OAK RIDGE National Laboratory

March 19, 2025 | Engelberg, Switzerland

# Sustainable HPC Software: Lessons from the Trenches (A Maintainer's Perspective)


Damien Lebrun-Grandié

Computational Sciences and Engineering Division



U.S. DEPARTMENT OF ENERGY

ORNL IS MANAGED BY UT-BATTELLE LLC
FOR THE US DEPARTMENT OF ENERGY

# PESO

## Partnering for Scientific Software Ecosystem Stewardship Opportunities

https://pesoproject.org

The PESO Project exists to preserve, sustain, and advance the investments made by the Exascale Computing Project in a robust, versatile, and portable HPC software ecosystem and the people who make the ecosystem effective. Partnership with CASS.

**PIs: M. Heroux and L.C. McInnes**

### PESO PARTNERSHIPS

| STAKEHOLDER ENGAGEMENT AND CONSORTIUM PARTNERSHIPS | | | COMMUNITY DEVELOPMENT |
|---|---|---|---|
| Consortium | Applications Community | Computing Facilities | Better Scientific Software (BSSw) Fellowship Program<br><br>Workforce |
| Commercial HPC Companies | US Agencies | Industrial Users | |

**KEY PESO GOALS**

- Enable applications to realize benefits of a software ecosystem
- Emphasize software product quality, continued fostering of software product communities, and delivery of products, working with CASS

### PESO SERVICES

**INTEGRATION PARTNERSHIPS**

Provide resources and support for portfolio build, integration, testing
- Spack integration
- CI testing
- Portfolio support & management

**Collaboration with SSOs**

**SQA & SECURITY**

Provide infrastructure to support and leverage product team SQA
- Supply chain, product quality
- Testing, documentation

### PESO PRODUCTS

**E4S AND SPACK**

- Support for product integration
- Features for consortium products
- Documentation, training

**PORT & TEST PLATFORMS**

- Frank test & devpt system
- Cloud resources
- Documentation, training

**BSSw.io CONTENT**

- Articles on scientific software productivity and sustainability

### Scientific software ecosystem benefits (technical and community)

**100,000+**
Lines of code replaced with high-quality libraries and tools

**10,000+**
Community members via ecosystem collaborations

**1,000+**
Code teams share ecosystem costs and benefits

**100+**
Speedup using advanced devices like GPUs

**10+**
Reduction in build times via Spack build caches

**1**
Source code base for all computing systems

**Stakeholders:**
Applications Community
Commercial HPC Companies
Industrial Users
US Agencies

**Computing Facilities:**
ALCF, NERSC, OLCF

**CRLC:**
ANL, BNL, LBNL, LLNL, LANL, ORNL, PNNL, SNL

**Advisory Board:**
Reps from ANL, LBNL, LLNL, LANL, ORNL, SNL

**CASS Consortium**
PESO, COLABS, CORSA, FASTMath, RAPIDS, STEP, SWAS, S4PST

**DOE Program Managers**
**ASCR:** H. Finkel, B. Brown, B. Spotz, S. Hier-Majumder, R. Pino, D. Rabson
**NNSA:** S. Hammond

PESO

# PESO Organizational Chart

**PIs: Mike Heroux (ParaTools) and Lois Curfman McInnes (ANL)**

## PESO PARTNERSHIPS

**STAKEHOLDER ENGAGEMENT** (Mike Heroux, ParaTools)
**CONSORTIUM PARTNERSHIPS** (Terry Turton, LANL)

**Strategic engagement with consortium partners, applications, facilities, industry and agencies**
(in collaboration with and co-funded by SSOs)

William Godoy, ORNL, On-node programming systems (w. S4PST)
Rajeev Thakur, ANL, Inter-node programming systems (w. S4PST)
Sameer Shende, Univ of Oregon, Tools (w. STEP)
Sherry Li, LBNL, Math libraries (w. FASTMath)
Berk Geveci, Kitware, Data and viz (w. RAPIDS)
Lavanya Ramakrishnan & Hannah Cohoon, LBNL, Workflows (w. SWAS)
Mahantesh Halappanavar & Marco Minutoli, PNNL, ML/AI (w. FASTMath)

