2nd International Research Software Funders Workshop Summary

December 20, 2023

Contributors

Rena Bakhshi: 0000-0002-2932-3028
Michelle Barker: 0000-0002-3623-172X
Ishwar Chandramouliswaran: 0000-0001-9697-9599
Neil Chue Hong: 0000-0002-8876-7606
Carole Goble: 0000-0003-1219-2137
Brian Corrie: 0000-0003-3888-6495
Fares Dhane: 0000-0002-3559-8239
Christopher Erdmann: 0000-0003-2554-180X
Richard Gunn: 0009-0005-7743-2164
Kim Hartley: 0000-0002-4345-9044
Daniel S. Katz: 0000-0001-5934-7525
Fabio Kon: 0000-0003-3888-7340
Mark Leggott: 0000-0003-1392-7799
Richard Littauer: 0000-0001-5428-7535
JB Poline: 0000-0002-9794-749X
Malvika Sharan: 0000-0001-6619-7369
Andrew Treloar: 0000-0002-8911-3081
Qian Zhang: 0000-0003-1549-7358
Shurui Zhou: 0000-0002-6346-6073

1. Introduction to the 2nd International Research Software Funders Workshop

Background, theme, participants

This report summarizes the 2nd International Research Software Funders Workshop, including its discussions and suggestions. From September 18-20, 2023, the Digital Research Alliance of Canada (the Alliance) and the Research Software Alliance (ReSA) co-convened this hybrid workshop, which brought together 50 representatives from more than 35 organizations from across the globe – including funders and organizations that support research software – to collaborate in Montreal, Canada, and online.

People are at the heart of research software sustainability and funders can lead the way by investing in the people who develop and maintain research software. That is why the workshop theme was "Investing in People: Anticipating the Future of Research Software". The event aimed to set the future agenda for government, philanthropic, and industry funders to support sustainable research software and communities.

During the three-day workshop, experts explored how research software funders can facilitate capacity building through funder practices; research software platforms, infrastructure, and communities; and new initiatives. Moreover, Version 1.0 of the Amsterdam Declaration on Funding Research Software Sustainability (ADORE.software) was released and is now available for signing. The recommendations in ADORE.software informed the structure of the workshop's breakout sessions, with a focus on research software practice; research software ecosystem; research software personnel; and research software ethics.

Background

This second iteration of the funders workshop built on the inaugural International Funders Workshop: The Future of Research Software, co-hosted by the Netherlands eScience Center and ReSA in November 2022 in Amsterdam. During this first workshop, funders gathered to set the agenda for supporting sustainable research software and focused on the drafting of ADORE.software. The first workshop marked a further step in the development of the research software community. Stakeholders have continued to build on this momentum through involvement in the Research Software Funders Forum, convened by ReSA, and its funder-led working groups. The Alliance and ReSA, together with the workshop Steering Committee, organized the 2nd International Research Software Funders Workshop to continue this important work and facilitate global collaboration among funders and other key decision makers and influencers across the research software ecosystem.

Participants

Of the 35 organizations that participated in the workshop, approximately 57% were funders and 43% were key stakeholders in the research software community. Of the 50 total participants, 30 participated in person and 20 participated online. Participants included representatives from the international research software funder community (government, philanthropic, and industry organizations); academics; institutional leaders; non-profit organizations; and independent researchers.

2. Workshop content

Pre-workshop – September 18

How to Explicitly Support Research Software as Part of Your Funding Program

The event commenced with a pre-workshop, including a half-day session on How to Explicitly Support Research Software as Part of Your Funding Program, presented by ReSA. As a result of discussions within the Research Software Funders Forum and engagement with the global research software community, ReSA identified a need to introduce funders to research software (or to help them understand that they already fund research software). This pre-workshop aimed to address the following: why it is important for funders to recognize and support research software; ways in which funders support research software, both through explicit programs and through research funding where research software support may be hidden; how to support research software through both specific funding and other types of funding programs; and how to make the case internally for increased support for research software.

During the workshop, participants reflected on their engagement with research software and considered sources of data that could be useful, such as research software citations and mentions; a database of groups working on collaborative research software to reduce duplicative work and better allocate funds; software registries' usage information; grant proposals and demographic information; funders' tracking actual dollars invested and co-invested; re-use of components or frameworks in which funders have invested; the Research Software Directory (RSD), a free open-source platform, that facilitates "F" and "A" parts of the FAIR principles; and existing work (e.g., Chan Zuckerberg Initiative's dataset of software mentions, Dataseer, Data Citation Corpus, GROBID; and OpenAIRE and DataCite who are investing in mention/citation monitoring and building public knowledge graphs).

Brian Corrie, Technical Director, iReceptor, Simon Fraser University, and Consultant, Digital Research Alliance of Canada, Research Software Strategy Working Group, gave a lightning talk on The road to a national strategy to fund RS in Canada. And James Ricci, Director, Schmidt Futures, presented on Schmidt Futures' Virtual Institute for Scientific Software (VISS).

There was a discussion about ways funders can support research software through specific funding and in other calls. The following resources were shared: ADORE.software and the ADORE.software toolkit; Ten simple rules for funding scientific open source software (Strasser et al., 2022); and Policy recommendations to ensure that research software is openly accessible and reusable (McKiernan et al., 2023). Simple ways in which funders can support research software in funding calls include the use of Software Management Plans (or inclusion in Data Management Plans) – see Dutch Practical Guide to Software Management Plans; enabling Research Software Engineers to be Principal Investigators; and specifying policy on sharing using Open Research Funders Group (ORFG) Policy Clause Bank policy levels.

Some other ideas shared by participants:

- Recognize that operational funding is often needed to support research software after it has been developed.
- Share impact generated as a result of the investment (e.g., Australian Research Data Commons (ARDC) Impact Case Studies). This increases the awareness of software, and potentially leads to further investment from others.
- Invest in software re-engineering to support wider development.
- Provide funds to support RSEs or teams of RSEs themselves to work across projects, not just supporting them through soft money for individual projects.
- Provide a separate “follow-on” funding for “impact acceleration” of existing research grants that have produced software as an output (e.g. for additional maintenance, publicity and outreach).
- Require grant proposals to list the software they are using/reusing as part of the software management plan/data management plan. This helps identify important parts of the software infrastructure and encourage applicants to think more carefully about how they use software.
- Require adoption of platforms such as RSD to collect data on projects’ outputs, including software.

Funders who are interested in learning more about supporting research software are invited to join the (free) Research Software Funders Forum by contacting ReSA at info@researchsoft.org. The Funders Forum offers a formal mechanism for funders to share practices and consider how to address common challenges to achieve the significant cultural change needed across the research sector globally. For further information, please refer to this Introduction to Research Software and the ADORE.software Toolkit.

CiteSoftware

The Cite.Software initiative aims to simplify citing software for researchers through unified community guidance and automated community accepted solutions. It is a collaborative community effort between multiple organizations including SciLifeLab, The Turing Way, Open Life Science, The Netherlands eScience Center, the Research Software Directory, Public Library of Science, Dutch Research Council (NWO), Digital Research Alliance of Canada, and Crossref. The initial concept was presented during this workshop in Montreal. Based on the guidance of attendees, the initial work of the group will focus on creating a

ReSA working group (and potentially an RDA Tiger Group) that will bring stakeholders together to help develop common guidance (website) while leveraging modular content from The Turing Way. The second phase of the effort will focus on leveraging the Research Software Directory to automate aspects of the citation guidance, and potentially, software management planning. Currently, the group is working on a proposal to fund a diverse team from The Turing Way and OLS to guide the initial work. CiteSoftware is looking for support.

