An Overview of Research Software Science Approaches

ECP EXASCALE COMPUTING PROJECT

Michael A. Heroux
Senior Scientist, Sandia National Laboratories
Director of Software Technology, US Exascale Computing Project
Scientist in Residence, St. John's University, MN

NNSA National Nuclear Security Administration

U.S. DEPARTMENT OF ENERGY | Office of Science

Outline

- A brief and partial history of the pursuit for better scientific software development and use
- Characterization of Research Software Science (RSS)
- RSS Why and How
- RSS in the context of the US Exascale Computing Project
- Leadership Scientific Software Initiative (https://lssw.io)
- The Science of Scientific-Software Development and Use (Workshop and Report)
- Trends that increase the value of RSS

A Brief and Partial History

- Software Sustainability Institute (SSI), U.K.
  - International leader, https://www.software.ac.uk
  - Mentor to IDEAS Project, e.g., BSSw Fellows
- US National Science Foundation
  - SI2/CSSI: Direct funding for broadly used products in the scientific computing community
  - URSSI: U.S. Research Software Sustainability Institute, focused on workshop-related topics, https://urssi.us
- Research Software Engineering (RSE) Movement
  - Increasingly recognizable career track
  - Growing number of people who consider themselves part of the RSE community
  - https://society-rse.org and https://us-rse.org
- IDEAS Productivity Projects
  - IDEAS-Classic (2014), IDEAS-ECP (2017), IDEAS-Watersheds (2019), xSDK (2014, 2017)
  - Focus on developer productivity and software sustainability, communities of practice
  - Research Software Science (RSS) – Inspiration for this workshop

What is Research Software Science?

- Definition: *Applying the scientific method to understanding and improving how software is developed and used for research*

  - **Scientific Method**
    - Use formal observation and experimentation to obtain & disseminate knowledge
    - Current approach is ad hoc, engineered: See a problem, explore options to improve, pick one, move on
    - Yes, there is software engineering research, so let’s call it science too
  - **Understanding and Improving**
    - Obtain data to detect correlation, design experiments to identify cause and effect
  - **Developed and Used**
    - Developer/User, User-only, individuals, teams, communities
    - Leverage cognitive and social sciences
  - **Research**
    - Focus on software used in service of scientific advances

Origin of My Use of Research Software Science

- Founding member of IDEAS Productivity & Sustainability Project (https://ideas-productivity.org)
  - A US DOE Office of **Science** Advanced Scientific Computing Research (ASCR) sponsored project, started 2014
  - Focused on improving scientific software development and use – considered an engineering project
  - **Challenge: Struggled to articulate how scope of IDEAS project could become part of core ASCR**
  - Project continued and gained success, but how to fit efforts into ASCR core persisted for five years
- 2019: Listening to a Michael Lewis book on the history of data science, inspired the RSS term
  - A scientific focus opens the door to Office of Science funding!
  - How different is Research Software Science from software engineering research?
    - Lots of overlap, but using the term science is important
    - As scientists, we appreciate and practice science
    - expanding the scope to include our software development and use activities is natural!

Original RSS article:
https://bssw.io/blog_posts/research-software-science-a-scientific-approach-to-understanding-and-improving-how-we-develop-and-use-software-for-research

RSS Components

- Technical component
  - Research software addresses highly technical domains
    - Participation requires advanced degrees, on-going participation in domain community – significant time investment
    - Reason why “off-the-shelf” software tools & processes often need adaptation, or may not address high-priority needs
- Social component
  - Scientific software development and use are increasingly a team (and team of teams) activity
  - Teams often composed of members who are unaware (and uninterested?) in exploring human factors
  - Community engagement is increasingly important
- Cognitive component
  - Research software community members are problem solvers, love new and challenging problems
  - Are also sometimes described as “herds of cats”, resistant to prescriptive approaches

Why Research Software Science, not just Engineering?

