Check for updates

OPEN LETTER

REVISED Understanding and advancing research software grant funding models

[version 2; peer review: 2 approved, 3 approved with reservations]

Daniel S. Katz [1], Eric A. Jensen [1,2], Michelle Barker [3]

[1]University of Illinois Urbana-Champaign, Urbana, Illinois, USA
[2]Institute for Methods Innovation, Dublin, Ireland
[3]Research Software Alliance, Portland, OR, USA

v2 **First published:** 25 Jul 2025, **5**:199
https://doi.org/10.12688/openreseurope.20210.1
Latest published: 11 May 2026, **5**:199
https://doi.org/10.12688/openreseurope.20210.2

Abstract

Research software funding currently operates across a disconnected landscape of public and private grant-making organizations, leading to inefficiencies for software projects and the broader research community. The lack of coordination forces projects to pursue multiple, often overlapping opportunities, and forces funders to independently evaluate projects and proposals, resulting in duplicated effort and suboptimal resource distribution. By examining existing collaboration models, including centralized and distributed approaches, we highlight how joint decision-making mechanisms could improve sustainability for reusable software resources. An international set of examples illustrates how cross-organization cooperation for research software funding can be structured. Such collaborations can optimize grant disbursement and align priorities. Increased collaboration could allow funders to better address the ongoing maintenance and evolution of research software, lowering barriers that hamper discovery across multiple research domains. Encouraging both bottom-up user-driven and top-down coordination mechanisms ultimately supports more robust, widely accessible research software, improving global research outcomes.

Keywords

research software, research funding, research policy, digital tools, international research systems, digital infrastructure

Open Peer Review

Approval Status ✓ ? ? ? ✓

	1	2	3	4	5
version 2 (revision) 11 May 2026	✓ view				✓ view
version 1 25 Jul 2025	? view	? view	? view	? view	✓ view

1. **Jukka Ruohonen**, University of Southern Denmark, Sønderborg, Denmark
2. **Katherine Skinner**, invest in open infrastructure, USA, USA
3. **Filomena Canterino**, Polytechnic University of Milan, Milan, Italy
4. **Mahirah Kamaludin**, Universiti Malaysia Terengganu, Kuala Terengganu, Malaysia
5. **Ludo Waltman**, Leiden University, Leiden, The Netherlands

Any reports and responses or comments on the article can be found at the end of the article.

This article is included in the Horizon Europe gateway.

Corresponding author: Eric A. Jensen (eric@methodsinnovation.org)

Author roles: Katz DS: Conceptualization, Data Curation, Funding Acquisition, Investigation, Project Administration, Resources, Supervision, Validation, Writing – Original Draft Preparation, Writing – Review & Editing; **Jensen EA**: Conceptualization, Formal Analysis, Investigation, Validation, Writing – Original Draft Preparation, Writing – Review & Editing; **Barker M**: Conceptualization, Validation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No. 101177493 (Rare earth and magnets hub for a resilient Europe [REMHub]). The Alfred P. Sloan Foundation funded this research via a grant to the National Center for Supercomputing Applications at the University of Illinois Urbana-Champaign under grant agreement No. G-2022-19344 (Charting the Course: Policy and Planning for Sustainable Research Software).
The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2026 Katz DS *et al*. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Katz DS, Jensen EA and Barker M. **Understanding and advancing research software grant funding models [version 2; peer review: 2 approved, 3 approved with reservations]** Open Research Europe 2026, **5**:199 https://doi.org/10.12688/openreseurope.20210.2

First published: 25 Jul 2025, **5**:199 https://doi.org/10.12688/openreseurope.20210.1

REVISED **Amendments from Version 1**

This revised version clarifies the scope, framing, and intent of the Open Letter and strengthens alignment between the abstract, body, and conclusions. Several passages have been revised to make explicit that the article is not a hypothesis-testing research paper but a policy-oriented contribution grounded in the authors' collective experience as research software funders and intermediaries.

The definition and scope of "research software" have been clarified to emphasise a focus on reusable software intended for use beyond a single publication or project, distinguishing it from single-use analytical code. Additional context has been added to explain how the proposed software stages were constructed and how they relate to existing definitions, while acknowledging the fuzzy boundary between research software and surrounding infrastructure.

The motivation section has been strengthened through the addition of explicit discussion of software maintenance as a long-term cost and structural challenge, including why maintenance has historically been difficult to support through grant-based funding mechanisms. Claims about funding challenges and inefficiencies are now more clearly framed as informed observations drawn from practice and funder engagement, rather than as empirically established findings.

The paper now includes clearer signalling around the origins of the funding challenges listed, improved definitions of specialised terms and acronyms, and modest restructuring of longer paragraphs to improve readability. References have been updated where newer peer-reviewed publications have become available.