**Unfunded partners: Strategic engagement with NNSA, communities of practice, applications, facilities, industry and agencies**

David Bernholdt, ORNL, RSE engagement (funded by COLABS)
Ulrike Yang, LLNL, NNSA software (funded by NNSA)
Partners at ALCF, NERSC, OLCF (funded by facilities, software integration)

**COMMUNITY DEVELOPMENT**
(Lois Curfman McInnes, ANL)

**Better Scientific Software (BSSw) Fellowship Program**

Elsa Gonsiorowski, LLNL, Coordinator
Adam Lavely, LBNL, Deputy Coordinator

**Workforce**

Mary Ann Leung, Sustainable Horizons Institute
Daniel Martin, LBNL
Suzanne Parete-Koon, ORNL, lead of HPC Workforce Action Group

## PESO SERVICES

**INTEGRATION PARTNERSHIPS**
(Jim Willenbring, SNL)

**Software portfolio management and integration**
(in collaboration with and co-funded by SSOs)

Damien Lebrun-Grandie, ORNL, On-node programming systems (w. S4PST)
Hui Zhou, ANL, Inter-node programming systems (w. S4PST)
Bill Hoffman, Kitware, Tools (w. STEP & CORSA)
Satish Balay, ANL, Math (w. FASTMath)
Patrick O'Leary, Kitware, Data & viz (w. RAPIDS)
Matteo Turilli & Mikhail Titov, BNL, Workflows (w. SWAS)
Sam Browne, SNL, NNSA software (funded by NNSA)

**SQA & SECURITY**
(David Bernholdt, ORNL)

Ross Bartlett, SNL; Berk Geveci, Kitware; Jim Willenbring, SNL

## PESO PRODUCTS

**E4S** (Sameer Shende, U Oregon)

Luke Peyralans, Wyatt Spear, Jordi Alcaraz, Erik Keever

**Spack** (Todd Gamblin, LLNL)

Greg Becker, Tammy Dahlgren

**PORT & TEST PLATFORMS (T. Gamblin and S. Shende)**

Partnership with U Oregon, cloud, etc.

**BSSw.io CONTENT (w. COLABS)**

Ross Bartlett, SNL; Keith Beattie, LBNL
Pat Grubel, LANL; Mark Miller, LLNL

Strategy & Integration – Members are part of other SSO teams & NNSA, for tight collaboration

# Maintaining HPC Software Is Challenging

**Stewarding the scientific computing software ecosystem presents unique challenges.**

**I'll use examples from my experience as Kokkos maintainer to explore these challenges.**

**My journey:**
User -> Contributor -> Developer -> Maintainer/Lead


**Kokkos in a few numbers:**
50% ECP C++ software technologies and applications
2k users registered on Slack
2.1k stars on GitHub
151 contributors
20+ developers from 7 institutions

**Kokkos' reach necessitates careful maintenance.**
**Carelessness: not catastrophic, but costly.**

What does the **maintainer** do?

- Loosely aware of the entire project
- Track ongoing work and make sure that it gets reviewed and merged in a timely manner
- Direct the orchestra of **developers** and **reviewers**
- Has final responsibility
    - Reviews when no reviewer can be found for an important contribution
    - Develops when no developer can be found to fix an important bug