- But isn’t RSS just an extension of RSE?
- Yes, somewhat, but not completely
- Engineers learn in order to build
  - Want an improved tool or process
  - ID a few possibilities, test, select best, move on – only incidental team memory, no focus on dissemination
- Scientist build in order to learn
  - Want to understand underlying principles, correlation, cause-and-effect
  - Design studies, capture data, analyze, publish
- Doesn’t the software engineering community do research?
  - Yes, but not always directly applicable to research software
  - Yes, but then let’s call it what it is: science

> “A scientist builds in order to learn; an engineer learns in order to build.”
> - Fred Brooks

Why A Software Science Focus now: The “No CS” Scenario

Scenario: Suppose our research centers had no formally trained computer scientists and CS work had to be done by people who learned it on their own, or just happened to study a bit of CS as part of their other formal training. This situation is undesirable in three ways:

1. We have non-experts doing CS work, making them less available in their expertise (opportunity cost)
2. CS work takes a long time to complete compared to other work (effort cost)
3. We get suboptimal results and pay high ongoing maintenance cost (quality cost)

Replace ”CS” with “Software” in this scenario and the situation describes much scientific software today

Why focus on software science now:

- The role of software has become central to much of our work and the knowledge base is too sophisticated to rely only on software non-experts
- Scientific software success depends on producing high-quality, sustainable software products
- Investing in software as a first-class pursuit improves the whole scientific ecosystem

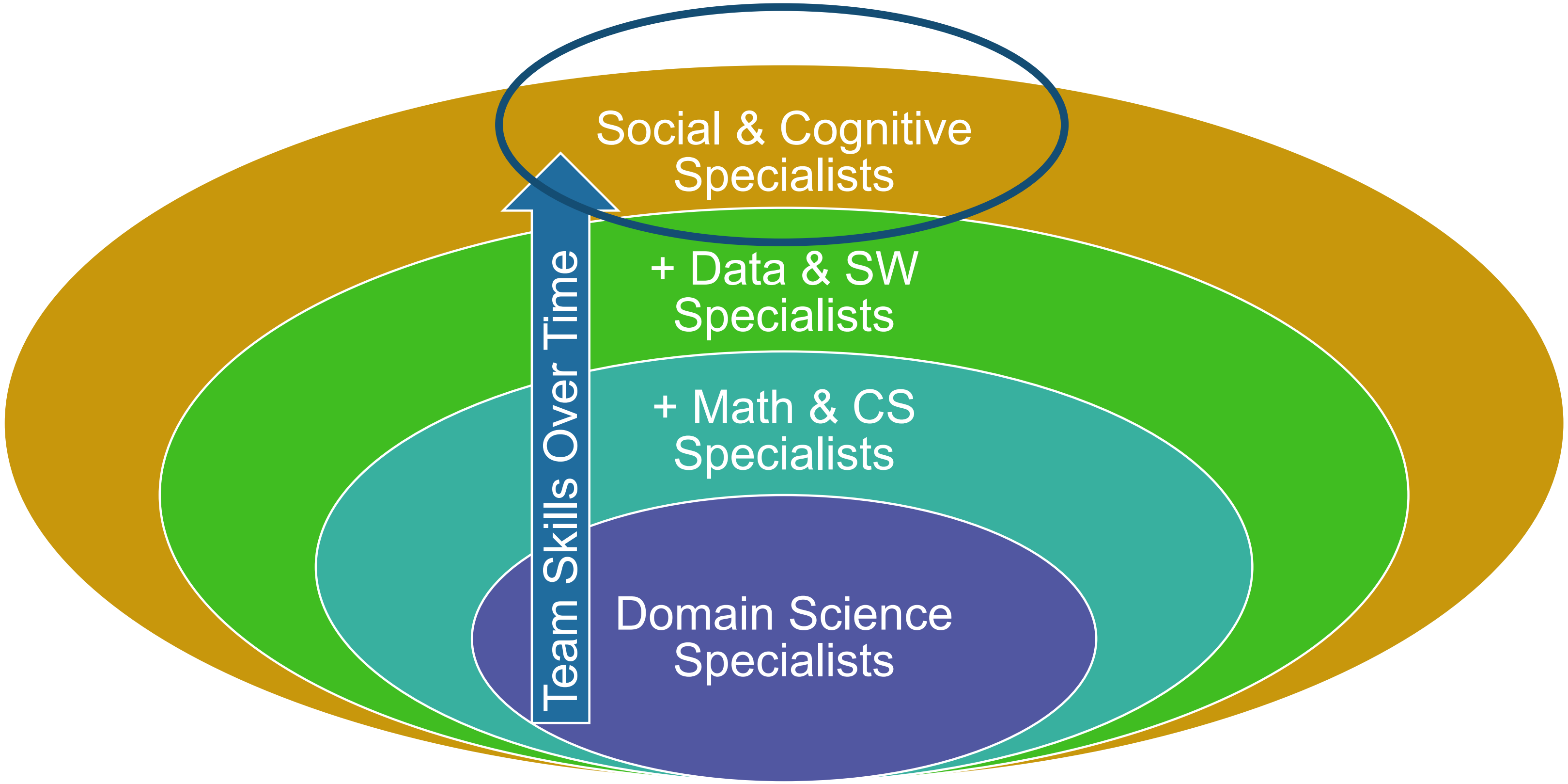
Expanding Software Team Skills: Research Software Science (RSS)

