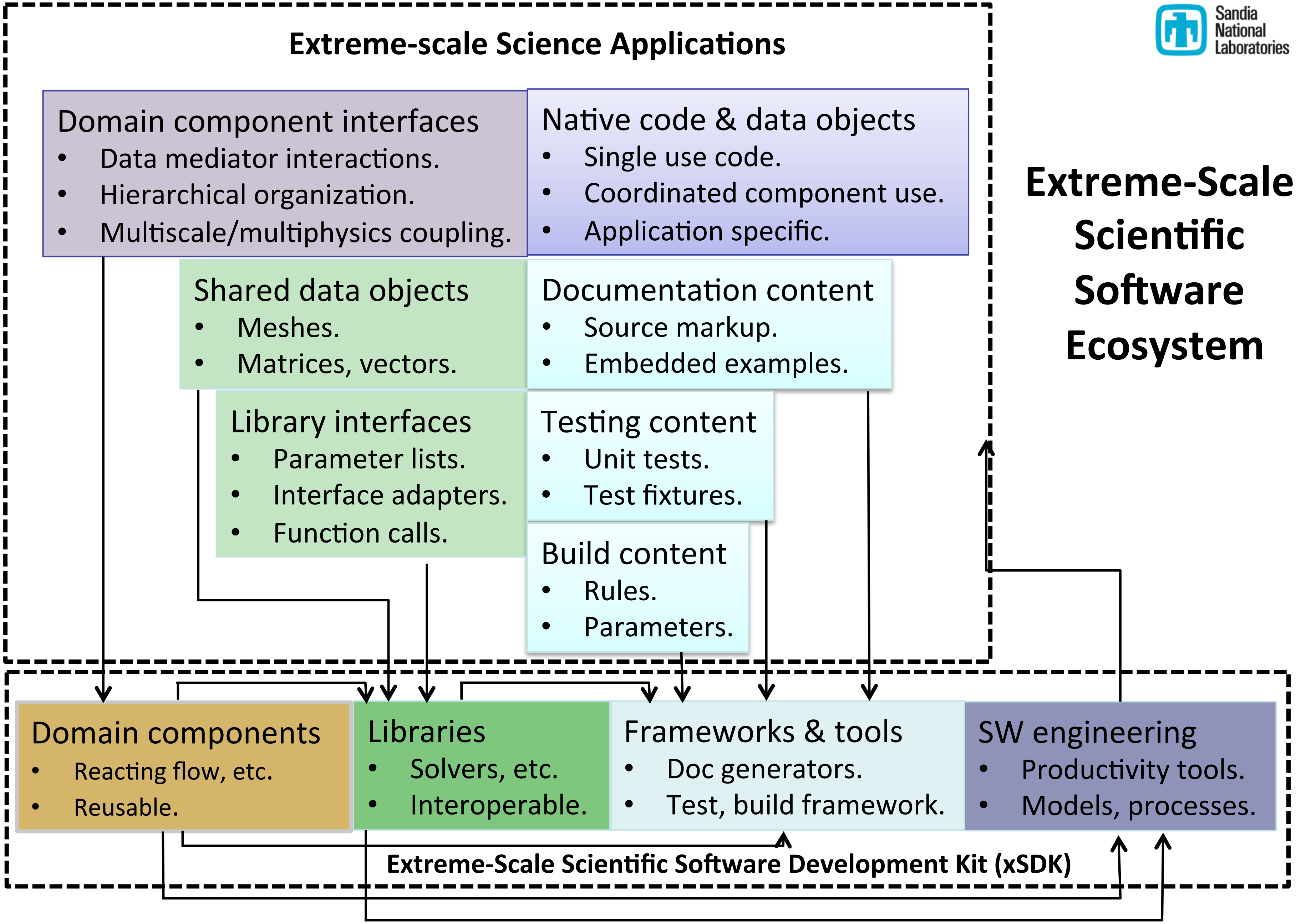
- https://github.com/betterscientificsoftware/betterscientificsoftware.github.io/blob/master/README.md
- Or search “github betterscientificsoftware”.

Acknowledgments

- Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2016-8466 C.