To access CiteSoftware presentations from the Research Software Funders Forum, see below:

Erdmann, C., Sharan, M., de Jong, M., Maassen, J., Cadwallader, L., Dhane, F., Cruz, M., Martinez, C., Leggott, M., & Lammey, R. (2023, September 18). Cite.Software. Zenodo. https://doi.org/10.5281/zenodo.8356278

Maaike de Jong, & Jason Maassen. (2023, September 18). The Research Software Directory - a brief introduction for Cite.Software. Zenodo. https://doi.org/10.5281/zenodo.8359462

Sharan, M. (2023, September 18). The Turing Way: A Digital Commons for Open Science and Reproducibility. Zenodo. https://doi.org/10.5281/zenodo.8353824

Cadwallader, L. (2023, October 19). Open software and code at PLOS - Cite.Software presentation. Research Software Funders Workshop, Palais des congrès de Montréal, Montréal, Canada, and virtual. Zenodo. https://doi.org/10.5281/zenodo.10022544

Martinez-Ortiz, C., Martinez Lavanchy, P., Sesink, L., Olivier, B. G., Meakin, J., de Jong, M., & Cruz, M. (2023). Practical guide to Software Management Plans (1.1). Zenodo. https://doi.org/10.5281/zenodo.7589725

Workshop Day 1 – September 19

The first day of the workshop included presentations that set the scene for participants to explore the crucial role funders play in supporting sustainable research software and communities.

Welcome remarks

George Ross, CEO of the Alliance, and Michelle Barker, Director of ReSA, welcomed participants to the second International Research Software Funders Workshop.

George provided a brief overview of the Alliance and its work. The Alliance was formed in 2019 after several years of work by the Canadian research community. The establishment of the Alliance has been very important for the country. The Alliance provides critical infrastructure for the research endeavor in Canada, and is a consolidation of three main pillars: advanced research computing, research data management, and research software.

Since its inception, the Alliance has focused on making significant investments. While research software is the Alliance's least mature pillar, the organization is committed to and has done a number of key activities recently to highlight its importance, including the completion of a national research software strategy and the upcoming release of a research software strategy document. Moreover, George noted the importance of international connections. For Canada, leveraging international relationships, especially with regard to research software, is important.

Michelle introduced ReSA, including its vision that research software and those who develop and maintain it are recognized and valued as fundamental and vital to research worldwide and its mission to advance the research software ecosystem by collaborating with decision makers and key influencers. ReSA formed in recognition that no international organization for research software as a whole existed at that time. While there were many community, national, disciplinary, and topic-focused initiatives happening worldwide, there was no way in which they came together formally to think about how to solve problems collectively. ReSA addresses this gap – it aims to catalyze and facilitate the community to jointly identify and develop solutions to both local and international challenges. Michelle shared a number of ways to engage with ReSA. One of ReSA's primary engagement activities involves dispersing information to the community (e.g., via newsletters and social media). ReSA also encourages the community to form working groups that sit under ReSA auspices. These working groups aim to solve problems and create international solutions from which everyone can benefit. For example, the ReSA task force on FAIR Principles for Research Software (FAIR4RS) – convened in partnership with RDA and FORCE11 – introduced the FAIR4RS Principles in this *Scientific Data* article, and a range of organizations are now adopting them. ReSA also offers opportunities for leadership as part of the ReSA Steering Committee and Organizational Membership program.

What is research software and why is it critical to the research endeavor

This opening talk [1] by Professor Carole Goble set the scene for the workshop by reminding the audience why software is critical to research and introducing six different perspectives on what is research software.

The majority of researchers acknowledge that scientific software is important for their own research and would not be able to do their research without software; moreover, a large number develop software as part of their research. Software is critical to every step of the research data lifecycle – plan, collect, process, analyze, preserve, share, reuse – and drives/enables modern modes of research activity such as team science, and both data driven and hypothesis driven science. Software is essential for the pooling of the curated knowledge of research outputs and the generation of knowledge through simulations and predictions. From an external stakeholders perspective, going into the research endeavour, funders have increasingly invested in research software, and coming out publishers such as PLoS report that 40-50% articles generate code and the most-cited papers are methods and software related [2].

The definition of Research Software varies depending on different perspectives, and Carole offered five, all of which can apply at the same time, have fuzzy edges, and vary in scope and weight between disciplines:

1. *An intent perspective* – the intention of the software at the outset, differentiating between Research Software created during the research process or for a research purpose and Software in Research used for research but not created during or with research intent [3]
2. *A role perspective* – what role the software is playing: as an instrument, infrastructure, results processor, or facilitator of collaboration [4]
3. *An owner perspective* – comparing software that is collectively owned and developed with good practices against legacy codes handed down from researcher to researcher, recognizing that it is easier and more rewarding to create one's own new software than to extend existing software, and re-invent small programs instead of collaborate to sustain and improve big ones [5]
4. *A maturity perspective* – three maturity stages of software: (i) Analysis Code (one-off "me" research where software is a side effect intended for disposal); (ii) Prototype Tools (research need, "professorware" which may or may not be reused) and Research Software Infrastructure (professionalized product intended for reuse). Software may transition between these states, and different developer types (researchers, researchers who code, researcher developers, research software engineers) contribute. [6][7]
5. *A stack perspective* – illustrated by the infamous xkcd comic whereby different software sit in different places in the stack from the user viewpoint (applications, tools, scripts etc) that are discipline- or project-specific and immediately visible, through to underpinning middleware (scientific and non-scientific infrastructure platforms, libraries, utilities, frameworks, etc.) that are often cross-domain generic reusable but overly familiar, invisible, and often neglected. The stack also highlights the dependencies between software and the fragility of those; software will break if not maintained and that, in turn, breaks other software.

The talk concluded with the observation that it is people that make, use, and resource software, and it is people who are at the heart of software sustainability. Software sustainability should be considered from a PESTLE analysis perspective (political, economic, social, technological, environmental, and legal).

[1] Carole Goble What is research software and why is it critical to the research endeavour, 2nd International Funders Workshop: The Future of Research Software 18-20 Sept 2023, Montreal, Canada (hybrid), 2023, https://zenodo.org/records/10138709
[2] PLoS Open Science Indicators dataset version 3 2023, https://plos.figshare.com/articles/dataset/PLOS_Open_Science_Indicators/21687686/3?file=41362998
[3] Gruenpeter et al., "Defining Research Software: a controversial discussion," 2021 https://doi.org/10.5281/zenodo.5504016
[4] Rob van Nieuwpoort and Dan Katz, "Defining the roles of research software," 1st Funders Workshop, 2022, https://doi.org/10.54900/9akm9y5-5ject5y

[5] Warrick Ball “How can RSEs stop researchers from re-inventing small programs instead of collaborating to sustain and improve big ones?”, RSECon 2023, https://virtual.oxfordabstracts.com/#/event/4430/submission/31
[6] Australian Research Data Commons. (2022). A National Agenda for Research Software (1.0). Zenodo. https://doi.org/10.5281/zenodo.6378082
[7] Dan Katz “Incentives and Frictions in Community Software Projects”, 2022 https://doi.org/10.5281/zenodo.6677821

Keynote by Josh Greenberg

Josh Greenberg, Technology Program Director at the Alfred P. Sloan Foundation, delivered a keynote that provided a funder’s perspective. He shared several pertinent questions for those who think about allocating resources for research:

(1) **How much (and when) should we worry about technical debt?** When writing software, sometimes developers take shortcuts. The choice to take the “good enough for now” over “perfect” results in technical debt. Like other forms of debt, it can accrue interest. Funders can split resources across more labs and projects by underfunding robust engineering practices that would keep technical debt lower. But in doing so, they may have to pay more in the future to enable greater use and impact. Neil Ernst’s book *Technical Debt in Practice: How to Find it and Fix it* explores technical debt around scientific software.

(2) **How much attention should we pay to user interface design in research software?** It is rare to find a UI/UX professional among any core set of contributors in research software. UX is generally overlooked in research software because a lot of research software is built by its primary users, colleagues, or students – and it may not seem worth the effort to address idiosyncrasies or minor frustrations. However, more investment in design and user experience of research software earlier in its lifecycle could contribute to earlier adoption, lower barriers to reuse, healthy diminishing maintenance costs, and healthy cycles of community user feedback that could inform how to build more relevant software.