Key observation: We are scientists, problem solvers. **Let's use science to address our challenges!**

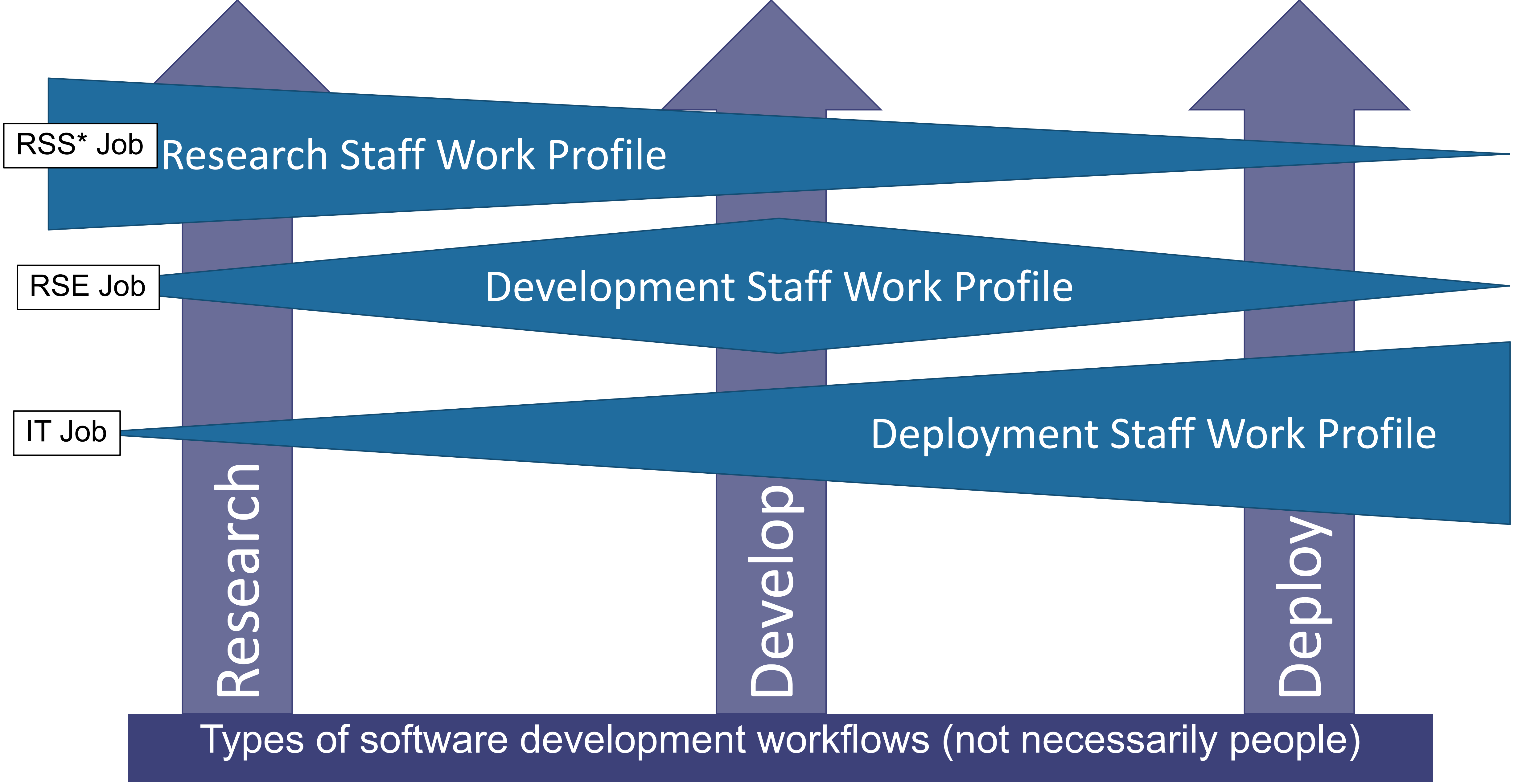
Now: Improved SW environments (Jupyter), integration of software specialists as team members, data mining of repos