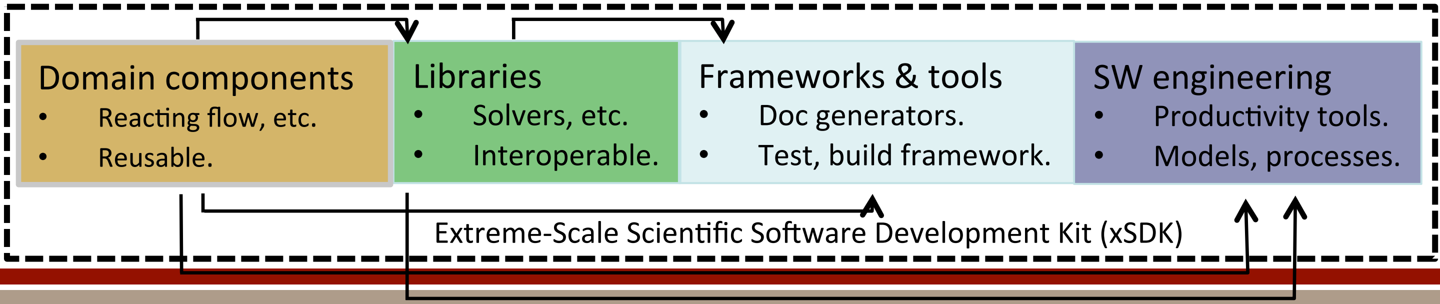
PRODUCTIVITY IMPROVEMENT: APPLICATION COMPOSITION AND SOFTWARE ECO-SYSTEMS

Extreme-Scale Scientific Software Ecosystem



xSDK focus

- Common configure and link capabilities
  - xSDK users need full and consistent access to all xSDK capabilities
  - Namespace and version conflicts make simultaneous build/link of xSDK difficult
  - Determining an approach that can be adopted by any library or components development team for standardized configure/link processes
- Library interoperability
- Designing for performance portability



Standard xSDK package installation interface

Motivation: Obtaining, configuring, and installing multiple independent software packages is tedious and error prone.

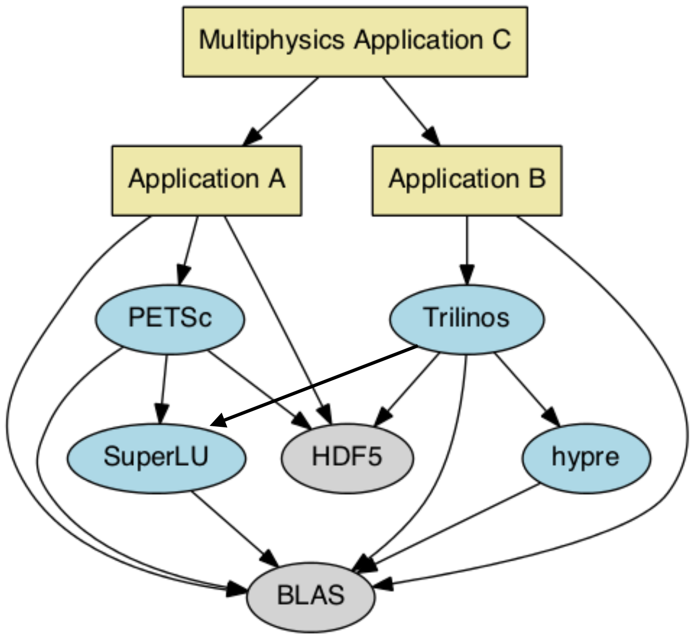
- Need consistency of compiler (+version, options), 3rd-party packages, etc.

Approach: Define a *standard xSDK package installation interface* to which all xSDK packages will subscribe and be tested

Accomplishments:

- Work on implementations of the standard by the hypre, PETSc, SuperLU, and Trilinos developers
- PETSc can now use the "scriptable" feature of the installers to simultaneously install hypre, PETSc, SuperLU, Trilinos with consistent compilers and 'helper' libraries.

xSDK Build Example



Impact: Foundational step toward seamless combined use of xSDK libraries, as needed by BER use cases and other multiphysics apps

xSDK Minimum Compliance Requirements:

- M1. Each xSDK compliant package must support the the standard xSDK cmake/configure options.
- M2. Each xSDK package must provide a comprehensive test suite that can be run by users and does not require the purchase of commercial software
- M3. Each xSDK compliant package that utilizes MPI must restrict its MPI operations to MPI communicators that are provided to it and not use directly MPI_COMM_WORLD.
- M4. Each package team must do a 'best effort' at portability to key architectures, including standard Linux distributions, GNU, Clang, vendor compilers, and target machines at ALCF, NERSC, OLCF. Apple Mac OS and Microsoft Windows support are recommended.
- M5. Each package team must provide a documented, reliable way to contact the development team; this may be by email or a website. The package teams should not require users to join a generic mailing list (and hence receive irrelevant email they must wade through) in order to report bugs or request assistance.
- M6 – 11…

https://ideas-productivity.org/resources/xsdk-docs

xSDK Recommended Compliance Requirements:

- R1. It is recommended that each package have a public repository, for example at github or bitbucket, where the development version of the package is available. Support for taking pull requests is also recommended.
- R2. It is recommend that all libraries be tested with valgrind for memory corruption issues while the test suite is run.
- R3. It is recommended that each package adopt and document a consistent system for propagating/returning error conditions/exceptions and provide an API for changing the behavior.
- R4. It is recommended that each package free all system resources it has acquired as soon as they are no longer needed.