Finally, the conclusion has been expanded to more clearly situate coordination proposals as illustrative options intended to stimulate discussion rather than prescriptive solutions, and to explicitly recognise the role of academic incentive systems alongside funding structures in shaping research software sustainability.

Any further responses from the reviewers can be found at the end of the article

1. Introduction

In this paper, we examine the challenges in the way the development and maintenance of research software is currently funded by an array of different public and private organizations. Here, we focus in particular on reusable research software. We note that there is typically little coordination between these funding organizations. This leads to inefficiencies for research software projects seeking funding, the funding organizations, and the overall research software field. We also discuss the existing models for cross-organization coordinated funding and models that could reduce inefficiencies and lead to better sustained research software that improves research overall. We recommend that new and existing funders wishing to increase research impact consider coordinated, collaborative approaches to research software resourcing.

2. Challenges to funding research software

Research software is defined by Gruenpeter *et al.* as "source code files, algorithms, scripts, computational workflows and executables that were created during the research process or for a research purpose," and distinguished from *software in research*, in turn defined as "software components (e.g., operating systems, libraries, dependencies, packages, scripts, etc.) that are used for research but were not created during or with a clear research intent" (2021). They also note that "this differentiation may vary between disciplines," for example, a compiler could be research software for a computer scientist performing research in compilation or programming languages, but likely would not be for an astrophysicist performing research on gravitational waves. We have chosen to use this term and definition in this paper rather than *scientific software*, as research software is a more inclusive term that also encompasses research fields such as the humanities and engineering.

Research software can be developed by one person/group and used just for their own research, or it can be intended for use by multiple groups. Software can be open source in both cases, since this is a function of licensing or permissions, not intent. In this paper, we focus on the research software that is developed for research use by people other than the developers. To be precise, we could call this *reusable research software,* but for simplicity's sake, we will refer to it just as *research software* going forward. We do not intend for this to include software that is reused solely for replication of results, which is one reason to make software open source without it being intended for others to use it. Note that Section 3 also contains a description of some of the many previous categorizations of research software.

There is also a challenge in how research software is considered by funders, who are often focused on novelty. The software we are discussing here may be novel, but it is probably better described as infrastructure than research. Most research funders recognize that they need to support and maintain infrastructure to enable the best research to be performed, for instance, instruments (e.g., telescopes, sequencers), data systems (e.g., archives), and platforms (e.g., ocean-going research vessels). However, most do not routinely consider software as part of the research infrastructure that

needs support for both development and maintenance. As is widely discussed in the software community, and as captured by Hinsen (2019), software needs ongoing maintenance to continue to work, as it is part of a dynamic environment: it needs to be kept working when the software and hardware used changes; it needs to be kept working correctly when a bug or other problems such as security holes are found; and for it to be useful, it needs to be updated as its research field changes with new applications and new methods.

2.1 Funding processes

Having defined what we mean by (reusable) research software in this paper, we now turn to the resources needed to develop and maintain it. These resources can be acquired from the software's users, as is the case for commercial research software or when a research project contributes funding or personnel to a software project to ensure that it can provide the features or support needed for the research project. The resources can also be acquired from the developers, who may work on the software because they need it for their own research (similar to the previous) because their institution sees it as valuable for future research or reputation. They can also be provided by volunteers, who could be interested in seeing the software succeed for a variety of reasons. Finally, the resources can be acquired from a third party who invests in the research software because they want to support the research it will enable. This is typically the case for research software that receives grants from government, philanthropic, or commercial sources. And this latter case is where we will focus.

The predominant model for grant funding of research software today is one of multiple grant-making organizations and multiple software projects seeking grants. In most cases, the grant-making organizations create calls for proposals with some focus, perhaps in terms of the discipline(s) that the software should impact, the type of software work that will be supported (development of new software, maintenance of existing software, conversion of existing private software into a community tool, community growth, increased diversity, etc.) The process by which an organization decides what areas to support varies widely, and inputs can be either internal or external, including the organization's management, external advisory bodies, or higher-level stakeholders (e.g., a government ministry). In this paper, we assume this decision process has already happened, and we focus on the implementation of the funding program.

A large number of research grant-funding programs exist, but only a small number focus on research software. Many of the latter are captured and shared by the Research Software Alliance (ReSA) in its public database of current and past research software funding opportunities (Research Software Alliance, 2025). Section 3 of this paper discusses some examples.

In this environment, it is up to research software projects that need support to be aware of all relevant grant opportunities and to write proposals to each, while being aware of possible policies against the submission of the same work to multiple organizations and against receiving multiple grants for the same work.

The grant-making organization then determines what to fund, typically using peer review. This review process can either provide inputs on the value of the proposals that the organization can use to determine which software to support, or the peer-review process can make the decisions for the organization.

