Using metrics to improve the sustainability of scientific software

ADAC PSI Seminar

Gregory R. Watson (ORNL), David E. Bernholdt (ORNL)

July 16, 2026

This work was supported by the U.S. Department of Energy (DOE) Office of Advanced Scientific Computing Research (ASCR) through the Next-Generation Scientific Software Technologies (NGSST) and Scientific Discovery through Advanced Computing (SciDAC) programs.

Overview

Introduction & Context

CASS, SSOs, and Working Groups

CASS Impact Framework

Purpose, principles, process

Dimensions of Metrics

Impact, Sustainability, and Quality metrics

Implementation Roadmap

A phased approach to sustainability assessment

CORSA Sustainability Dashboard

How to collect and measure sustainability

CASS: Stewardship and Advancement of the Scientific Software Ecosystem

- **Inward-facing activities:** Strengthening software products
  - Improve development practices, sustainability, quality, and trustworthiness
  - Enhance user experience and integration within the broader ecosystem
- **Outward-facing activities:** Community engagement and discovery
  - Curate and evolve the software portfolio
  - Help teams connect with and grow their user communities
  - Enable the broader community to discover and adopt useful software

Members

CORSA
Developing resources to provide sustainable pathways for scientific software

FASTMATH
Stewardship, advancement, and integration for math and ML/AI packages

PESO
Stewarding, evolving and integrating a cohesive ecosystem for DOE software

LEADS
Innovative, scalable and customized scientific machine learning

RAPIDS
Stewardship, advancement, and integration for data, visualization and ML/AI packages

S4PST
Stewardship, advancement and engagement for programming systems

STEP
Stewardship, advancement of software tools for understanding performance and behavior

Sponsored by the Department of Energy, Office of Advanced Scientific Computing Research

Working Groups

Impact Framework

Metrics

Integration

Software Ecosystem

User-Developer Experience

Workforce

CASS & Software Sustainability Organizations

Consortium for the Advancement of Scientific Software

Provides guidance and resources for member organizations

Helps navigate sustainability assessment and planning

Bridges software engineering research with scientific practice

Software Sustainability Organizations (SSOs)

Need to understand the status and progress of their software products

Desire to measure their effect on scientific discovery

Want to support specific scientific communities

Goal: Help research software achieve long-term sustainability and maximize scientific impact

CASS Impact Framework

Purpose and Principles

Gather information to understand each software product's progress and status

Help software teams improve their products

Identify impacts and accomplishments to highlight to sponsors and community

Focus on improvements over time, not on absolute measures or comparisons

Process

Create a "baseline" document that focuses on a starting point and pain points

Thereafter provide a document annually (or more frequently if desired)

Provide current status, progress from the previous year, plans for the next year

Identify relevant metrics for each accomplishment or plan

Work with software products to implement and gather metrics

Responds to a request from our sponsors for a common assessment framework capable of supporting the broad range of software products, purposes, and maturity levels across the CASS portfolio

Uniform But Flexible Approach

Not all software can be judged by same criteria

ParaView

The world's leading open-source post-processing visualization engine

- > 100K downloads/year (estimated)
- Canonical reference: DOI:10.1016/B978-012387582-2/50038-1 has ~3000 citations

AMReX

Software framework for massively parallel, block-structured adaptive mesh refinement applications

- 369 forks; 579 stars on GitHub
- Canonical reference: DOI: 10.21105/joss.01370 has ~430 citations

DARSHAN

HPC I/O Characterization Tool

Investigate and tune the I/O behavior of complex HPC applications

- Used primarily by HPC computing facilities
- Canonical references: DOI:10.1109/ESPT.2016.006, DOI:10.1145/3624062.3624207, DOI:10.1109/MSST.2011.5937212, ~130 total citations

Different starting points and interests

Goal: Build contributor community

- Codes of Conduct
- Ratio of outside contributors to core maintainers
- Governance Documents

Goal: Improve Documentation

- Availability of tutorials covering basic functionality
- Enforced requirement to update docs with code
- Contributor guidelines

Goal: Improve code quality

- Enforce requirement for peer code review
- Improve types of tests in test suite or code coverage of tests
- Implement continuous integration on key platforms
- Implement automated valgrind memcheck testing

The Three Dimensions of Software Sustainability

IMPACT

Adoption, citation, and integration into research workflows

SUSTAINABILITY

Community health, governance, and structural viability

QUALITY

Technical excellence, correctness, performance, reliability, and ease of use

Each dimension contains multiple metric categories with specific indicators and measurement methods

