Outreach for Better Scientific Software

David E. Bernholdt
Oak Ridge National Laboratory
bernholdtde@ornl.gov

for the IDEAS Productivity Project

(Co-lead institution)

ORNL is managed by UT-Battelle for the US Department of Energy
License, citation and acknowledgements

License and Citation

• This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).


Acknowledgements

• The entire IDEAS Productivity Project team has contributed to the results presented
  – Particularly Lois Curfman McInnes and Mike Heroux for presentation materials

• This work was supported by the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research (ASCR), and by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.

• This work was performed in part at the Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC for the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.
How Do We Increase Credibility of Scientific Software?

• Reproducibility
  – The ability of independent investigators to repeat the work using independent tools

• Replicability / Repeatability
  – The ability to repeat the work using the same tools
  – Independent investigators or the original investigators

• Validation
  – Are we building the right system?
  – Preferably compare with experiment/observation
  – Possibly compare with other numerical experiments (aka cross-comparison)
    • Other implementations of the same model, or different models

• Verification
  – Are we building the system right? (Does it meet the specifications?)
  – Software testing
  – Software quality
Credibility Relies on Good Software Engineering

• But do we do good software engineering???
• Some projects do…
• Why doesn’t every project?
  – The current culture of scientific computing doesn’t value it
    • Sponsors emphasize scientific advances
    • Evaluation emphasizes papers and proposals over software
  – Harder to publish
    • Scientific results > models/methods/algorithms >> software development
  – Lack of knowledge/experience
  – Limited time and other priorities

2018-07-03 PASC18 Basel/MS32-Increasing Credibility
The IDEAS Productivity Project

• IDEAS = Interoperable Design of Extreme-Scale Application Software

• IDEAS-Classic
  – Funded by DOE / ASCR + BER Terrestrial Ecosystem Modeling
  – Improving interoperability of key DOE numerical libraries

• IDEAS-ECP
  – Funded by Exascale Computing Project (ECP)

• Fundamental goals
  – Increase the awareness of, and attention to, software development, productivity, and sustainability throughout the HPC/computational science and engineering (CSE) software community
  – Engage with...
    • Research sponsors and other stakeholders
    • Specific software development teams
    • Broader HPC/CSE community
Stakeholder Engagement

Responding to the Software Crisis in DOE Scientific Computing

- Half-day minisymposium, held Sep. 2015 at DOE HQ
- Audience: DOE program managers
- Presentations by members of the DOE community
- Panel of DOE program managers
- https://ideas-productivity.org/resources/workshops/2015-09-software-crisis/

Computational Science & Engineering Software Sustainability and Productivity (CSESSP) Challenges

- Led by Mike Heroux and Gabrielle Allen
- Sponsored by the National Information Technology Research and Development Program (NITRD)
- Held Oct. 2015
- 2 days, 85 participants
- Workshop report available
  - https://www.nitrd.gov/csessp/
Working with Software Development Teams

• CSE code teams vary widely in the...
  – Depth of understanding of their own software development processes
  – Ability to identify and diagnose challenges they might face
  – Ability and comfort level to choose and implement methods and tools to improve the situation

• Need different kinds of support...
  – Self-starters may benefit from a little information/guidance/experience
  – Success stories provide inspiration and examples
  – Some may need deeper study – tutorials, books, etc.
  – Some may need help understanding their situation and possible solutions
Cutting Through Analysis Paralysis

• Many projects want to improve their software development practices, but there are too many “targets” and they end up paralyzed

• Other projects feel that things could/should better, but don’t know how to go about diagnosing the problem, much less implementing a solution

• PSIP = Productivity and Sustainability Improvement Plan
  – Start with an interview to gain a baseline understanding of the project
  – Then start improving one thing at a time
  – Repeat until satisfied
  – https://github.com/betterscientificsoftware/PSIP-Tools

Goal for each iteration is to make an improvement that allows more progress at a lower cost
Start

Workflow for Productivity and Sustainability Improvement Plans

- Summarize Current Project Practices
  - Write brief practices summary document.
  - High level description, a few pages.

- Set Goals
  - Identify practices ready for improvement.
  - Select those with near-term payoff.