(3) **Who should do which kinds of software?** Within the research software community, there has been much discussion about how to assign scholarly credit for software work. There is a lack of clear norms about the different roles in scientific software but also norms about what kind of equity contributors should accrue as they add value to a code base. While software development can be an important part of students’ and postdocs’ training, relying on them adds a lot of churn. Hiring, Managing, and Retaining Data Scientists and Research Software Engineers in Academia is a recent publication from the Academic Data Science Alliance (ADSA) and US RSE that presents challenges and suggestions for solutions to improve the recognition of RSEs and advance their career paths.

(4) **How central should “open source” be?** Engaging with open source can vary widely when looking at the research software funding landscape. Having an open-

networked resource and social version control, mixed with clear governance about who gets to make what type of decisions about individual changes in project direction, can lead to effective, distributed collaboration. Contemporary open source software is a mature example of this. Those interested in exploring this topic further should consider joining the Sustain OSS Academic Working Group, led by Richard Littauer, which is bringing together Open Source Program Offices (OSPOs), labs, and individuals thinking about software in the research context.

(5) **How can we help institutions build capacity for research software development (and maintenance)?** Technology advances are more likely to benefit science if universities and research institutions have the capacity to introduce, support, and maintain them for their labs and teams. Some institutions that have reached a critical mass of RSEs have sustainability and a competitive advantage. Early programs, such as CANARIE's local software support and the Schmidt Futures Virtual Institute for Scientific Software (VISS) program that directly funds teams of RSEs in a capacity-building mode, are good examples. There is also a lot of exciting activity around OSPOs. For example, OSPO++, led by Jacob Green, connects government, academic, and civil society OSPOs. Through Sloan's funding of OSPOs, it has become clear that OSPO staff at these institutions are not only increasing adoption of version control, but also encouraging better project governance, and collecting metrics to validate open source work for hiring, tenure, and promotion.

Finally, what about AI? Josh noted that his talk takes for granted that human beings are developing and using research software. There could potentially be a transformation in the role of automation in software production, maintenance, and use.

Breakout Session #1 – Capacity Building through Funder Practices

1. Research Software Practice - The importance of funding re-use and maintenance of research software, and how to create funding programs that do this. Facilitator: Maria Cruz.

This session was related to the Amsterdam Declaration on Funding Research Software Sustainability (ADORE.software) recommendation that "Funders should incentivize the reuse and improvement of existing research software". There was a lively discussion, both in person and online. The big themes that came out of the discussion were these:

- The need to invest both before (community/need development) and after (ongoing maintenance and re-use) project funding.
- There are dangers of thinking about funding research software through the lens we use to think about funding research (an emphasis on PIs not teams, novel not boring, one-off not re-use).
- There is a challenge in how to balance investing in existing software/ideas rather in new software/ideas.
- There are specific challenges for new funders in working out what to prioritize/what is the best funding model.

- Funders need to balance flexibility with compliance requirements (and this will get worked out in different ways depending on individual circumstance).
- There is a surprising diversity in funding approaches. The group discussed whether this was a positive or negative, and could not decide. Regardless, how could we draw on this diversity to provide more common opportunities for research funding?

Next steps to continue the conversation:

- Explore joining a proposed ReSA Working Group (WG) on multilateral funding calls (see this presentation by Fabio Kon for more details).
- Capture lessons learned from existing funding calls (like the Ten Simple Rules for funding scientific open source paper in PLOS).
- Funders could collaborate to develop generic guidance for research software developers, such as best practices they should follow.
- Research Software Funders Forum - Introduction to Research Software
- Develop a matchmaking program between funders to align understandings/incentives.
- Funders and developers should ensure that active funding calls are listed in ReSA's Research Software Funding Opportunities database.

2. Research Software Ecosystem - Building a multilateral funding initiative. Facilitator: Mark Leggott.

This session was related to the ADORE.software recommendation that "Funders (including public, private, and philanthropic) should be aware of each others' investments and work in a coordinated manner, as the research software ecosystem exceeds institutional and national boundaries".

Fabio Kon from FAPESP provided an overview of the Research Software Funders Forum WG on a multilateral funding initiative. Fabio noted that several agencies are interested in a potential cross-agency or cross-country call. Although the working group has drafted a call, getting commitments from organizations has been a challenge. There are other joint calls that already exist and these could potentially include research software (e.g., Belmont Forum, CHIST-ERA, Trans-Atlantic Platform - Digging into Data, NGIAtlantic.eu).

Session participants explored the ways in which the Funders Forum WG could proceed. The group could partner with an existing organization or move ahead with a new approach for the call the WG created. Moreover, the breakout session group discussed possible ideas for a specific focus of the call/challenge. For example, a training-based theme or shifting the focus to the "middle layer" pieces that are critical components being used in research (e.g., Python libraries, DOI minting libraries/services, JupyterHub/2i2c, OpenRefine).

To continue the conversation, the group suggested organizing a Pooled Funds Projects meeting to learn from existing calls and organizations.

Resources:
Home: It Takes a Village - It Takes A Village (lyrasis.org)

3. Research Software Personnel - How to support and sustain Research Software Engineers. Facilitator: Qian Zhang

This breakout session was related to the ADORE.software recommendation that “Funders should stimulate the training, hiring, and funding of both professional research and technical staff able to reuse, develop, and maintain sustainable research software.” Participants who represented both research funders and research performing organizations shared their experience and challenges they have encountered.

The unique characteristics of RSEs:

- Driver to RSEs is often not the salary, as they choose research over well-paid industry roles because they want to do cutting-edge research with social impact.
- RSE roles vary in the evolution chain of software and professionalization of software, and may or may not in academic settings. RSEs provide teaching, accelerate research, and make PIs (and their team/organization) more competitive in the award market. RSEs transfer knowledge across disciplines, and build capability and capacity in the university.
- Although RSEs’ work does not necessarily need a full-time person, they need a steady state and a career development path.
- RSEs are a type of contributor to a project. Professional positioning is building the RSE, not the position in the project.

Good practices and consensus for funding and supporting RSEs:

- Building on payment of appropriate market prices is important.
- Creating a centralized RSE pool plus educating PIs and administrators/HR in universities regarding the RSE significance are critical for RSE retention.
- The UK has different levels of granularity of RSE groups that most other countries do not.
- UKRI RSE fellowships encourage tech professionals to lead initiatives.
- Code for Science and Society (CS&S) has one initiative that supports RSEs outside university spaces through a cohort model, to build bonds within the cohort.

Conversations to be continued:

- Where should funders build the RSE capacity? For example, the Netherlands eScience Center (NLeSC) is a national-level instance. However, universities sometimes have difficulty subcontracting.
- Distinction for funders: How to incentivize researcher-developers to develop better software engineering practices versus funding RSE professionals?
- How to de-risk inequitable funding of the groups?
- OSPOs and RSEs: Patterns for centralizing engineering effort into an OSPO exist in the private sector, and tend to fall into one of two categories: upstream contributions to projects that the company commercializes (cf. VMWare), and upstream contributions to projects on which the company depends (cf. G-Research). Some of it comes down to what role you want to play in spreading the culture. Centralizing the effort can realize some efficiencies but create siloed culture, decentralizing will spread the culture but can require having to explain the work more.

Possible next steps:

- To invite RSE Directors in organizations (e.g., SocRSE and regional RSE associations) at the table for future discussions.
- To do some case studies of centralized RSE pools to understand how to best support RSEs.