IMPACT Dimension

Assesses software influence based on adoption, citation, and integration into research workflows

Software Citation & Adoption

Example Indicators:

- Number of citations/mentions
- CITATION.cff and codemeta.json metadata
- DOI resolutions or downloads
- Dependent packages count
- Inclusion in training materials
- Spack recipe instances

Field Research Impact

Example Indicators:

- Non-bibliographic DOI references
- HPC Center recommendations
- Facility software stack listings
- Forks/stars (community uptake)
- Scientific discoveries enabled
- New methodologies supported

High-impact software attracts resources and institutional support, creating positive feedback loops

SUSTAINABILITY Dimension

Encompasses human and structural elements crucial for long-term project resilience and continuity

Governance & Community

- Codes of Conduct
- Contributor Guidelines
- Governance Documents
- Licensing & FAIR Compliance

Engagement & Outreach

- Issue Response Time
- New Contributors/Cycle
- Welcomeness Metrics
- Psychological Safety

Financial Sustainability

- Funding Source Diversity
- Multi-year Commitments
- Non-grant Income
- RSE Award Programs

Institutional Support

- Dedicated RSE Positions
- Industry Adoption
- Staff Contributions
- Career Path Support

Community Health & Longevity

Active Maintenance

- Commit activity patterns & gaps
- Maintenance mode indicators
- Release frequency analysis
- Contributor abandonment forecasting

Project Longevity

- Active contributors over time
- Contributor diversity & retention
- Knowledge distribution (bus factor)
- Succession planning indicators

Collaboration & Interoperability

- Cross-project collaborations and integration with broader scientific software ecosystem
- Dependency analysis: Spack, conda-forge, PyPI, domain-specific package managers
- API standards compliance and data format standardization
- Participation in external events, cross-project forums, domain-wide mailing lists

Vibrant communities offer redundancy in maintainership, diverse perspectives, and knowledge transfer mechanisms

QUALITY Dimension

Technical excellence and user-facing features that reduce maintenance burdens while enhancing confidence

Reliability & Robustness

Static analysis, security practices, CERT guidelines, test coverage

Development Practices

CI/CD, testing, linting, peer code review, issue trackers

Reproducibility

Containerization, FAIR4RS compliance, workflow support

Usability

Documentation, API references, tutorials, UEQ scores

Portability

Spack/Conda/CMake, containers, architecture compatibility

Performance

Benchmarking, scalability, GPU utilization, SCI metrics

Quality Measurement Infrastructure

Static Analysis	DeepSource, CodeAnt.ai, SemGrep, Valgrind, Heaptrack
Security	SAST/DAST tools, dependency scanning, CERT Guidelines compliance
Performance	HPC Challenge, SPEC benchmarks, NVIDIA Nsight, Score-P, Darshan
Environmental	ISO/IEC 21031:2024 SCI, CodeCarbon, Green Software Foundation tools

Emerging: Environmental Efficiency (SCI) and HPC Scalability metrics for sustainable computing

CASS Metrics & OpenSSF Best Practices Alignment

How CASS metrics map to the OpenSSF Best Practices Badge (Passing Level)

CASS Metric	OpenSSF Requirement	Alignment
Governance & Codes of Conduct (4.2.1)	Contribution & Governance	**Strong**
Licensing & FAIR (4.2.2)	License (OSI-approved)	**Direct**
Active Maintenance (4.2.3)	Change Control / Releases	**Aligned**
Engagement (4.2.4)	Bug Reporting / Issue Tracker	**Partial**
Reliability & Security (4.3.1)	Security & Static Analysis	**Strong**
Development Practices (4.3.2)	Testing & Version Control	**Direct**
Reproducibility (4.3.3)	Build System	**Aligned**
Portability (4.3.5)	Installation	**Aligned**

CASS Unique: Scientific Impact, Community Psychology, HPC Performance | OpenSSF Unique: Security Reporting Mechanisms

Beyond OpenSSF: What CASS Metrics Uniquely Addresses

OpenSSF focuses on security and basic development practices. CASS metrics adds critical dimensions for scientific software sustainability.