Next: **Research Software Science**

- Use scientific method to understand, improve development & use of software for research.
- **Incorporate cognitive & social sciences.**



https://bssw.io/blog_posts/research-software-science-a-scientific-approach-to-understanding-and-improving-how-we-develop-and-use-software-for-research



RSS* Job

Research Staff Work Profile

RSE Job

Development Staff Work Profile

IT Job

Deployment Staff Work Profile

Research

Develop

Deploy

Types of software development workflows (not necessarily people)

*RSS – Research Software Scientist (new job type)

Applying Social & Cognitive Science to Software Teams

- Reed Milewicz – Emerging research software scientist
- Elaine Raybourn – Sandia social scientist
- New scientific tools to study and improve developer productivity, software sustainability
- Correlation: Happiness and connectedness

New Professional Role: *Research Software Scientist*

Talk to Me: A Case Study on Coordinating Expertise in Large-Scale Scientific Software Projects

Reed Milewicz and Elaine M. Raybourn
Sandia National Laboratories, 1611 Innovation Pkwy SE, Albuquerque, New Mexico 87123

***Abstract*—Large-scale collaborative scientific software projects require more knowledge than any one person typically possesses. This makes coordination and communication of knowledge and expertise a key factor in creating and safeguarding software quality, without which we cannot have sustainable software. However, as researchers attempt to scale up the production of software, they are confronted by problems of awareness and understanding. This presents an opportunity to develop better practices and tools that directly address these challeng**
that end, we conducted a case study of developers of th
project. We surveyed the software development
dressed and show how those problems are
they know and how they communicat
provide a series of practicable rec
path forward for future research.
what
data, we
, and outline a

people of different backgrounds
success, there must be clos
parties [2]. Unfortu
projects, we a
and cult
wh
ves; to have
action among those
empt to scale up these
barriers – logistical, technical,
it hard for people to share and apply
hese challenges increase both the cost and
software development and maintenance which
ately threatens sustainability.

From a software engineering perspective, more work is needed to create better tools and methodologies to manage and maintain that software development knowledge. However, as Dennehy and Conboy observe, the culture and context of a software project are "critical determinants of software development success", and that "a method, practice, or tool

Source: https://arxiv.org/pdf/1809.06317.pdf

Research Scientific Software and the US Exascale Computing Project (ECP)

EXASCALE COMPUTING PROJECT

ECP is a complex system

- ECP is specifically about preparing ~25 application code for Exascale computing systems and creating a sustainable underlying software stack, but it is much more
- The complexity of the ECP mission (cost, scope and schedule) require innovation, risk taking
- Requirements and solutions emerge over time
- ECP is a big human experiment: Why I joined!

ECP lesser-known outcome: A sustainable, reusable software ecosystem

- ECP is driving the creation of a portfolio approach for reusable scientific software:
  - Available to you from laptops to supercomputers
  - Portable across CPU and GPU architectures
  - Available as open source for you to use, contribute to, and collaborate with
- Creating a future software organization that is a first-class citizen in the leadership computing ecosystem
- And You:
  - Consider using E4S: https://e4s-project.github.io/download.html
  - Consider contributing to E4S: https://e4s-project.github.io/join.html
  - Consider contributing to one of the SDKs, e.g.: https://xsdk.info

ECP lesser-known outcome: Building a better future

- Improving how we do our work
- Engaging a broader community
- And You:
  - Receive our BSSw email digest: https://bssw.io/pages/receive-our-email-digest
  - Contribute to BSSw.io: https://bssw.io/pages/what-to-contribute-content-for-better-scientific-software
  - Apply for 2023 BSSw Fellowship (summer 2022): https://bssw.io/fellowship
  - Attend:
    - HPC Best Practices Webinars: https://ideas-productivity.org/events/hpc-best-practices-webinars
    - Working Remotely Panels: https://ideas-productivity.org/events/strategies-for-working-remotely-panels
    - Our tutorials: https://bssw-tutorial.github.io