Resources

- 2019 – Research Software Development & Management in Universities: Case Studies from Manchester's RSDS Group, Illinois' NCSA, and Notre Dame's CRC https://arxiv.org/abs/1903.00732
- Katz, Daniel S. and Haines, Robert and McHenry, Kenton and Reinking, Caleb and Jones, Catherine and Haupt, Carina (2019) *Developing and Managing Research Software in Universities and National Labs.* SC19, 17.-22. Nov. 2019, Denver, US. (event page; talks and audience poll/responses)
- The Changing Role of RSEs over the Lifetime of Parsl https://www.researchgate.net/publication/372487522_The_Changing_Role_of_RSEs_over_the_Lifetime_of_Parsl, https://arxiv.org/abs/2307.11060

4. Research Software Ethics - How do you measure research software impact? Facilitator: Colette Bos.

This session related to the ADORE.software recommendation that “Funders should encourage the responsible use of appropriate indicators to assess the degree of permanence, reusability, and impact of research software”. The group noted that there is a wealth of data on the repository level; however, how to assess impact is unclear. Some stakeholders count a variety of measures, such as software citations, mentions, and downloads. Participants considered how to map these measures to metrics that assess impact. They also discussed what metrics they would like to use. Having proper project information about where the software is used would be beneficial (e.g., software used in teaching, software used in soft skills, software that has a social impact outside of research).

Participants also discussed the idea of evaluating the narratives or stories of research software. Doing an analysis on software mentions is one consideration; however, it is difficult to gather data for this. They noted that not all citations/mentions are equal, and that there are many levels of impact. For example, citations might include a platform as a list of possible sources for data; a platform where the researchers got the data that was critical to their results; or a software that was essential to their research. Can impact be inferred from the text of citations? Suggestions include combining mention analysis and dependency analysis; having a centralized repository for such reports; and having a set of high-level classes of impact (e.g., science impact, societal impact, community impact, policy impact).

Resources: How do software management plans advance open science and contribute to research quality? by Carlos Martinez Ortiz, Maria Cruz, and Maaike de Jong.

Keynote by Malvika Sharan

Malvika Sharan from The Alan Turing Institute and Open Life Science provided a research software community perspective in her talk on Exploring 'Do No Harm' Principles in Open Research Communities (see video). Malvika's compelling keynote encouraged funders to become leaders in the research software ecosystem by incentivizing and recognizing all types of research roles; investing in capacity-building goals that focus on bridging scientific and economic divides; and applying community-oriented frameworks to extend the benefits of research to the broader community that 'do no harm', thereby ensuring open science practices enable equitable research and research outcomes.

Malvika discussed the shared responsibility of all who engage critically in research infrastructure, so we can understand what participation, collective action, and progress means for the global research community. The goal is to make our collaborative efforts more effective, accountable, and efficient while reducing harm towards different stakeholders and beneficiaries in research. The economic and knowledge divide are interlinked. A lack of investment in research infrastructure in low-to-middle income countries (LMIC) and a significantly lower number of researchers in secure positions in the Global South has continued to widen this gap.

Both reproducibility and openness reinforce each other in creating research that enhances transparency, equity, and inclusion at all stages. Open source software, open data, open education, citizen science, open access, and so on, enhance diversity of knowledge, leading to greater economic justice. While goals of openness may differ, there is a common mission to produce public good, encourage collaboration, and broaden the diversity of knowledge-producing actors. Malvika noted that barriers are different for Global South researchers, including language and culture barriers; a lack of investment in research infrastructure; and imbalanced research collaborations. These barriers have resulted in unintentional exclusion from participation in knowledge production. Even though open science facilitates collaboration between Global South and Global North stakeholders, authorship may not be sufficient for LMICs to gain the economic benefit that they need, and local communities may not benefit from the research.

In a research software context, Malvika highlighted that it is not enough to just include people – the key is to ensure diverse stakeholders are contributors to the decision-making process in a way that benefits them. Moreover, when considering research infrastructure roles, providing a sustainability pathway for the open source ecosystem – the software itself, people involved, production and maintenance work, funders' investment, and community of users, as well as the context where these things are happening – is an important aim.

The "Do No Harm" principle and framework helps organizations to become more effective, accountable, and efficient by considering the context, responding to unintentional negative impacts of their work, and amplifying voices from local communities. The "Do No Harm" (Simple) framework includes the following elements: rights of beneficiaries (i.e., all actors of research and research software communities have the right to science); functioning of communities and relationship between actors; local economy; and environment and

contexts. It is important to consider three areas used to address or minimize harm of any intervention, including implementation approaches, infrastructure support, and behaviour/culture.

To commit to ensuring rights of beneficiaries, stakeholders need to take into account the emerging research landscape and commit to openness and reproducibility; inclusive project design; transparent communication; collaborative culture; research ethics; and community building (EDIA). Recognizing and funding both technical and social infrastructure in research is critical. For example, funding and supporting diversifying research roles. In considering the functioning of communities and stakeholder relations, governance practices to reach the community's goals, assign responsibility, establish a system, and assess outcomes of collective action is needed. The Turing Way is a community-led handbook to best practices in Data Science. Turing involves and supports a diverse community to make research reproducible, ethical, and collaborative for everyone.

In conclusion, community and capacity building should focus on bridging the scientific and economic divide by sharing benefits. A community-oriented framework can extend the benefits of research to the broader community, where funders can lead the way by setting the right incentives in place that *do no harm*.

Breakout Session #2 - Capacity Building through Research Software Platforms, Infrastructure, and Communities

1. Research Software Practice - Benefits of Open Source Program Offices (OSPOs). Facilitator: Richard Littauer.

As a starting point for discussion, this session was initially related to the ADORE.software recommendation that "Funders should stimulate the documentation, licensing, open-source distribution and accessibility of research software to enable the reproducibility of research outcomes". There are at least two other ADORE recommendations that were relevant: that "funders should stimulate the training, hiring, and funding of both professional research and technical staff able to reuse, develop, and maintain sustainable research software" and that "funders should stimulate the development and maintenance of a research software ecosystem, including people, communities, and infrastructure, to ensure research software sustainability."

Open Source Program (or Programme) Offices (OSPOs) are increasingly being seen as a potential tool for meeting these recommendations. In short, they are centres of excellence in a university setting – a single person or a group of people who are nominated by the university as the experts on open source, who can help set guidelines for use of open source technology, assist with brokering conversations between academics and administrations, foster collaborations across universities and research institutions, and so on.

Open source software and research software are not identical; while much research software is licensed under open source license, or built using open work paradigms, there is

some that is not. OSPOs, as well, have a broader remit than research software; for instance, they may advise on teaching students open source best practices in courses, or they may be involved with working with the technology transfer office of a university. However, the overlapping Venn diagram of OSPOs and research software leaves a large enough space that they may be considered vehicles for helping advance research software and the ADORE recommendations.

Several organizations that work with OSPOs were represented by the members of the breakout – Richard Littauer, from the SustainOSS Academic Working Group, which focuses on systemic risks to open source in academia; Jacob Green, of OSPO++, an organization that seeks to foster collaboration between OSPOs in all areas, including industry and government; Michelle Barker, who works with ReSA; and Duane O'Brien, who has worked extensively with the TODO Group, which also has an OSPO working group. The conversation started by talking about the similarities and scopes of these groups. Other groups were mentioned, like CHAOSS, which works to provide metrics for communities, and which also has an academic group, where metrics are often made with the goal of being used by and for the OSPO at a given university. The European OSPO Alliance, out of Eclipse, and the Open Source Observatory were also mentioned as similar groups.

While there are a plethora of groups working together on OSPOs, there is still a lot to be learned and developed about how OSPOs work on the ground and how they can best collaborate and support each other. More resources exist for OSPOs in industry – however, these are significantly different from academic ones, as they have different incentives and power structures. Commercial OSPOs often run into trouble as they are started as a “catch-all” for any open source issues, but then are tasked with solving specific problems, and often are not placed in the organization to execute on those demands. This leads to their susceptibility to budget cuts. In academia, the placement of an OSPO is equally important, but the incentives and runway may differ. With regards to research software, the OSPO needs to have a charter that aligns with supporting it, as well as the authority to execute on that charter. As the concept is still relatively new, it's unclear how this can best be achieved generally across different settings and institutions.