**If something goes wrong, it's eventually the maintainer's fault**

# Bus Factor: How Vulnerable Are You?

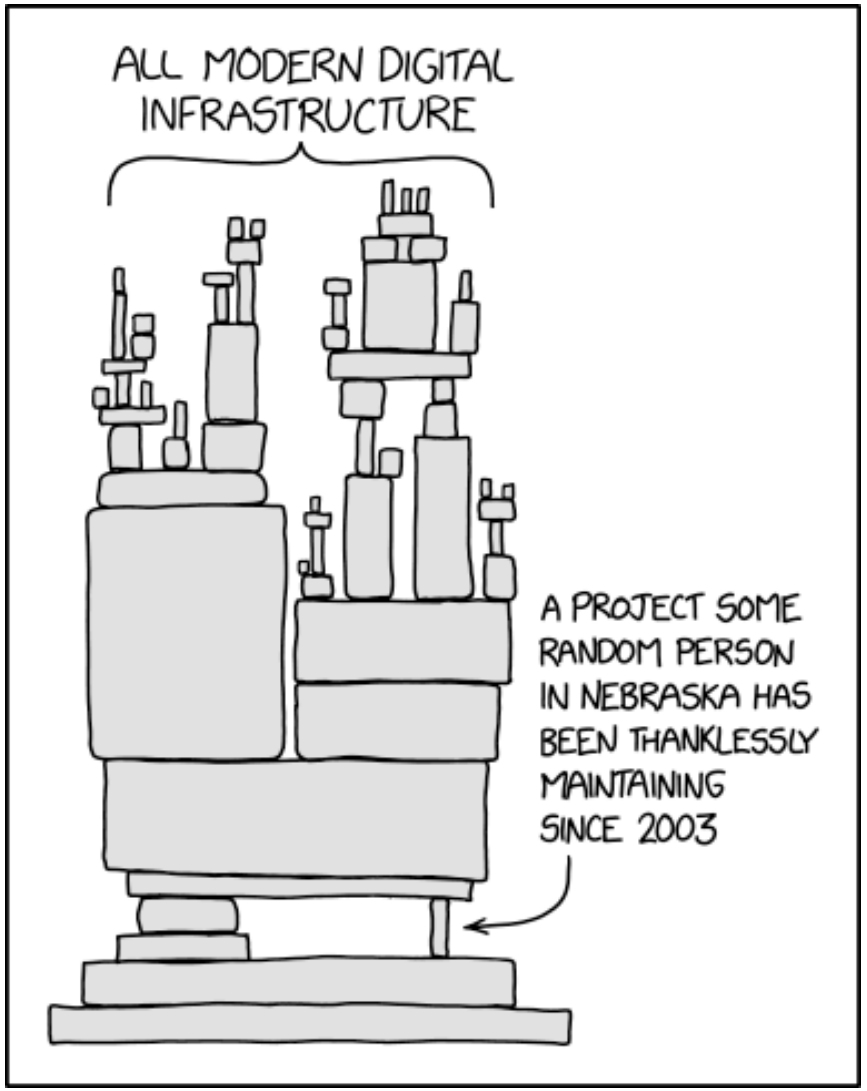
**What?** Single point of failure.

**Risks:**
Loss of critical expertise.
Stalled development and maintenance.
Increased vulnerability to bugs.
Difficulty onboarding new contributors.

**Mitigation:**
Cross-training and knowledge sharing.
Comprehensive documentation.
Modular code design.
Code reviews and pair programming.
Establish clear ownership and responsibilities.

**Kokkos' setbacks:** 3 developers gone to Google, 1 to AMD in 2019

**Distribute knowledge, especially in specialized HPC domains.**
**Be proactive and mitigate the risks.**




https://xkcd.com/2347

# The Silent Drag: Technical Debt

**What is it?** "Quick fixes" create future rework.
Performance, scalability & maintainability suffer.

**Sources:** Deadlines, legacy code, evolving hardware, lack of refactoring.

**Impact:** Slows development, increases bugs, hinders innovation,
burns out maintainers.

**Maintainer's Reality:** Constant patching, frustration, struggling to keep up.

**Solution:** Prioritize refactoring, testing, documentation, and code reviews.

**Kokkos' anecdotes:** OpenMPTarget, Qthreads, Tasking
No plan to add new backends at the start of the 3.X series.

**Technical debt is *not* always avoidable, but it *must* be managed.**
**It's a hidden cost that significantly impacts long-term sustainability.**

# Hyrum's Law: Implicit Dependencies Bite

> With a sufficient number of users of an API,
> it does not matter what you promise in the contract:
> all observable behaviors of your system
> will be depended on by somebody.