ECP lesser-known outcome: A sustainable software R&D community

- Growing an R&D community of diverse contributors toward exploring and realizing leadership computing, generation to generation
- Exploring emerging algorithms, software, and computing platforms to deliver capabilities for solving emerging scientific problems
- And You …

Leadership Scientific Software Community Discussions

Leadership Scientific Software (LSSw) Portal https://lssw.io

The LSSw portal is dedicated to building community and understanding around the development and sustainable delivery of leadership scientific software

- LSSw Town Hall Meetings (ongoing)
  - 3rd Thursday each month, 3 – 4:30 pm Eastern US time
  - 100+ attendees at each meeting, sessions recorded
- Town Hall Topics
  - Meeting 1: Overview of the ECP Software Technology Focus Area
  - Meeting 2: Progress, impediments, priorities & gaps in LSSw panel
  - Meeting 3: Expanding the LSSw User Communities - Panel
  - Meeting 4: Expanding the LSSw Developer Communities – Panel
  - Meeting 5: Retrospective on Previous Meetings – Discussion
  - Meeting 6: Other HPC Software Ecosystems – Panel
  - Meeting 7: Expanding the Scope of What is Reusable – Panel
  - Meeting 8: Brainstorm topics
  - Meeting 9: Discussion of US DOE SW Sustainability RFI Responses

LSSw Town Halls – Forum for exploring future, building community

Topic: Progress, impediments, priorities and gaps in leadership scientific software

- Ann Almgren, Berkeley Lab, PI of the AMReX project
- Todd Gamblin, Lawrence Livermore National Lab, PI of the Spack project
- Paul Kent, Oak Ridge National Lab, PI of the QMCPACK project
- J. David Moulton, Los Alamos National Lab, PI of the IDEAS Watersheds project
- Todd Munson, Argonne National Lab, PI of the PETSc/TAO project

Themes:
- Improved SW quality and availability accelerates scientific discovery
- Maintaining SW workforce is essential through visible, sustained career paths
- Engaging, growing, & sustaining a user base is essential for viable products
- Regular testing & integration are essential for providing trusted SW components
- Complexity is growing in many dimensions, coordinated SW efforts can mitigate it

Topic: US Agency Use of DOE HPC Software

- Shawn Brown, Pittsburgh Supercomputing Center
- Jeff Durachta, NOAA
- Alice Koniges, University of Hawai’i Data Science Center
- Piyush Mehrotra, NASA
- Andrew Wissink, US Army

Themes:
- Open-source community-based software products are attractive resources
- Heterogeneous platforms (GPUs) represent a significant challenge for apps
- Lack of stable programming environments, transition costs are blockers for GPUs
- Spack is used or is on the radar for all panelist communities
- DOE math libs, perf tools, portability layers & E4S used or on the radar of most

Topic: Expanding Leadership Scientific Software Developer and User Communities

- Deb Agarwal, Berkeley Lab
- Anshu Dubey, Argonne National Laboratory
- Bill Hart, Sandia National Labs
- Addi Malviya-Thakur, Oak Ridge National Laboratory
- Katherine Riley, Argonne National Laboratory

Themes:
- All panelists support the expanded definition of leadership to include their domain
- New leadership definition enables holistic strategy for quality scientific SW
- The represented communities have much in common with HPC communities
- In future, HPC and these communities have emerging collaboration opportunities
- SW practices & tools from these communities can help HPC teams improve

Topic: Scientific Software Ecosystems

- Anita Carleton, CMU, SEI
- Theresa Windus, Iowa State, MolSSI
- Lorraine Hwang, UC Davis, CIG
- Elizabeth Sexton-Kennedy, Fermi Lab, HSF
- Andy Terrel, Xometry, NumFocus

Themes:
- Ecosystem membership criteria tend to be informal for most ecosystems.
- A product is welcome if it has a user community, funding & fits in the ecosystem
- Most ecosystems have a lean budget and live on soft funding
- A major contribution of ecosystems is training: developers, user, leaders
- With some exceptions, software quality criteria are not explicitly stated

LSSw Meeting 10: Thursday, July 21, 2022, 3 - 4:30 pm ET

Topic: Expanding Laboratory, University, and Industry Collaborations: An Industry Panel Discussion