IMPACT

(Not in OpenSSF)

- Citation counts & bibliometrics
- DOI resolutions & downloads
- Dependent package analysis
- HPC facility adoption
- Training material inclusion
- Field research references
- Community uptake signals

SUSTAINABILITY

(Expanded beyond OpenSSF)

- Psychological safety metrics
- Contributor retention rates
- Bus factor analysis
- Funding source diversity
- Multi-year commitments
- Institutional RSE support
- Cross-project collaboration

QUALITY

(HPC-specific additions)

- HPC portability (Spack, E4S)
- Accelerator compatibility
- Performance benchmarks
- Energy efficiency (SCI)
- Container availability
- Cross-platform testing
- Reproducibility indicators

Why it matters: OpenSSF is necessary but not sufficient for scientific software sustainability

CASS metrics help demonstrate value to funders, track community health, and ensure long-term viability

Implementation Quick Wins

Satisfy OpenSSF Best Practices Badge while targeting CASS metrics

Immediate Wins	Ensure LICENSE, CONTRIBUTING.md, CODE_OF_CONDUCT.md, and README.md are present *CASS 4.2.1 & 4.2.2*
Process	Formalize release notes and semantic versioning practices *CASS 4.2.3*
Technical	Implement automated test suite and CI/CD pipeline *CASS 4.3.1 & 4.3.2*
Security	Create SECURITY.md describing how to report vulnerabilities privately *CASS 4.3.1*

Prioritize CASS Sections 4.2 (Sustainability) and 4.3 (Quality) to achieve OpenSSF compliance

CORSA Open-Source Document Templates

Ready-to-use guidance, examples, and templates for your project's governance documents

Governance

Minimum Viable Governance, decision-making structures, leadership models

Codes of Conduct

Contributor Covenant templates, enforcement guidance, community standards

Contributing Guides

How to contribute, development workflows, PR templates

Licenses

License selection guidance, OSI-approved options, attribution requirements

Roadmaps

Project planning templates, milestone tracking, feature prioritization

DEI Resources

Inclusive community practices, the DISCOVER Cookbook, accessibility

github.com/corsa-center/oss-documents

69 stars • CC-BY-4.0 licensed • Community-contributed examples from NumPy, Astropy, Kubernetes & more

Implementation Roadmap

A phased approach to sustainability assessment

1	2	3
Getting Started	**Building Momentum**	**Long-term Development**
Months 1-3	Months 4-12	Year 2+

Start with foundational indicators and progress to more complex metrics over time

Phase 1: Getting Started (Months 1-3)

Goal: Establish baseline understanding of project health and implement foundational governance

1. Establish baseline assessment using automated tools/dashboards for basic metrics
2. Implement essential governance documentation: CODE_OF_CONDUCT.md, CONTRIBUTING.md, LICENSE
3. Set up basic citation tracking using available APIs and tools (CORSA dashboard, Semantic Scholar, OpenAlex)
4. Assess current funding situation and document sustainability planning needs

Phase 2: Building Momentum (Months 4-12)

Goal: Deepen insights with targeted metrics and formalize sustainability planning

1. Expand community health monitoring using CORSA/CHAOSS tools where appropriate
2. Implement quality assurance practices appropriate to project size and context
3. Develop stakeholder engagement around sustainability planning
4. Adopt a sustainability dashboard for regular monitoring and reporting

CHAOSS GrimoireLab provides multi-repository analysis, AI-powered sentiment analysis, and community health indicators

Phase 3: Long-term Development (Year 2+)

Goal: Embed sustainability as a core strategic function and contribute to the broader community

1. Establish comprehensive sustainability strategy based on metric insights
2. Implement advanced measurement approaches where warranted and cost-effective
3. Contribute to validation research by sharing experiences and outcomes
4. Mentor other projects in sustainability assessment and planning

Become a leader by contributing knowledge back to the broader research software community

Avoiding Common Pitfalls

1. Metric Overload

Focus on metrics most relevant to your project context and goals. Prioritize outcomes, not raw output.

2. Automation Bias

Remember that automated tools provide indicators, not definitive assessments. Context matters.

3. Context Ignorance

Adapt frameworks to your specific domain, culture, and organizational context.

4. Validation Assumptions

Acknowledge uncertainty and use metrics as guidance rather than definitive measures.

5. Resource Mismatch

Ensure metric implementation efforts align with available resources and priorities.

6. Stakeholder Skepticism

Ensure transparent communication about why metrics are tracked and how they influence decisions.

7. Inaction on Insight

Create a process for regularly reviewing metrics and converting findings into actionable steps.