**Impact:** Hidden dependencies block change.

**Results:** Breaking changes = pain, refactoring = hard.

**Fix:** Strict APIs, testing, versioning, communication.

**Kokkos:**
Public/private headers in Kokkos 3.X – Creation of Compatibility Guidelines
View of views incident in 4.3 – Tooling and Introduction of New Semantics

**Users will use anything they can, even unintended features.**
**Be proactive and mitigate the risks.**


https://xkcd.com/1172/

# Kokkos Support Policies

- Build systems
  Supporting multiple ways to build Kokkos has a real cost in increased testing and maintenance work.
- C++ language standard
  Maintaining support for any particular C++ standard forever is impractical.
  Since C++ standards are never formally deprecated or EOL'd, need to come up with own criteria.
- Compilers
- CPU/GPU microarchitectures
- Breaking changes
  With enough users, every change is potentially a breaking change for someone.
- Backwards and future compatibility guidelines
- Deprecations
- Experimental features

**Develop and publicize support policies.**
**If you don't test it, you don't support it.**

# Closing Thoughts on Code Quality Metrics: It's Not Just a Test




**Best Practices:** OpenSSF, xSDK guide us.
Post-ECP CASS Metrics Working Group

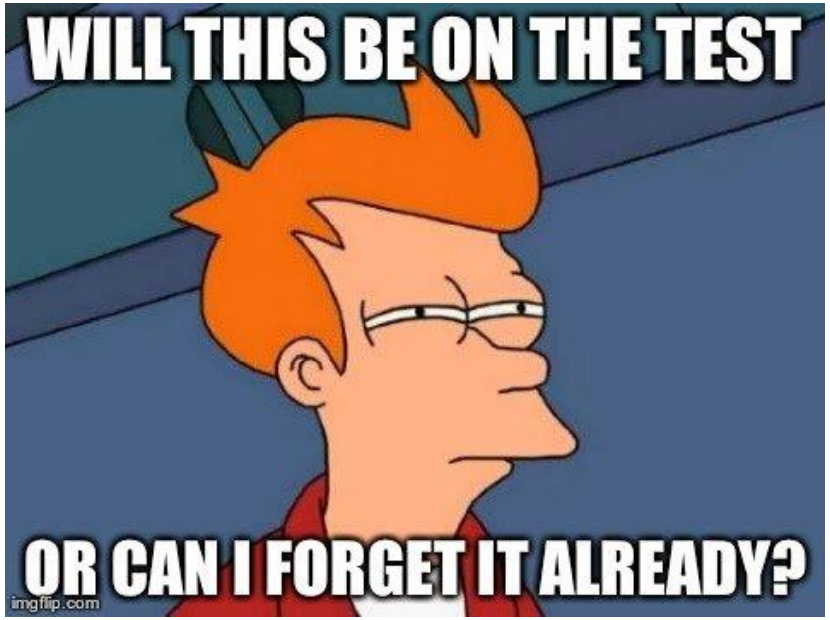
**Metrics are Tools:** Not just grades.

**Focus: Improvement:** The journey matters.

**Continuous Quality:** It's not a one-time test.

**Kokkos' efforts:**
Clang-Tidy bugprone-* checks
Contributor/Organization dependency from LFX
OpenSSF Scorecard Report

**Don't just "study for the test".**
**Metrics show where to improve, not if you're good.**



# The End

## Let's work together to build a future of sustainable, reliable, and impactful HPC software!

**Funding Acknowledgments:**

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Next-Generation Scientific Software Technologies program, under contract number DE-AC05-00OR22725 (ORNL).


Thank you for your attention.
Contact: Damien L-G <lebrungrandt@ornl.gov>