- **Description:** The open-source scientific software community benefits from complementary and leveraged contributions from universities, laboratories, and industry. Numerous partnerships are already in place but more opportunities exist. The cost of making high-quality scientific software libraries and tools has decreased due to widely used tools and platforms such as GitHub, and the need for high-quality software ecosystems has increased due to growing scientific demands and increased interconnection between scientific disciplines. The importance of collaboration in sustaining and leveraging laboratory, university, and industry investments is even more important as we go forward. In this panel discussion, we bring community members with strong industry experience together to explore how we can further improve leverage and complementarity so that the whole scientific community can realize the benefits of new software capabilities as they emerge.
- This month our panelists are:
  - John Cary, Tech-X Corp
  - Barbara Chapman, HPE, Inc (Tentative)
  - Jeff Larkin, NVIDIA Corp
  - Bob Lucas, ANSYS, Inc
  - Pat Quillen, MathWorks, Inc (Tentative)
- In opening remarks, panelists briefly address the following questions from their perspectives:
  - What are some existing examples of scientific software collaboration between federal agency-sponsored programs (at labs and universities) and software vendor product development?
  - What has worked and not worked well with past leverage and complementarity efforts?
  - What are some near-term opportunities to improve leverage and complementarity?
  - What are some long-term opportunities and constraints on leverage and complementarity?
- Why attend: To discuss the feasibility, strategies, and opportunities for improved leverage and complementarity of agency and industry.

https://lssw.io

First-of-a-kind US DOE Workshop

- The Science of Scientific-Software Development and Use
  - Dec 13 – 16, 2021
  - https://www.orau.gov/SSSDU2021
- Workshop Brochure available:
  - https://doi.org/10.2172/1846008
- Workshop Report in progress:
  - 3 Priority Research Directions
  - 3 Cross-cutting Themes

BASIC RESEARCH NEEDS IN

The Science of Scientific Software Development and Use

Investment in Software is Investment in Science

SSSDU Priority Research Directions

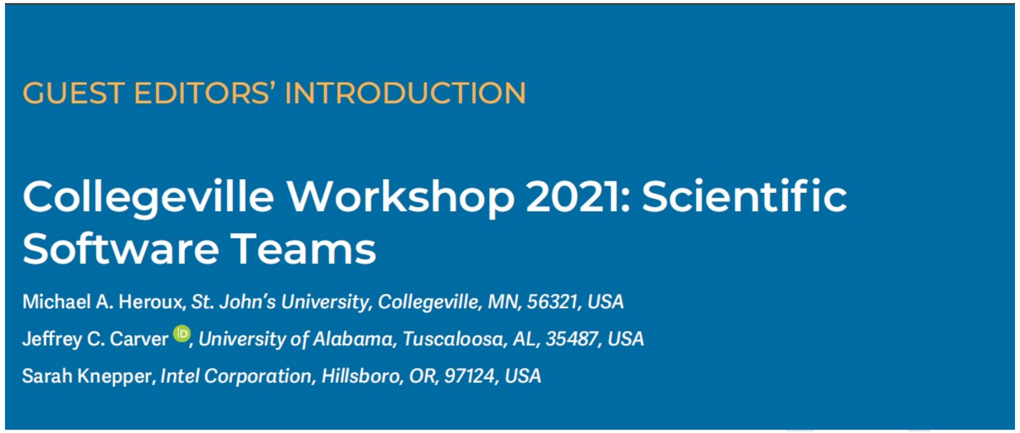
- **PRD1: Develop methodologies and tools to comprehensively improve team-based scientific software development and use**
  - **Key question:** *What practices, processes, and tools can help improve the development, sustainment, evolution, and use of scientific software by teams?*
- **PRD2: Develop next-generation tools to enhance developer productivity and software sustainability**
  - **Key questions:** *How can we create and adapt tools to improve developer effectiveness and efficiency, software sustainability, and support for the continuous evolution of software? How can we support and encourage the adoption of such tools by developers?*
- **PRD3: Develop methodologies, tools, and infrastructure for trustworthy software-intensive science**
  - **Key questions:** *How can we facilitate and encourage effective and efficient reuse of data and software from third parties while ensuring the integrity of our software and the resulting science? How can we provide flexible environments that "bake in" the tracking of software, provenance, and experiment management required to support peer review and reproducibility?*

SSSDU Cross-cutting Themes

- Theme 1: We need to consider both human and technical elements to better understand how to improve the development and use of scientific software.
- Theme 2: We need to address urgent challenges in workforce recruitment and retention in the computing sciences with growth through expanded diversity, stable career paths, and the creation of a community and culture that attract and retain new generations of scientists.
- Theme 3: Scientific software has become essential to all areas of science and technology, creating opportunities for expanded partnerships, collaboration, and impact.