Several opportunities for more work were identified:

- provide a clear breakdown of the overlap between research software and OSPOs; promoting conversations between OSPOs;
- share each other’s sessions and calls;
- build a national resource for how to interact with OSPOs as both a funder and as an RSE
- make a list of resources for OSPOs publicly available
- build a resource on where to put your OSPO from a funder perspective
- work with ReSA to help existing, funded OSPOs figure out how to support and understand RSE work on their campuses
- put together a list of which organizations provide some of the services of an OSPO, to help academics, researchers, and administrations on their way towards building an OSPO

2. Research Software Ecosystem - Why and how to support research software communities. Facilitator: Dario Taraborelli.

This session was related to the ADORE.software recommendation that “Funders should stimulate the development and maintenance of a research software ecosystem, including people, communities, and infrastructure, to ensure research software sustainability.” This group explored what sustainability means and how funders define it. Sustainability can involve money, maintenance, planning, people, communities, organizations, governance, UX/UI, and so on. Offering training for sustainability lacks an obvious set of experts or guidance. Bringing together the maintainers themselves and creating a community of practice to share lessons learned in sustainability planning would be beneficial.

There is a need for more funders to commit additional resources for maintenance, not only through funding but also by identifying other opportunities to support the ecosystem. For example, recognition for work and contribution to OSS, training all undergraduates to write better software, and creating better citation mechanisms and metrics that use them, which are tied to jobs. Moreover, funders should consider funding capacity-building initiatives that support the ecosystem as a whole, in addition to supporting individual projects. Building community, governance, and documentation are also linked to sustainability. There are existing resources, frameworks, and guidance available (e.g., maturity model - when funding is no longer needed for software, but to sustain the community, CHAOSS metrics, academic research on OSS). As these knowledge resources are dispersed, bringing them together through projects/teams operating in this space would be helpful. Other suggestions include seeking more evidence-led recommendations, to learn from what sustainability means to currently funded projects; leveraging efforts that already have traction and expertise in this area; considering core support for organizations that provide central services (e.g., in the form of training) to multiple funders and their grantee communities; and reversing the funding mechanism to fund projects after the fact, as a function of how much they are used.

Resources:

- Ten simple rules for funding scientific open source software by Carly Strasser et al. includes a discussion of multiple dimensions of sustainability in scientific OSS.
- ITAV: Open Source Software Sustainability toolkit offers a helpful framework for sustainability at the different stages of software.
- Research Software Sustainability takes a Village by Carole Goble.
- Helmholtz Metadata Collaboration
- UKRI Platform Grants (funds teams, not projects)
- Horizon Europe: Research Infrastructures
- European Virtual Institute for Research Software Excellence (EVERSE) aims to create a framework for research software and code excellence, collaboratively designed and championed by the research communities, in pursuit of building a European network of Research Software Quality and setting the foundations of a future Virtual Institute for Research Software Excellence

3. Research Software Personnel - Research software in research assessment reform. Facilitator: Neil Chue Hong

This session was related to the ADORE.software recommendation that “Funders should consider the value and impact of research software as a research output in its own right, to facilitate appropriate reward and recognition measures that enable career progression for all people involved”. Participants in the discussion represented both research funders and larger research performing organizations with a significant contingent of Research Software Engineers.

The initial part of the discussion focussed on current practice, noting that very few research software outputs are being assessed in any of the countries represented in the discussion (Norway, Belgium, UK, Canada) and any indicators are fragmented and field-specific. Some examples of how software is being assessed include NWO’s Strategy Evaluation Protocol and Guidance for Software in REF2021.

Funders do publish data on the outputs of the research they fund (e.g., FRIS, Gateway to Research, NWOOpenAPI) but it is up to researchers to self-report and, unlike data, research software outputs are typically not specifically requested or tracked, meaning the information that can be derived on the value and impact of research software from these databases is poor. Also, the move towards narrative CVs – while positive in many other respects – has the potential to make it harder to assess at scale.

Therefore, using software as a research output as a basis for reward and recognition needs to be done with care, to ensure that metrics and indicators are not misused.

Some practical steps that can be taken by research funders to improve this include:

- Share any existing guidance around how software is being evaluated in a single place to make it findable
- Share existing examples of research indicators that include software
- Identify and agree the different types of impact that research software can have (related to the earlier discussion on metrics)
- Have evaluators with research software expertise on review panels
- Share examples of good narrative CVs and grant proposals that show the impact of the software
- Provide flexibility in what can be requested on a research grant, so you can fund software development professionals like an RSE, rather than just a PhD
- Add “and research software” whenever there is any policy or statement around research (e.g. anything that mentions research outputs, research data)

4. Research Software Ethics - Does anyone measure or address the environmental impacts of research software? Facilitator: Carina Kemp.

This session was related to the ADORE.software recommendation that “Funders should explicitly consider the environmental and social impact of the use of research software”. Participants noted that the environmental impact of research software is an emerging field and something funders should be considering. Moreover, there was some discussion about how to get researchers to address sustainability. For example, how can funders encourage researchers to think about making their code efficient and to be purposeful about their carbon footprint across their research overall? Are there incentives to make it sustainable?

And funders should be thinking about their carbon footprint as well and consider developing policy for sustainability.

Participants discussed existing initiatives and shared some resources:

- Green Algorithms | Green Algorithms (green-algorithms.org)
- The Environmental Impact of Digital Technologies and Data | DataCamp
- UKRI Net Zero Digital Research Infrastructure Scoping Project
    - Final technical report
- Wellcome's report on Advancing environmentally sustainable health research.
- From the ADORE.software Toolkit:
    - Reduce, reuse, recycle: save the planet one GitHub action at a time (Smeets & van Rijn, 2023)
    - Greener principles for environmentally sustainable computational science (Lannelongue et al., 2023)
    - Ten simple rules to make your computing more environmentally sustainable (Lannelongue et al., 2021)
    - The Green Software Foundation
        - Awesome Green Software
        - 2023 State of Green Software report
    - Tracking the environmental impact of research computing (Byrne et al., 2023)

Possible next steps include drafting a statement for funders to use in their funding calls about how researchers are considering the carbon footprint of their work. Participants also noted that developing guidelines for researchers would be helpful. Finally, community work needs to happen alongside these efforts.

Update on ADORE.software

Daniel S. Katz presented an update on the ADORE.software declaration, which came out of early ReSA discussion, where research software funders wanted to exchange experience and identify ways to have a greater impact on research software. These ReSA Funders Forum discussions and other concurrent discussions brought together to create "International Funders Workshop: The Future of Research Software" in Amsterdam in November 2022, with 45 funding organizations represented. In this meeting, the initial draft declaration was discussed, and feedback was provided, leading to a second version (0.3) which was made available for open consultation.

Based on these discussions and comments, we wrote version 1.0, which is also available via https://adore.software. In addition to drafting this, we formed a stewardship group, comprising representatives of the signatories and supported by the ADORE.software secretariat, that is now responsible for overseeing and promoting the endorsement process, which is now starting.

The aim of the declaration is to raise awareness of the role of funding practice in the sustainability of research software, and to improve that practice. It is a first step towards

formalizing, on a global level, the basic principles and recommendations related to funding the sustainability of research software, including the people needed to achieve this goal.

The declaration is structured as a preamble (explaining the need for the declaration, particularly today), and four sets of three recommendations, around Research Practice, the Research Software Ecosystem, People and Research Software, and Research Software Ethics.

In addition to the declaration itself, it includes a link to the ADORE.software toolkit, which provides additional definitions, details, and examples to help adoption and implementation of the ADORE recommendations. It is intended to be a living document, which will be updated regularly, with the suggestions of interested parties.

At the time version 1 was released, about 20 organizations had expressed interest in the declaration. Now, the community is looking for signatories, endorsers, and implementers, as well as simply communication about the declaration. It also would appreciate suggestions for the toolkit, and interested organizations can join the ReSA community to keep updated.

Workshop Day 2 – September 20

Breakout Session #3 – Building Capacity through New Initiatives

5. Research Software Practice - Implementing FAIR for Research Software (FAIR4RS) in funder policy. Facilitator: Daniel S. Katz.