- Construct Progress Tracking Card (PTC)
  - Construct from PTC catalog.
  - Select only a few items.

- Record Current PTC Values
  - Set baseline values for future reference.

- Create Plan For Increasing PTC values
  - Define practice improvement steps.
  - Be specific, track issues.

- Execute Plan
  - Increase PTC values by improving selected practices.
  - Track issues progress.

- Assess Progress
  - Track PTC values.
  - Adjust strategy if needed.

Repeat
- Start process again.
Tutorials

- A tried and true way of helping people learn

- Many venues in HPC/CSE
  - Supercomputing and ISC
  - SIAM CSE and PP
  - Argonne Training Program in Extreme-Scale Computing (ATPESC)
  - Computing facilities

- IDEAS produces its own tutorials...
  - E.g., Software engineering and community codes, Good scientific process requires software engineering practices, Tools for controlling change in your software, Introduction to GNU Make and Autotools, Documenting your code, Testing your code, Software refactoring, Planning simulations, Improving Reproducibility through better software practices, Better (small) scientific software teams, Introduction to software licensing, ...

- … And helps to organize other providers
  - E.g., “Modern” C++, Cmake, …
An Aside: Consider HPC Carpentry

- The Software Carpentry movement has pioneered a new approach to teaching computing skills

- Now morphed into The Carpentries
  - Umbrella for numerous “Lesson Organizations”
  - [https://carpentries.org/](https://carpentries.org/)

- The first CarpentryCon was held in Dublin at the end of May
  - 110 participants from around the world

- Workshop on HPC Carpentry
  - Attracted ~30 attendees
  - In 3 hrs we produced the initial draft of an “HPC Novice” lesson
  - Expect follow-up and expansion

- Follow the discussions: [https://carpentries.topicbox.com/groups/discuss-hpc](https://carpentries.topicbox.com/groups/discuss-hpc)
Another Aside: Learning About Learning

• We do a lot of learning outside of a formal classroom setting
  – 95% of our time is spent outside the classroom
• We are busy, our time to learn is limited
• There are a myriad of resources available in many different formats and media
• Trans-media learning frameworks
  – Taking advantage of the different types of content available in different media to learn more efficiently and more effectively (connect and integrate)
  – Leverage trends in technology, grounded in social science research
• Tutorial at ECP 2nd Annual Meeting, HPC-BP webinar
  – Elaine Raybourn, Sandia National Laboratories
  – https://ideas-productivity.org/events/hpc-best-practices-webinars/#webinar018
Best Practices for HPC Software Developers Webinar Series (HPC-BP)

• Initially 7 webinars between May-August 2016, rebooted as a monthly series in June 2017
  – Initially (2016), alternating, software best practices and other HPC topics
  – In collaboration with DOE/ASCR computing facilities (ALCF, NERSC, OLCF)
  – Now, strong emphasis on software practices

• Total of 19 webinars to date, offered live and archived
  – https://ideas-productivity.org/events/hpc-best-practices-webinars/
  – Mailing list of ~1200 people who have registered for a webinar or opted in for notifications
  – Also announced to ECP members, DOE computing facility users, etc.

• Recent topics include…
  – Python in HPC, Intermediate Git, Using the roofline model, Barely sufficient project management, Managing defects in HPC software development, Bringing best practices to long-lived production codes, Jupyter and HPC, Software development with Eclipse, Software citation, On-demand learning, Reproducible computational pipelines
More Outreach: Promoting Conversations about Software

• We help opportunities to talk about software development, productivity, and sustainability in more “academic” environments
  – https://ideas-productivity.org/events/

• Minisymposia
  – PASC (2018)

• Thematic poster sessions
  – SIAM CSE (2017)

• Birds of a Feather sessions
  – Software Engineering and Reuse for Computational Science and Engineering
    • SC15, SC16, SC17, see http://bit.ly/swe-cse-bof

SIAM CSE17 Miniposterium posters, archived on FigShare
Building an Online Community

https://bssw.io

• New community-based resource for scientific software improvement exchange
• A central hub for sharing information on practices, techniques, experiences, and tools to improve developer productivity and software sustainability for computational science & engineering (CSE)

Goals

• Raise awareness of the importance of good software practices to scientific productivity and to the quality and reliability of computationally-based scientific results
• Raise awareness of the increasing challenges facing CSE software developers as high-end computing heads to extreme scales

Site users can…

• Find information on scientific software topics
• Propose to curate or create new content based on their own experiences. The backend enables collaborative content development using standard GitHub tools and processes
Resources For Developer Productivity And Software Sustainability

Better Planning
Strategies for planning in order to improve software productivity, quality, and sustainability.