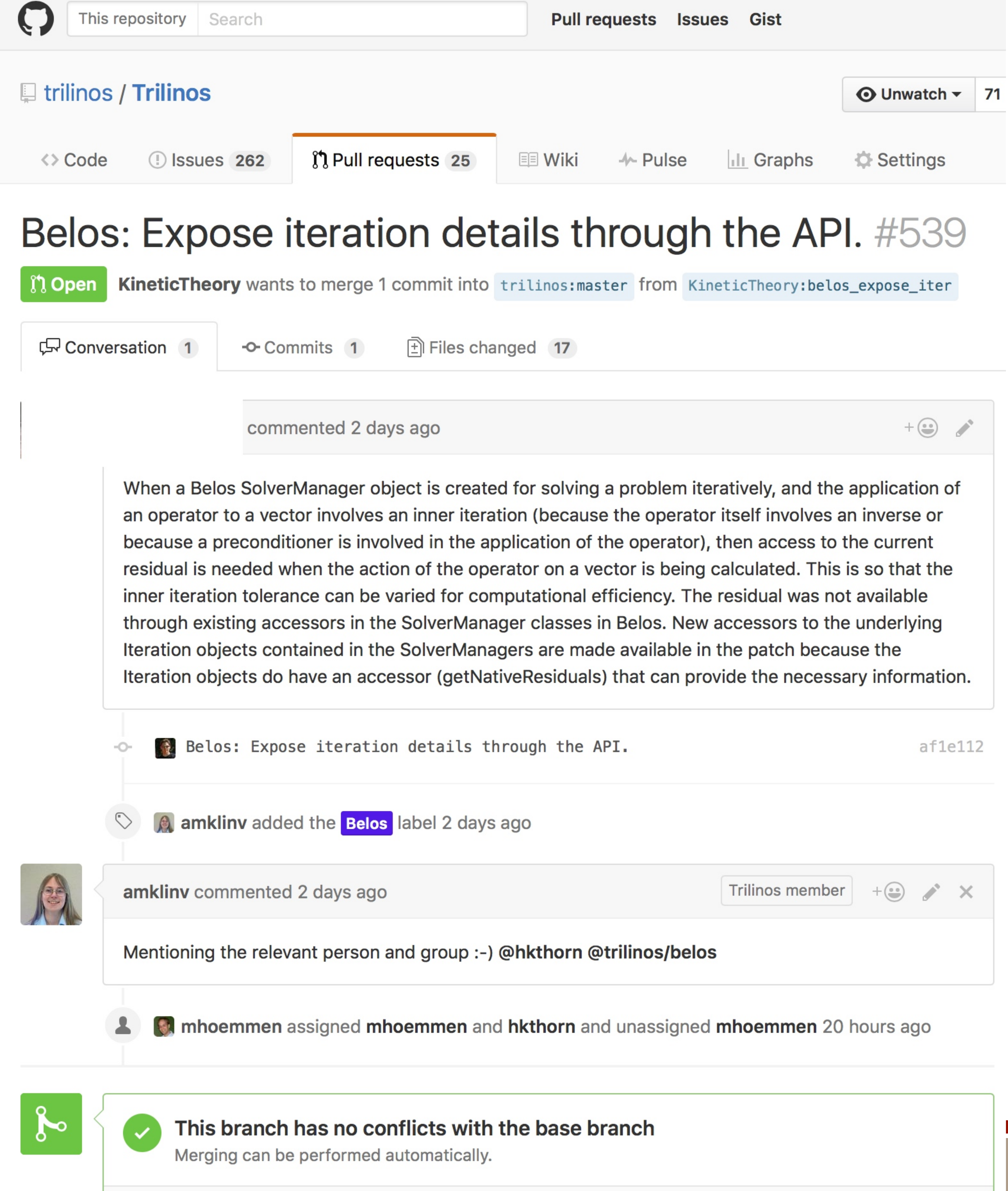
Message to This Audience

Consider what software ecosystem(s) you want your software to be part of and use.

SUSTAINABILITY IMPROVEMENT

"Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live."

- John Wood


This repository | Search | Pull requests | Issues | Gist

trilinos / Trilinos

Unwatch 71

Code | Issues 262 | Pull requests 25 | Wiki | Pulse | Graphs | Settings

Belos: Expose iteration details through the API. #539

Open KineticTheory wants to merge 1 commit into `trilinos:master` from `KineticTheory:belos_expose_iter`

Conversation 1 | Commits 1 | Files changed 17

commented 2 days ago

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.` af1e112

amklinv added the Belos label 2 days ago

amklinv commented 2 days ago — Trilinos member

Mentioning the relevant person and group :-) @hkthorn @trilinos/belos

mhoemmen assigned mhoemmen and hkthorn and unassigned mhoemmen 20 hours ago

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

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.

Message to This Audience

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