This session was related to the ADORE.software recommendation that “Funders should include research software in open science policies, following the principle ‘as open as possible, as closed as necessary’”. Now that we have FAIR principles for Research Software (FAIR4RS), we discussed what funders can do.

The first topic was based on the fact that some funders are developing guidance, which leads to a set of options: Should funders create guidance individually? Should funders create guidance collectively? How should the community help funders do this?

Second, we discussed that a lot of FAIR is really just a set of routine or best practices. However, a lot of people do not follow these practices.

A third issue is tooling, where different validation tools give different answers, because they interpret FAIR in different ways. This is partially because the principles are described at a high level; what the bare minimum is for all of the principles has not really been determined. Funders might agree on the base set that they care about, and then specific guidelines for disciplines.

Fourth, in implementing the principles, we need different guidance for different levels of people (with reasonable expectations at each level), such as an ordinary researcher who develops software on the side versus a software developer who develops software as the aim of a grant.

Fifth, narrative is important both for applicants and for reviewers. But unless this is a software specific funding opportunity, there will be pockets of knowledge (and of ignorance) in both groups. This is a challenge now with data management plans (DMPs), and it is even more patchy for software.

Sixth, interoperability is particularly difficult to achieve. In some cases, communities have interoperability standards for research software, and funders can list (or vet) organizations that have such standards. However, typically new software development does not make the software interoperable the first time, and so one needs to rewrite/refactor it; funders need to be aware of this.

Finally, funders are now starting to fund programs to develop and maintain software, but these programs are typically aimed at software developers and maintainers directly. They should also consider funding support services that will help these projects succeed, such as what NSF is doing with its POSE program.

6. Research Software Ecosystem - Improving reuse through research software metadata and persistent identifiers (PIDs). Facilitator: Chris Erdmann.

This session was related to the ADORE.software recommendation that “Funders should ensure that funding instruments are fit for purpose for both sustainability and innovation, so that research software is both maintained and developed for the longer term, to encourage a healthy research software ecosystem”. The group focused on improving reuse through research software metadata and persistent identifiers (PIDs). They noted that many funders are still at the sharing data stage, and are not yet addressing research software. Moreover, ORCIDs are not widely used in the developer ecosystem.

Examples of how funders use PIDs include requiring ORCIDs for principal investigators and co-investigators, as well as DOIs for grants awarded and for specific outputs and activities. For others, using PIDs is not a priority.

Possible next steps or approaches:

- Discuss with publishers how to work together on requirements for PIDs (a FORCE11 group has done some work on this already).
- Form a working group to aggregate all PID-related policies for funders, starting with Research Software Funders Forum members. Information to collect includes PIDs used and PIDs mandated or desired.
- Prepare a list of research software-related outputs that are not “PIDable”. An example is the consensus around a new release, and all the threads of feedback and discussion.

Participants also highlighted that there are different options for different outputs, and documentation depends on what researchers want to achieve. Examples include ORCID, Zenodo, COS, Rohub, RAiD (national bodies become registration authority for jurisdiction; also provide service and infrastructure); article PIDs (e.g., JOSS), RRID - Research Resource Identification, ROR ids (Research Organization Registry) and IGSNs (International Generic Sample Number Organization).

Other resources:

- Research Data Alliance's National PID strategies are highlighting identifiers to focus on (ORCIDs for people; RORs for organizations; Crossref/DataCite DOIs for research works/objects; RAiDs for research projects; and Grant ID (Crossref) for research funding. See Open Sciency Sprint.
- Track research and evidence impact with Researchfish by Interfolio is good for capturing publications, but not for capturing other research outputs like software.
- Australian National PID Strategy - v1 - final (ardc.edu.au)

7. Research Software Personnel - Research software directories. Facilitator: Liseanne Cadieux.

This session was related to the ADORE.software recommendation that "Funders should require responsible citation practices for research software that recognise all contributors." Participants explored the role of research software personnel, specifically within the context of research software directories.

This group discussed the effectiveness of the Research Software Directory (RSD) in supporting research software citation. The RSD shows research software in contexts (e.g., research contexts, a page that provides all information but does not store actual data, allows others to see related materials, and shows impact of RSEs). The RSD tries to automate the metadata ingestion – once a GitHub link is there, everything else is updated automatically. Currently, there is no software citation, but this will hopefully be a feature of the RSD in the future. There was a discussion about how to solve the issue of lack of communication between software repositories and directories, as well as how to convince publishers to cite software, including entire software stacks. One recommendation is to include a citation.cff file in your Github. See also CodeMeta.

Participants noted that the software community should challenge the traditional incentive and rewards systems. Citation is not an ideal way to assess impact, as not everyone publishes papers, and that is not the only way software is used in research. Data from other places, such as downloads, dependencies, and so on, can be measured to actually recognize the contributions. Participants noted that more than one metric should exist to be effective, and considered how to evaluate the impact of research software beyond papers. For example, number of downloads (usage) and citation; size of community involved in creating or maintaining the software; research projects using the software; mentions or use in policymaking; user base (number of people using it); and reverse dependency (other projects using it).

Possible next steps:

- Build a guide for researchers on citation. A lot of citation information can be obtained from Zenodo and GitHub. Apart from citation, what other values can we extract from research software?
- Centralize key information (fragmented knowledge) and implement a more coordinated approach for documentation sharing via The Turing Way.
- Create instructions for developers for how to help search (citation and metadata files, but also some best practices).
- Improve metadata for research software.
- Establish a publisher forum to encourage buy-in on software citation requirement (both traditional publishers and research software publishers).

Resources:

- Citation File Format
- Ask researchers to cite software they use before allowing publication in JOSS
- https://github.com/openjournals/joss/issues/1277
- Global Core Biodata Resources: Concept and Selection Process
- Understanding How Researchers Find Research Software for Research Practice by Frankie Stevens, which was based on a survey of researchers conducted by the ARDC and delivered in 2022.
- ARDC report on research software visibility undertaken by Karthik Ram and James Howison, about infrastructure needed to make research software more visible.
- Research Software Capability in Australia by Michelle Barker and Markus Buchhorn.
- Ecosyste.ms as an aggregator of registry data.

8. Research Software Ethics - Engaging with a diverse research software community. Facilitator: Michelle Barker.

This session was related to the ADORE.software recommendation that “Funders should explicitly recognise that diversity, equity, and inclusivity are significant factors in making research software sustainable”. Due to a small number of participants, this group had a brief chat about general issues, then disbanded.

Overview of the Research Software Funders Forum

Kim Hartley, ReSA Community Manager, provided an overview of the Research Software Funders Forum. The Funders Forum – convened by ReSA – is a collaboration of more than 30 funding organizations committed to supporting research software, and those who develop it, as fundamental and vital to research. The Funders Forum began in early 2022, with support from the Alfred P. Sloan Foundation, and has successfully engaged a broad range of participants from government, philanthropic, and industry organizations across the globe.

The Funders Forum provides a formal mechanism for funders to share funding practices, address research software community challenges, and facilitate networks and collaboration. It aims to consider how to achieve long-term sustainability for research software.

The group meets online approximately every six weeks, with two sessions held 12 hours apart and rotating meeting times to accommodate participation in different time zones. During the meetings, funders share sector news, talk about their funding programs, and participate in a discussion topic led by ReSA or their peers. The Funders Forum has a few working groups, led by members. Current working groups are focusing on the following themes: multilateral funding call for research software; implementation of the FAIR for research software principles (FAIR4RS); and towards sustainable and coordinated funding approaches.

The (free) Funders Forum is open to any initiative that funds research software. Organizations interested in participating are invited to contact info@researchsoft.org for more information.

Lightning Talks

Multilateral Call & Global Research Council - Fabio Kon, FAPESP, Brazil - slides
Fabio presented the results of a survey with 280 scientists worldwide about the interest of the international community on a Call for Proposals (CFP) on Research Software Development.