Better Development
Aspects of scientific software development that should be systematically addressed in order to improve software productivity, quality, and sustainability.

Better Performance
Approaches for developing code that is efficient, scalable, and portable—from laptops to emerging extreme-scale architectures.

Better Reliability
Methods for testing and verification to ensure that software is robust and produces reliable results.

Better Collaboration
Ways to facilitate and distribute work across teams, promote partnerships via software, and contribute to stronger communities.

Better Skills
Ways to improve productivity and sustainability from an individual perspective.

View All Resources
BSSw Resource Topics

Better Performance:
- High-performance computing
- Performance at LCFs
- Performance portability

Better Planning:
- Requirements
- Design
- Software interoperability

Better Development:
- Documentation
- Version control
- Configuration and builds
- Deployment
- Issue tracking
- Refactoring
- Software engineering
- Programming languages and tools

Better Reliability:
- Testing
- Continuous integration testing
- Reproducibility
- Debugging

Better Skills:
- Personal productivity and sustainability
- Online learning

Better Collaboration:
- Licensing
- Strategies for more effective teams
- Funding sources and programs
- Projects and organizations
- Software publishing and citation
- Discussion and question sites

Software Productivity & Sustainability

Planning
Development
Reliability
Skills
Collaboration

Site content spans a broad range of topics.
Resource examples

Curated links: A brief article that highlights other web-based articles or content. Your article should describe why the CSE community might find value.

An Introduction To Software Licensing

This tutorial provides a brief introduction to software copyright and licensing for researchers in computational science and engineering. Explains the difference between closed and open source software, and copyleft and permissive open source licenses. Outlines a variety of factors researchers might want to consider when selecting a software license. Provides links to some key web resources as a starting point for deeper exploration.

Prerequisites

What Is Software Intellectual Property?

https://bssw.io/resources/an-introduction-to-software-licensing

Planning For Better Software: PSIP Tools

Scientific software teams are typically focused on the creation of a new set of features that will enable the next set of computational experiments. Teams seldom have the time to stop development and focus solely on improving productivity or sustainability. However, teams can incorporate improvements on the way to developing new science capabilities.

Prerequisites

CSE Software Requirements?

What Are Strategies For More Effective Teams?

PUBLISHED NOVEMBER 21, 2017
CONTRIBUTOR MIKE HERDOUX

The Productivity and Sustainability Improvement Planning (PSIP) process recognizes that productivity and sustainability improvements for scientific software benefit from an incremental, iterative approach. The PSIP-Tools GitHub repo is a collection of documents that enable the adoption and use of PSIP for a software team. The PSIP-Tools repo contains everything from a template for the first introduction letter to a complete interview guide, interview prompts and expected timeline.

The PSIP process has been successfully used to help scientific software teams achieve incremental, sustainable process improvement, while still achieving their science goals.

https://bssw.io/resources/planning-for-better-software-psip-tools

https://bssw.io/resources/an-introduction-to-software-licensing
BSSw blog articles

Better Scientific Software (BSSw) presents articles from expert community members on topics related to software productivity and sustainability.

Would you like to share your ideas through a blog article? The BSSw blog provides a platform to inform, inspire, and mobilize the community toward better software practices. Please see details on how to contribute to BSSw.
Communities Overview
The Better Scientific Software umbrella encompasses a rich variety of communities who are working to advance the methods, practices, and processes of CSE software.

Community landing pages, tailored to unique perspectives and priorities, provide a variety of starting points for using the BSSw site and promote a shared understanding of CSE software issues. Curators of a community landing page can customize content to serve the needs community members through highlighted resources and other custom content.