Collegeville Workshops on Scientific Software

Special Issue: IEEE Computing in Science and Engineering, May/June 2022



GUEST EDITORS' INTRODUCTION

Collegeville Workshop 2021: Scientific Software Teams

Michael A. Heroux, *St. John's University, Collegeville, MN, 56321, USA*

Jeffrey C. Carver, *University of Alabama, Tuscaloosa, AL, 35487, USA*

Sarah Knepper, *Intel Corporation, Hillsboro, OR, 97124, USA*

- ***The PETSc Community as Infrastructure***
  Mark Adams, et. al.

- ***Challenges of and Opportunities for a Large Diverse Software Team***
  Cody J. Balos, et. al.

- ***Structured and Unstructured Teams for Research Software Development at the Netherlands eScience Center***
  Carlos Martinez-Ortiz, et. al.

- ***Experiences Integrating Interns into Research Software Teams***
  Jay Lofstead

- ***In Their Shoes: Persona-Based Approaches to Software Quality Practice Incentivization***
  M. R. Mundt, R. M. Milewicz, E. M. Raybourn

- *Three Days:*
  - *Experiences and Challenges*
  - *Technical Approaches for Improvement*
  - *Cultural Approaches for Improvement*

- *Themes:*
  - *2019: Sustainability*
  - *2020: Productivity*
  - *2021: Teams*
  - *2022: Skip due pandemic workforce challenges*
  - *2023: Design*

2023 Collegeville Workshop on Scientific Software

Software Design

July 24 - 27, 2023

https://collegeville.github.io/CW23

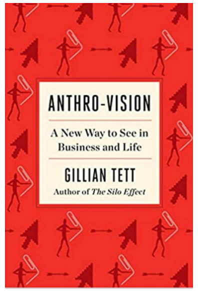
Trends (I see) in Scientific Software that increase value of RSS

- AI-assisted development
  - Elevated thinking – intent to C++
  - Not unlike C++ to machine code
  - Fewer programmers? Maybe
  - Opportunity: More emphasis on purpose & design

tabnine

GitHub Copilot

- Deeper awareness of technology and society
  - Software systems adapted to fit scientists
  - Broaden usability, accessibility, impact



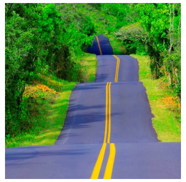
ANTHRO-VISION

A New Way to See in Business and Life

GILLIAN TETT

Author of *The Silo Effect*

- UX applied to scientific software products
  - Personas & journey stories – not new
  - Applied to scientific software teams of developer-users – less common?
  - Just getting started



Summary

- Research Software Science
  - Has been an informal and *ad hoc* set of activities for many years, in my experience
  - Represents a natural next layer of expansion for our increasing diverse research software efforts
- Social and cognitive sciences need a seat the research software table
  - Much to learn from these communities
  - Much to teach members of these communities about our goals, passions, and quirks!
  - Incorporating these sciences can make rigorous what we already do informally
- Lots of opportunities to explore:
  - US DOE: LSSw Town halls, SSSDU workshop follow up
  - AI/ML methods & tools are promising; effectiveness and efficiency require understanding people
  - UX applied to scientific software developer-user teams
- Labeling this kind of work as research software science is still speculative
  - Goal is to see if “there is a there there”
  - If so, let’s figure out what the “there” looks like and how we can proceed

Thank you

https://www.exascaleproject.org

This research was supported by the Exascale Computing Project (17-SC-20-SC), a joint project of the U.S. Department of Energy's Office of Science and National Nuclear Security Administration, responsible for delivering a capable exascale ecosystem, including software, applications, and hardware technology, to support the nation's exascale computing imperative.

ECP Director: Doug Kothe
ECP Deputy Director: Lori Diachin

Thank you to all collaborators in the ECP and broader computational science communities. **The work discussed in this presentation represents creative contributions of many people who are passionately working toward next-generation computational science.**

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Sandia
National
Laboratories