Then, he presented the political difficulties in organizing a Multilateral CFP and suggested we try to use an existing framework (e.g., Chist-era, TAP, Belmont Forum).

Finally, he suggested ways of bringing the Research Software theme to the Global Research Council (GRC), initially as a side event in 2024 and maybe as a main topic in 2025.

Andrew Treloar, Australian Research Data Commons (ARDC) - slides

Andrew Treloar, Director, International Strategy, at ARDC discussed two strands of activity. The first was a program led by Kerry Levett, which involved investing in e-research platforms. The program included 26 projects, AUD $22 million, and a co-investment of one and a half times that from outside the projects. It produced a whole range of different pieces of research software across a variety of domains. The other strand of work was led by Tom Honeyman, and this was developing a research software agenda for Australia. Over the last three years, ARDC has been pursuing research software activities both from the theoretical perspective and the practical perspective of investing directly in research software.

The research software activity identified three different kinds of software: Analysis Code – what typically gets created during a research project, capturing a representation of knowledge generated by that project (the ARDC's ***See*** program of work was created to support that); Prototype Tools – some of that analysis code gets turned into tools for wider use (the ***Shape*** program of activity, i.e., shaping that research software for wider deployment); and Research Software Infrastructure – some of those tools support a range of different projects (the ***Sustain*** program).

Research Data Commons brings together data and the models or software and the underlying computing infrastructure, storage for data, and skills development, to provide this package of services for researchers to use. ARDC currently has three research data commons: the People Research Data Commons (focused on health and medical data); Planet Research Data Commons (ecological and earth science data); and the Humanities, Arts, Social Sciences (HASS) and Indigenous Research Data Commons.

The Australian government is currently developing a national digital research infrastructure strategy. The strategy includes six strategic outcomes, and the fifth strategic outcome is software. The following challenge has been identified: Software is critically important for impactful research; however, its place in the NDRI system is not well defined. This strategic outcome includes recognizing research software tools as critical NDRI resources; enabling software engineering support for qualifying research projects; improving software availability; and maintaining and curating software important to significant digital assets. A draft version of the NDRI strategy will come out later this year. ARDC has released a number of software reports in support of this national agenda. And International Data Week 2025 will take place in Brisbane, Australia.

Keynote by Jean-Baptiste Poline

For the final keynote, Jean-Baptiste Poline from McGill University provided a researcher's perspective on the impact of funders' investments in research software (see video). JB discussed software's impact on FAIR science – having containers that have changed the landscape, but low level open source libraries and numerical instability; analytical flexibility; and software is rarely tested.

He presented the role of research software from multiple viewpoints, beginning with the new research generation (i.e., student perspective). The new research generation needs to learn the following: development, installation, and deployment principles; code and software reusability; peer review publishing software; ethical and legal aspects of software; and reuse (see The Carpentries). From an old research generation (i.e., tenure and promotion committees) viewpoint, consider how to convince deans and provosts of the importance of research software; the long-term versus short-term efficiency and impact of research objects; software as a peer-reviewed research object (the citation piece is needed); and the need for incentives for research software (i.e., no different than for other research objects). With regard to editors and publishers, JB highlighted that crediting software needs education of authors, editors, and reviewers. He noted that many articles do not cite software and this needs to change. Moreover, software journals are still rare (but see JOSS). Finally, sustainability and governance are critical and require work.

JB shared criteria of what makes a good project, specifically distributed development and governance and including contributors' bus factor, labs or principal investigators' bus factor, governance transparency, diverse funding sources, and links to communities. He shared an example of a project in which they aimed to use those criteria from the start of the project: Neurobagel. Neurobagel relies on standards and community; its key concepts include

distributed data governance, sharing, and processings. The project is user-focused, scalable and community standard-based, and developed with partners.

There is also a need to build new publishing platforms and reputation systems. JB has worked on society-based publishing via the Aperture platform – an open publishing platform for the neuroscience community. Aperture focuses on high-quality publishing, open access, low cost, diversity of research objects, and it is community driven. Jupyter notebook preprint server is another example.

JB's keynote offered key takeaways such as the value of building a community of researchers through collaborative projects; the need for community-endorsed standards and practices that support open and FAIR (Findable, Accessible, Interoperable, and Reusable) scientific research (e.g., INCF Network) to maximize research value; the importance of addressing incentives first; and the idea of funding new technologies and large labs but giving precedence to well-governed communities more than centralized projects. He recommended creating the tools needed to change culture, improving training and open practices, developing community standards, and changing incentives.

Do-a-thon

The workshop culminated with a Do-a-Thon, "a short collaboration where participants from different perspectives and skill sets work together on challenges, projects, or to learn something new" (OpenCon Cascadia). Prior to the workshop, participants were invited to submit do-able ideas. During the Do-a-Thon, attendees worked in small groups on the following topics: the Map of Open Source Science; linking software to community usage; and ten things applicants should do to make it easier to assess the impact of funding programs, to build on Ten simple rules for funding scientific open source software by Carly Strasser et al. This dynamic collaboration resulted in participants considering next steps, crowdsourcing tips, and determining some practical and generalizable approaches to address the challenges explored.

1. **The map of open source science. Facilitator: Jonathan Starr**

The Map of Open Source Science (MOSS) aims to:

- Make it simpler for researchers to find what tools will help with their immediate needs
- Make it simpler for researchers to find which projects have the community to help them tweak a tool to their specific needs
- Make it simpler for supporters of OS to find which tools need funding and which funding will create the greatest impact
- Identify gaps in the ecosystem – which domains need more tools
- Avoid abandonware by making it simple to identify points of possible collaboration and integration
- Map OS contributors so that contributors can get credit and recognition for their work

For this do-a-thon, participants viewed MOSS in its current state and explored questions that have arisen from early experiments, including on the scope of the project, how to best build

a map of open source research software, and the details of such a map. Participants discussed the different initiatives that have tried to do this before and how to bring these groups together to form a singular path. The aim is to connect all the initiatives that are mapping repositories, mapping people, and mapping papers separately – to tie this all together to create an outcome that is useful to researchers and funders. They also discussed the idea of mapping policies around the world and the people who influence these policies. Finally, the group explored visualization tools to build this map, and outlined next steps.

2. Linking software funding to community usage. Facilitator: Richard Gunn

The goal of this do-a-thon was to arrive at some practical and generalizable approaches to provide mechanisms to fund software development/maintenance, based on community usage. The general thread was around ways that funders might ring-fence funding from general research programs and then allocate some of that funding towards the development and maintenance of research software through related or possibly distinct funding programs. In many cases, research software funding calls are decoupled from community usage, therefore the session sought to explore practical mechanisms for connecting maintenance and development funding to usage (as a proxy for contribution to research). Ring-fencing of funding was seen as a good model for funders to systematically move towards explicitly supporting RS in their programs. The general thought was that software that is “used” (we did not define “use” in this session other than provide examples such as “citation”) should be funded through such programs.

Initial discussions were around mechanisms to have funders track citations for funded software which would result in ring-fencing funding for that software. This funding could be accessed by those maintaining and developing that software. This could result in a low barrier for peer review and limited administrative overhead, since both the citing project (peer review through publication) and the software itself (peer review through grant funding) had already been peer reviewed. There was discussion of commissioning a third party to assign funding to appropriate maintainers. This immediately brought to light a number of issues, including tracking citations, the piece-meal approach of allocating small amounts of money, awarding funds outside of national boundaries and/or funder remits is challenging, and key parts of the RS software stack are rarely cited. The group also noted that institutional capability to engage in software maintenance activity can be challenging. For example, where incentive structures focus on novelty, as is often the case in funding calls. This led to some discussion on how such capacity could be built, for example by looking to industrial models where software and development teams are separated, and the use of legal structures such as Community Interest Companies (CICs) in the UK, low-profit limited liability companies (US) and Community Contribution Companies (Canada).