Better Scientific Software Communities:
- Exascale Computing Community
- Scientific Libraries Community
- Community of Supercomputing Facilities and Their Users
- Software Engineering Community
- Environmental System Science Community

We want your input and perspectives. Please contact us if you would like to start a community-specific landing page.
Community landing pages

The Community Of Supercomputer Facilities And Their Users

Many computational scientists access HPC resources through centralized supercomputing facilities, which serve many users.

Learn More About Communities

Portability Across DOE Office Of Science HPC Facilities

Achieving good performance across a range of platforms is an important goal for high-performance CSE software.

Prerequisites

- How To Enable Performance Portability For CSE Applications?
- How To Estimate Operational Intensity
- What Is Performance Portability For CSE Applications?

PUBLISHED FEBRUARY 86, 2018  CONTRIBUTOR TIM WILLIAMS

This DOE performance portability website is intended to be a living-growing documentation hub and guide for applications teams targeting systems at multiple DOE Office of Science facilities. The site discusses differences between the systems, the software environments, and the job-submission processes. The site also discusses how to define and measure performance portability and provides recommendations based on case studies for the most promising performance-portable programming approaches.

Contributor
Tim Williams, ANL
BSSw current status

• BSSw site now includes a sampling of resources
  – But many topics need content!
• Current site is a **starting point for CSE community collaboration** to share information on developer productivity and software sustainability
• Over time, build up rich content resources

**We need your contributions!**

Github profile pics and links of all site contributors go here.
BSSw Fellowship Program

We are looking for applications from people with the following characteristics:

• Passionate about scientific software
• Interested in contributing powerful ideas, tools, methodologies, and more that improve the quality of scientific software
• Able to use the fellowship to broadly benefit the scientific software community
• Willing to participate as an alum in subsequent years to guide selection of future fellows and promote better scientific software in their community

Look for announcement for 2019 class in Fall 2018

BSSw Fellowship Application Form

Question Summary

Experience:
• Describe your work relevant to scientific software (1000 - 1500 characters).
• Describe your background and experience relevant to being a BSSw Fellow (1000 - 1500 characters).

Proposed work and impact:
• What would you do as a BSSw Fellow? (1000 - 1500 characters).
• What impact do you foresee from your efforts? (1000 - 1500 characters).
2018 BSSw Fellows

**Jeffrey Carver**
*University of Alabama*
Professor, Computer Science

**BSSw Focus:** Contemporary peer code review in scientific software development

**Daniel S. Katz**
*University of Illinois at Urbana Champaign*
Assistant Director for Scientific Software and Applications, NCSA

**BSSw Focus:** Making scientific software more sustainable by providing credit to its developers via software citation

**Ivo Jimenez**
*University of California Santa Cruz*
Ph.D. Candidate, Computer Science

**BSSw Focus:** Generation of computational experimentation pipelines that are easy to re-execute and validate

**Andrew Lumsdaine**
*Pacific Northwest National Laboratory*
Fellow, Advanced Computing, Mathematics, and Data Division

**BSSw Focus:** Practices for high-performance and high-quality scientific software in modern C++
Putting it All Together: Outreach for Better Scientific Software

- Tutorials
- HPC-BP Webinars
- Minisymposia, Workshops, BOFs, etc.
- Better Scientific Software

HPC/CSE Community

ECP Code Teams
Join Us in Reaching Out for Better Scientific Software!

• Be part of the community talking about and acting on software development, productivity, and sustainability!
• Participate in the discussion that follows
• Visit us at our poster (CSM-14), 7:30-9:30pm today

• IDEAS Productivity Project: https://ideas-productivity.org/
  – Events: Upcoming and Past
    • Tutorials, minisymposia, posters, presentations, birds-of-a-feather, etc.
    • HPC Best Practices Webinar Series
  – Email Us: IDEASProductivity@gmail.com

• Better Scientific Software: https://bssw.io/
  – Contributing: https://bssw.io/contributes/new
  – Mailing list: https://bssw.io/pages/receive-our-email-digest (~1/month)
  – Contact Us: https://bssw.io/contact