Research Software Engineering in 2030

Daniel S. Katz
NCSA & CS & ECE & iSchool
University of Illinois Urbana Champaign
Urbana, IL, USA
ORCID: 0000-0001-5934-7525

Simon Hettrick
Software Sustainability Institute
University of Southampton
Southampton, UK
ORCID: 0000-0002-6809-5195

Abstract—**This position paper for an invited talk on the "Future of eScience" discusses the Research Software Engineering Movement and where it might be in 2030. Because of the authors' experiences, it is aimed globally but with examples that focus on the United States and United Kingdom.**

Index Terms—**research software, research software engineer, research software engineering**

I. INTRODUCTION

Research software has existed since at least the early 1950s. Formal software engineering started in the 1960s to understand operational software such as operating systems and NASA flights, and later business software, with research software infrequently the focus of study. Bringing these two concepts together and identifying this as a named discipline—research software engineering (RSEng)—with a complementary set of practitioners—Research Software Engineers (RSEs)—was first formalized by a small group in the UK in 2012, as described by Brett et al. in 2017 [1].

Research software is essential to most research [2] and research software engineering (building and maintaining software in a research context using relevant software engineering knowledge) is performed by many people, including RSEs who focus on this work, as well as others in the research community. The RSE Community has grown substantially since 2012. Today (August 2023), there are eight national and multinational RSE associations with established communities and activities, and at least two more developing, all coordinating via an International Council of RSE Associations [3]. The largest, the UK-based Society of Research Software Engineering, has a community of about 4800 people, including about 700 dues-paying members, while the next largest, the US Research Software Association, has about 1900 members. At least 40 UK universities have formal RSE groups, as do many US universities and national laboratories.

In 2021, Cohen et al. [4] defined research software engineering as being built on four pillars: software development, community, training, and policy. In the rest of this position paper, we use these pillars and hypothesize about how the RSE landscape will have changed in 2030, an intervening period

that represents a little over half of the current age of the RSE community.

II. HOW RESEARCH SOFTWARE ENGINEERING MIGHT BE IN 2030

In this section, we predict some qualitative changes to research software engineering that we might expect by 2030. Where it seems reasonable to us, we also make quantitative predictions that we consider plausible, and perhaps even probable.

Software development:
- Research software developers (including but not limited to RSEs) will be generally aware of research software engineering as a concept and its best practices, at about the same level that software developers today are aware of source control packages like git. This will be due to a greater awareness of the role of software in research and changes to community, training, and policy.
- This greater awareness of research software, and the existence and availability of RSEs, will lead to research software being more reliable, maintainable, and better tested. This change to one of the most important underpinning tools of research will make research results more reliable and reproducible.
- More research software will be sustained because funders will better understand that funding the maintenance of research software will improve the return on their initial investment into it. This switch to recognising and supporting successful research software will increase awareness of successful research software and reduce reinvention of existing functionality.
- Research software developers will begin to receive recognition, not just for their contribution to research, but also for their contribution to software within academia. This will allow the design of an incentive system that aligns career progression for research software developers with the practices needed to develop reliable and reproducible research software.

Community:
- At least 50 countries globally will have RSE associations.
- While RSEs will generally be members of their own local RSE organization, there will also be communication channels they use to talk to and collaborate with RSEs in other national organizations.

- There will be a global flagship RSE conference that moves to a different country each year and that also supports virtual participation.
- There will be at least one research software engineering journal.
- There will be at least 50 professors whose focus is on research software engineering.
- Most professional scholarly societies will have research software tracks or themes in their conferences, and will offer prizes to research software developers.
- Awareness of the RSE career will be significantly more widespread, which will lead to around a quarter of new recruits being recruited after their undergraduate degree.

Training:
- At least one university Masters of Research Software Engineering (MRSE) program will exist, and many universities will offer RSE certificates or minors at the undergraduate level.
- Some secondary students will be aware of research software, and will consider an RSE career when entering university.
- Both Software Carpentry and a merged program beyond it, similar to programs today such as INTERSECT [5] and CodeRefinery [6], will offer frequent synchronous and asynchronous research software engineering training.
- As a part of more standardized RSE career paths, skills required at various levels will be specified and training for RSEs new to these levels will be available both synchronously and asynchronously.
- Training for the initial RSE level will be offered in internship programs.
- Courses in software engineering techniques will be either mandatory or strongly encouraged for all new doctoral students.

Policy:
- The value of software, and its vital role in the production of research results, will be accepted across all levels in research-performing organizations and by policymakers across government and funders.
- In the US, 75% of R1 universities will have RSE organizations. In the UK, all research-led universities will host an RSE Group.
- Funding organizations will have specific programs to fund 1) the development of new sustainable research software, 2) the maintenance and sustaining of existing research software, and 3) the transition of software developed in research activities into sustainable research software.
- Most US universities will have RSE career paths that contain multiple levels, allow for both technical and management career progression over 30+ years, and allow RSEs to be PIs.

- RSE career paths will be mostly standardized, and it will be as easy to move from one organization to another at a given RSE level as it as today at a given faculty level.
- Institutions with RSE groups will also have Open Source Program Offices and vice-versa, and they will work together to promote and strengthen research software at their institutions directly and via policy changes.
- Research-performing organizations (e.g., universities, national laboratories) will recognize and reward research software work in all research-performing positions, as will national research assessment activities.

III. The value of predictions

While the predictions we make in this position paper may not come to pass, as we've said, we think they are plausible. In some cases, they simply extend current trends, while in others, they require new collaborative work to accomplish. We urge community members to consider how they might make these predictions more like to be fulfilled, working within existing RSE communities and with organizations such as the Software Sustainability Institute [7], the US Research Software Sustainability Institute [8], and the Research Software Alliance [9].

Acknowledgment

References

[1] A. Brett, M. Croucher, R. Haines, S. Hettrick, J. Hetherington, M. Stillwell, and C. Wyatt, "Research Software Engineers: State of the Nation Report 2017," Apr. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.495360

[2] S. Hettrick, "It's impossible to conduct research without software, say 7 out of 10 UK researchers," https://www.software.ac.uk/blog/2016-09-12-its-impossible-conduct-research-without-software-say-7-out-10-uk-researchers, 2014.

[3] "International council of RSE associations," https://researchsoftware.org/council.html.

[4] J. Cohen, D. S. Katz, M. Barker, N. Chue Hong, R. Haines, and C. Jay, "The four pillars of research software engineering," *IEEE Software*, vol. 38, no. 1, pp. 97–105, 2021. [Online]. Available: https://doi.org/10.1109/MS.2020.2973362

[5] https://intersect-training.org.

[6] https://coderefinery.org.

[7] S. Crouch, N. C. Hong, S. Hettrick, M. Jackson, A. Pawlik, S. Sufi, L. Carr, D. De Roure, C. Goble, and M. Parsons, "The Software Sustainability Institute: Changing research software attitudes and practices," *Computing in Science & Engineering*, vol. 15, no. 6, p. 74–80, Nov. 2013. [Online]. Available: https://doi.org/10.1109/MCSE.2013.133

[8] J. C. Carver, S. Gesing, D. S. Katz, K. Ram, and N. Weber, "Conceptualization of a US research software sustainability institute (URSSI)," *Computing in Science & Engineering*, vol. 20, no. 3, p. 4–9, May 2018. [Online]. Available: https://doi.org/10.1109/MCSE.2018.03221924

[9] D. S. Katz and M. Barker, "The Research Software Alliance (ReSA)," Apr. 2023. [Online]. Available: https://doi.org/10.54900/zwm7q-vet94