Research Software Science: Expanding the Impact of Research Software Engineering

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
- The Science of Scientific-Software Development and Use (Workshop and Report)
- Trends that increase the value of RSS

Related Paper:
Research Software Science: Expanding the Impact of Research Software Engineering
Submitted to IEEE Computing in Science and Engineering (CiSE)

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?

Applying the Scientific Method to

Use formal observation and experimentation to obtain and disseminate knowledge via repeatable and reproducible processes

Understanding and Improving

Obtain data to detect correlation; design experiments to identify cause and effect; publish results for broad impact

How Software is Developed and Used

Combined focus on developers and users; impact individuals, teams, and communities

For Research

Focus on software used in the pursuit of scientific knowledge and insight

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:

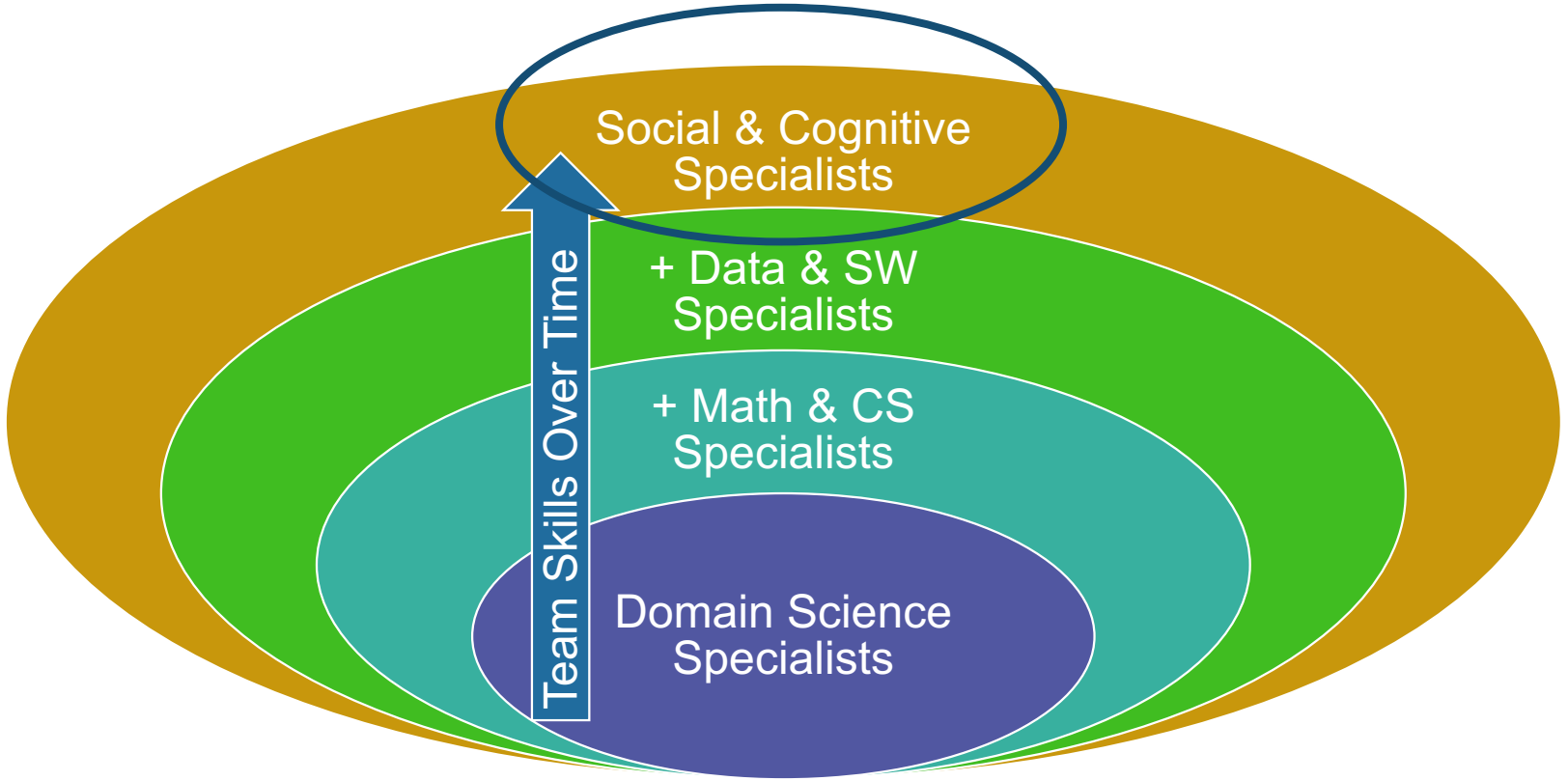
- The role of software has become central to much of our work and the knowledge base is too sophisticated to rely only on software non-experts
- Scientific software success depends on producing high-quality, sustainable software products
- Investing in software as a first-class pursuit improves the whole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

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

SSSDU Priority Research Directions

- **PRD1: Develop next-generation tools to enhance developer productivity and software sustainability** Focus: Developer Impact
  - **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?*
- **PRD2: Develop methodologies and tools to comprehensively improve team-based scientific software development and use** Focus: Team Impact
  - **Key question:** *What practices, processes, and tools can help improve the development, sustainment, evolution, and use of scientific software by teams?*
- **PRD3: Develop methodologies, tools, and infrastructure for trustworthy software-intensive science** Focus: Societal Impact
  - **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.

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

- ***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

Collegeville Workshops on Scientific Software

- *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*

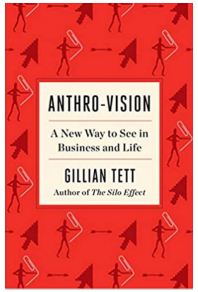
Trends (I see) in Scientific Software that increase value of RSS

- AI-assisted development
  - Elevated thinking – intent to C++
  - Not unlike C++ to machine code
  - Fewer programmers? Maybe
  - Opportunity: More emphasis on purpose & design
  - Link to video using CoPilot to generate Kokkos norm function

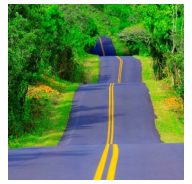
tabnine

GitHub Copilot

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



- 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: 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.