CORSA Sustainability Dashboard

A hub for showcasing the sustainability of scientific open source projects

Project Catalog

Browse and search scientific software projects with sustainability metrics at a glance

Dependencies Explorer

Visualize package dependencies across the scientific software ecosystem

Spack Dependencies

Analyze Spack package relationships and build configurations

Project Metric Visualizer

Interactive charts showing sustainability trends over time

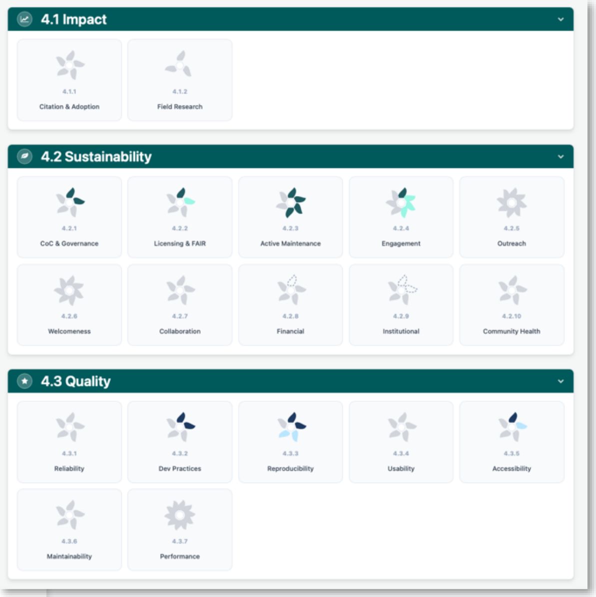
Track your software's live metrics easily from the dashboard
https://corsa.center/dashboard

Case Study: HDF5 on the Dashboard

https://corsa.center/dashboard/catalog/?category=all&repo=HDFGroup%2Fhdf5

HDF5: A Model for Sustainability Tracking

HDF5 is a high-performance data management library used across scientific domains—from climate modeling to genomics. The CORSA dashboard tracks sustainability metrics, demonstrating mature, well-maintained research software.



Dashboard Metrics Visible for HDF5:

Activity

Stars, forks, commit frequency, release history

Community

Contributors over time, issue response patterns

Dependencies

Downstream packages that depend on HDF5

Quality

CI status, documentation, license compliance

Why This Matters: Projects can **calibrate their success measures** alongside proven models like HDF5, providing stakeholders with consistent data that clearly highlights development progress.

Discovering Software Usage & Dependencies

CORSA is developing tools to help projects understand their downstream impact

The Challenge: Who is using my software?

Understanding which applications depend on your software package—directly or through third-party libraries—is crucial for measuring impact, prioritizing maintenance, and communicating value to stakeholders.

Direct Discovery

- Package manager analysis
- Repository dependency graphs
- Build system inspection

Third-Party Integration

- Spack, conda-forge, PyPI APIs
- Libraries.io, OpenAlex integration
- GitHub dependency insights

Benefits: Quantify real-world adoption • Identify high-impact dependents • Prioritize compatibility & support • Strengthen funding proposals

Key Takeaways

- Three dimensions—Impact, Sustainability, and Quality—provide a comprehensive view of software health
- Start small with foundational metrics, then expand as your project matures
- Metrics are tools for strategic planning, not absolute measures—context matters
- Avoid common pitfalls: metric overload, automation bias, and inaction on insights
- CASS is available to support you with tools, templates, and mentorship

Resources & Next Steps

CASS Website
https://cass.community

Join the community
http://eepurl.com/iRiSnY

CASS Sustainability Metrics Report
Full report with detailed metric descriptions, measurement methods, interview questions, and implementation guidance. Available under CC BY 4.0 license. https://zenodo.org/records/17704989

Related Resources

- CASS Impact Framework (https://cass.community/working-groups/impact-framework.html)
- CORSA OSS Documents (https://github.com/corsa-center/oss-documents)
- Measuring What Matters in the Age of AI: The Research Software Metrics Landscape (10.13140/RG.2.2.25849.63847)
- Open-Source Software Foundations Are Rarely Direct Funding Mechanisms: What Scientific Software Projects Need, and What They Get (10.13140/RG.2.2.19138.75200)
- CHAOSS Community Metrics (https://www.chaoss.community/kbtopic/community)
- OpenSSF Best Practices Badge (https://www.bestpractices.dev/en)
- FAIR4RS Principles (10.1038/s41597-022-01710-x)

Join the CASS mailing list to stay informed of news and events!

Thank You!

Questions & Discussion