Building Bridges Between the Software Engineering Research (SER) and Research Software Engineering (RSE) Communities

Ian A. Cosden (Princeton University)
Robert Haines (University of Manchester)
David E. Bernholdt (Oak Ridge National Laboratory)

For the organizers and participants in Dagstuhl Seminar 24161

Research Software Engineering: Bridging Knowledge Gaps

- Dagstuhl Seminar 24161
  - Apr 14 - 19, 2024
- Organizers
  - Stephan Druskat (DLR)
  - Lars Grunske (HU Berlin)
  - Caroline Jay (Manchester)
  - Daniel S. Katz (UIUC)
- 40 total participants
  - Research Software Engineers (RSE)
  - Software Engineering Researchers (SER)


Schloss Dagstuhl, the Leibniz Center for Informatics

Seminar Background & Goal

- Hypothetically, the RSEng and the SER communities could benefit from each other
  - RSEs could leverage software engineering research knowledge to adopt state-of-the-art methods and tools
    - Improving RSEng practice towards better research software.
  - Software engineering research could adopt RSEng more comprehensively as a research object
    - Investigate the methods and tools required for the application of state-of-the-art software engineering in research contexts.
- Currently, gaps exist that make it hard for communities to connect
- Goal: Build bridges between the two communities to create mutual benefit through reciprocal collaboration

Shared research questions (examples)

How does RSE differ from SE and what specific activities have to be addressed by different methods and tools?

Can classic SE approaches, like the V-Model, RUP, Scrum, or Xtreme Programming be adapted to RSE needs? How?

Is there a notion of test coverage for research software?

How to manage long-term evolution with fluctuating developer groups?

How to measure the scientific quality of research software?

Can mathematical / physical laws themselves be codified as Domain Specific Language models and thus used for validation, testing or even code generation?

What tools are needed to easily automate various tedious activities that researchers would like to avoid?

How to deal with legacy software or legacy hardware in a long-lasting software infrastructure setting? How to migrate scientific software?

How to predict computational, storage and networking requirements for a software experiment/simulation/study upfront?

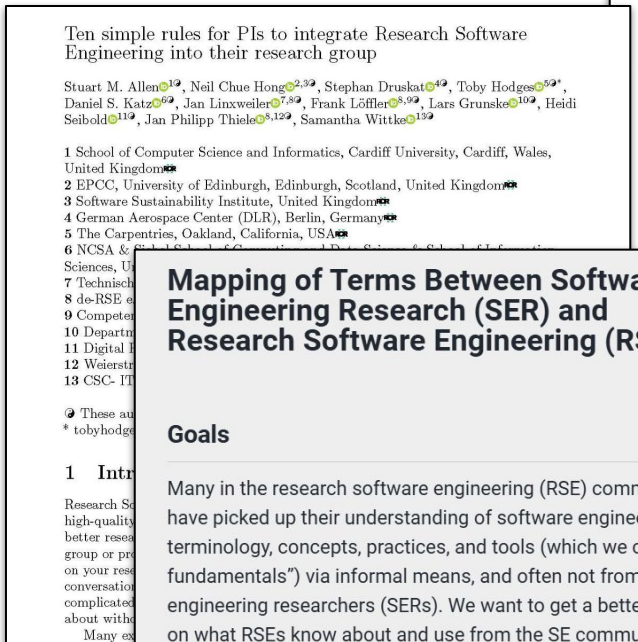
How to predict development efforts and costs of a scientific development process?

What have we done lately?

Outputs from the Seminar have been collected at https://ser-rse-bridge.github.io/

- Organized other events (2)
- Published journal special issues (1)
- Published papers (7) and reports (3)
- Presented posters (3)
  - Including the recipient of the Best Poster award at USRSE'25!
- Created a new website (1)
- Wrote songs (2)
- Created new discussion spaces (3)

and more (we're sure not everything has been reported yet!)


Ten simple rules for PIs to integrate Research Software Engineering into their research group

Stuart M. Allen, Neil Chue Hong, Stephan Druskat, Toby Hodges, Daniel S. Katz, Jan Linxweiler, Frank Löffler, Lars Grunske, Heidi Seibold, Jan Philipp Thiele, Samantha Wittke

1 School of Computer Science and Informatics, Cardiff University, Cardiff, Wales, United Kingdom
2 EPCC, University of Edinburgh, Edinburgh, Scotland, United Kingdom
3 Software Sustainability Institute, United Kingdom
4 German Aerospace Center (DLR), Berlin, Germany
5 The Carpentries, Oakland, California, USA


Mapping of Terms Between Software Engineering Research (SER) and Research Software Engineering (RSE)

Goals

Many in the research software engineering (RSE) community have picked up their understanding of software engineering (SE) terminology, concepts, practices, and tools (which we call "SE fundamentals") via informal means, and often not from software engineering researchers (SERs). We want to get a better handle on what RSEs know about and use from the SE community and

Call to Action

Seek out opportunities to help bridge the gap between software engineering research and software engineering!

And let the community know about it!

https://ser-rse-bridge.github.io

Ten Simple Rules for

Catalyzing Collaborations and Building Bridges between Research Software Engineers (RSEs) and Software Engineering Researchers (SERs)

Which rule resonates most? Which is most challenging to enact?
Vote with a sticker or through our QR linked survey

Plus, the QR code will help you find our paper in *Computing in Science & Engineering!*

Nasir U. Eisty, University of Tennessee, Knoxville, TN, USA
Jeffrey C. Carver, University of Alabama, Tuscaloosa, AL, USA
Johanna Cohoon, Lawrence Berkeley National Laboratory, Berkeley, CA, USA
Ian A. Cosden, Princeton University, Princeton, NJ, USA
Carole Goble, University of Manchester, Manchester, UK
Samuel Grayson, University of Illinois at Urbana-Champaign, Urbana, IL, USA

1 Recognize The Two Communities Are Different

RSEs and SERs must appreciate one another' s unique roles and cultures, celebrate their strengths, and avoid assumptions.

2 Acknowledge Collaboration Is Not Going to Just Happen

Partnerships must be deliberate; proactive initiation, inquiry into mutual goals, and consistent investment are needed.

3 Define Clear Goals and Outcomes

Open discussion, measurable and well documented objectives, and regular check-ins ensure progress, though flexibility is needed too.

4 SERs Must Engage with RSEs in Their Professional Environments

SERs should appreciate RSEs' institutional obligations and resource constraints. SERs can gain these insights and build trust by attending RSE conferences, workshops, and talks.

5 Identify the Intersection of Shared Research Software Challenges

Collaborations must address common challenges that are practically significant and academically valuable. x

6 Ensure Mutual Benefit in Collaboration

Both parties must earn immediate and long-term value. Incentive schemes must be understood. Authorship and leadership responsibilities should be explicitly addressed.

7 Maintain an Open Mind Toward Emerging Challenges

Collaborators should be adaptable and ensure continuous dialogue to identify new challenges and solutions.

8 Actively Advocate for Each Other

RSEs and SERs should showcase one another's work to demonstrate respect and the value of collaboration.

9 Maintain Vigilance and Recognize When Collaborations Are Off Course

To sustain a win-win relationship, SERs and RSEs must regularly review progress and concerns.

10 Secure Institutional Support

All parties should seek funding, advocate for institutional recognition of RSE and SER roles, and promote frameworks that support collaboration (e.g., joint appointments and recognition programs).

Images generated with ChatGPT.