Approaches on what and how to fund were discussed, including ring-fencing funding calls to explicitly fund RS software (more money is available if you are developing RS), funding explicitly for moving from “meware” prototype software to sustainable community software, ring-fencing funding that can be used by support organizations (e.g. DRI providers) to support RS after it has been developed, and in realizing that not all software will live on

forever funding the sunsetting of software should also be supported. The latter point produced a lively discussion on new models for sunsetting software programmes, such as providing funding to users to transition to suitable alternatives, which was considered to be a promising model for further exploration.

It was pointed out that ring-fencing models may align well with the emerging evidence that RS software development is widespread (e.g. 20% of NSF projects over 11 years discussed software, ~50% of researchers say they produce software as a primary output of their research). It was suggested that funding agencies should be able to come up with a formula to ring-fence a portion of funding for supporting the development and maintenance of this software. The funder could then provide funding programs for further development and/or sustainability of software that is heavily utilized through funding calls. RS projects can apply to these calls, with the burden of evidence on the projects to demonstrate "use" and "research value" of the software being developed. It was suggested that assessment of these proposals should be based on the "value of the research" that is enabled by the RS. It was pointed out that use might be broader than national use and therefore it is unlikely that one could rely on "use" and "value" from within the same funding program and/or national borders. Some programs like this exist already, two relevant cases being the Netherland eScience Centre's Call for Sustainable Software 2023[1] and CZI's Essential Open Source Software (EOSS) program[2].

In conclusion, the group agreed that, while there were practical barriers that would need to be overcome, it would be beneficial for ReSA members to share case studies and continue to develop these principles based on the evaluation of pilot activities conducted by members.

Ten things applicants should do to make it easier to assess the impact of funding programs. Facilitator: Neil Chue Hong

The objective of this do-a-thon was to gather a set of tips and tricks to be able to better measure the impact of software programs. The participants were charged with collecting their thoughts and discussing metrics and KPIs of funded software and tools that could easily be included in proposals to funders and subsequently monitored through the lifecycle of the award. Both quantitative and qualitative metrics were discussed to distinctly recognize adoption and impact of software and tools.

The quantitative metrics shared were the number of users of the software, downloads, and citations to name a few of the commonly used indicators. The group discussed measurement of contributors, especially those from outside the funded organization and the frequency of external contributions. Qualitative measures included those that could indicate research area advanced and breadth of use case applicability, maturity of the software, diversity of the contributors, longevity of the software, and such. CZI has a fairly detailed list of metrics they expect in applications and the group discussed each of those in https://chanzuckerberg.com/rfa/essential-open-source-software-for-science/. NSF suggests applicants to propose metrics they want their software evaluated. ARDC requires a

[1] https://www.esciencecenter.nl/calls-for-proposals/call-for-sustainable-software-2023-ss-2023/
[2] https://chanzuckerberg.com/eoss/

contractual requirement to provide an impact statement at the end of the project. Given the various approaches used by funders, the group thought it to be prudent that funders consider organizing a workshop with CHAOSS to develop KPIs together that can be broadly shared.

Based on the discussion, ten things applicants can consider including in order to indicate to funders and be able to measure the impact and outcome of their funded software:

1. Consider open development from the get-go with a roadmap
2. Include a DMP for the software to be developed that includes both contributors and consumers' perspectives of the software
3. Consider and include a communication/outreach plan to build a community of both users and contributors
4. Draft a management plan that includes plans for governance of the software
5. Identify and define quantitative (output measures) metrics to be measured –
    a. Software metrics: usage, downloads, citations, Commits
    b. People metrics: users, contributors etc.
6. Determine a few qualitative (outcome measures) metrics –
    a. Longevity of the software
    b. Maturity of the software
7. Determine a few efficiency metrics to track such as software quality or use of the software for use cases across domains
8. Actively seek collaborations across domains
9. Capture use cases (real-world) and share them openly
10. Monitor the KPIs with or without a dashboard and share as much as possible openly

Rapporteur - Towards a Sustainable Ecosystem

Hannah Hope, Open Research Lead at Wellcome Trust, summarized the workshop. Hannah began by reflecting on a point from JB Poline's keynote that incentives for research software are no different for other research objects. She noted that there are some core shared problems across all research output types (e.g., data and software) and research processes and that we need to come together collectively to try to tackle some of those and not form our own silos, which can work against each other.

Examples of shared challenges include how to effectively credit contributors to research (this is a shared issue irrespective of research outputs); how to fund and support communities of research and collaborations of research, not just principal investigators; and how can we assess research in a way so it is not a massive own goal? There are opportunities to work with the data community and movements such as COARA to collectively tackle all players together (e.g., institutions, funders, and ourselves as peers and assessors of research). Moreover, Hannah highlighted similarities between data and software, such as how to nurture your community (both contributors and collectors) and how to discover work and re-use and incentivize it.

Hannah noted that there are some differences between research output types. Research software, in particular, has a unique challenge related to sustainability. Given the different

levels of sustainability, how can projects migrate across that evolution? And how do we manage the death of resources? What can funders do to keep the community moving forward and sustaining it but without letting it grow beyond something that does not have long-term feasibility?

Another key theme that arose throughout the workshop was impact. More specifically, what is the value of citation and how do we do it? What are the different needs within the community? There are a range of technical, cultural, and infrastructural barriers to consider. Hannah suggested that there is real value in working with the data community, as there is a lot of momentum within that field currently. Another area for reflection is the idea of principal investigator versus team. How can funders get creative about supporting the infrastructure and software around the research? Moreover, Hannah emphasized the value and importance of the software community and non-monetary contributions that make software work. When reflecting on advancing software, we need to ensure we do not diminish those in-kind contributions that make software successful. Communities around the software will enhance the traction that you have both within the funder and also in maximizing the impact of the software.

Finally, there was a lot of discussion around “unicorns” and the need for detailed guidance for different users and different areas. Hannah noted that we need fragmentation. But we also need to federate our fragmented systems. What are the minimum-shared rules we need to have for software across the different research domains? Each domain or part of the ecosystem does not need to understand the entire system, however. How do we connect the software experts with different parts of the community and put it together?

3. Next Steps

This report is a complete summary of the Workshop, which is augmented by the blog post published by ReSA on October 24, 2023. Organizations that fund research software (as a primary or secondary mandate) are encouraged to participate in the ongoing Research Software Funders Forum meetings, get involved in ReSA’s working groups, and participate in the next Forum. Researchers derive a great deal of value from a healthy research software ecosystem, and any effort designed to bring researchers, software developers, funders, and policy makers together to discuss how to build a more sustainable ecosystem can only increase this value.

The SciLifeLab Data Centre and ReSA are hosting the next funders workshop in Uppsala, Sweden, in September 2024. The theme of the workshop is “Towards a monitoring framework to benchmark the ADORE.software recommendations and improve the sustainability of research software.”

The workshop will focus on operationalizing the Amsterdam Declaration on Funding Research Software Sustainability (ADORE.software) by developing a monitoring framework to benchmark how funders are currently supporting the sustainability of research software using the recommendations as a starting point. Additionally, the workshop will examine how

such a monitoring framework can be leveraged by funders and the community to prioritize critical areas for improvement.

The workshop will build on the first (2022) and second (2023) international funders workshops, and Research Software Funders Forum, involving 60+ funding organizations from across the globe. The 2022 workshop initiated the drafting of the ADORE.software, which was released for signing at the 2023 workshop.

4. Acknowledgements

ReSA would like to express its sincere gratitude to our community, Founding Members, and Organizational Members – without whom none of this work would have been possible. This project has also been made possible in part by grant 2021-000000 from the Chan Zuckerberg Initiative DAF, an advised fund of the Silicon Valley Community Foundation; and the Alfred P. Sloan Foundation.

The Alliance would like to thank ReSA and colleagues in the Funders Forum and broader research community for participating in this event. The Alliance supports ReSA because it is an effective and responsive organization facilitating the discussion to build a stronger global research ecosystem.