In this model, there is no coordination between the decision-making processes of each opportunity as provided by different organizations, which can lead to some projects being awarded multiple grants and others none, without full regard for the need for the software work. And because very few software projects are independent of all others, this can lead to a failure to support software projects on which other software projects, and in turn, research projects, are dependent.

A call for proposals is not the only model for grant-making organizations. Some organizations also work in a more discretionary and less open manner, where staff members find projects to which they want to consider making grants, and then work in private to plan and provide such grants. From the point of view of a software project seeking support, this has both positive and negative aspects. The positive part is that they may not need to be aware of funding opportunities, as the opportunity comes to them. Negative parts include that this may support well-known, well-publicized, and well-connected projects (or their leaders) rather than those with an equal need for support but not the same reputation or interpersonal connections. This also does not resolve the problem of the overall funding opportunities across funders not being matched to the needs across projects.

2.2 Funding challenges

Research software projects typically begin under a grant. Because research grants are of fixed duration, the project needs to find new resources to continue to develop and/or maintain the software when the grant ends. This becomes an ongoing process for most software projects, where the resources needed are typically a mix of grant funding for people to do work,

volunteers and collaborators, and other resources, and the work to be done includes software work such as development and maintenance, documentation, and testing, along with community work, such as user support, community development, etc. A typical small- to medium-sized project moves between times when resources are available because of a grant and work continues smoothly, and times when these resources are not available when project work falls to what volunteers can provide while they are also seeking new resources. Larger projects with staff dedicated to sustainability (including fundraising) can stay in the smoother periods more often. This is captured in workaround software project lifecycles, e.g., Yehudi *et al.* (2025), which proposed a lifecycle model that includes the concept of challenge events, such as a resource shortfall, and how these can affect the project.

In this paper, however, we focus on the challenges seen by funding organizations. Some issues are specific to funding research software, particularly understanding what is needed (required functions/capabilities) by research communities. Challenging questions include:

- Which software tools are most frequently used by scientists in any given field?
- How does the use of open source compare to proprietary software in any given field?
- Are emerging new tools replacing legacy ones?
- What is the prevalent programming language in any given field?
- Which software tools should be part of a student's computational curriculum?
- Which software projects should funders prioritize as critical infrastructure for science? (CZI, 2024)

These kinds of priority-setting and implementation challenges have also been documented in an international survey of research software funders (Jensen & Katz, 2025a).

Beyond this, multiple funding programs or organizations that want to collaborate, perhaps with multiple countries or organizations pooling resources to address common challenges, often encounter barriers that hinder collaboration. Based on our experiences as research funders and in discussions with funders while running the ReSA Funders Forum and its three conferences, these include:

- Alignment challenges. Divergent priorities due to different strategic goals or geographic focus areas, and timeline mismatches on granting cycles and decision-making processes.
- Risk aversion. Loss of control or reputational concerns related to collaborating on large-scale programs.
- Relationship and trust barriers. Lacking the established relationships needed to connect with other funders, and trust deficits as building the trust necessary for effective collaboration takes time.
- Operational complexity. Governance issues when establishing shared models for decision-making and accountability, and measurement and reporting expectations for reporting outcomes, along with differing requirements on proposals, and different policies for project budgets and project operations.
- Legal and compliance issues. Regulatory complexity due to differing compliance requirements across jurisdictions, and lack of legal frameworks to participate in pooled funding initiatives, where funds contributed by multiple funders are distributed through a centralized process, not by each funder individually
- Resource constraints. Limited administrative capacity and financial limitations, such as a lack of discretionary funds needed to participate in pooled funding.

Some funders have chosen to collaborate to address these challenges, and some models of successful collaboration are presented in Section 4.

3. Grant-funding programs for research software

The Research Software Alliance (ReSA) maintains a public database of current and past research software-specific funding opportunities (2025), and here we provide examples of some grant-funding programs, taken in part from that database. Table 1 shares examples across some of these categories to illustrate the types of research funding that currently exist. There are a number of different aspects to these programs, such as funder type, disciplinary focus, stage of software supported, funding mechanism, and duration of funding. An explanation of these and other relevant aspects is given below.

An example of a national and disciplinary-focused program that supports existing research software is NIH's Building Sustainable Software Tools for Open Science (OD-24-010). This program supports enhancing the sustainability and impact of research software tools by enabling the use of best practices and design principles in software development and by leveraging continuing advances in computing. This call is also aimed at facilitating the creation of partnerships between developers and users of software and tools, and promoting FAIR (Findable, Accessible, Interoperable, Reusable) (Chue Hong *et al.*, 2022) practices for research software to maximize research value.

While rarer, there are also industry research software funding programs. For example, the academic publisher Sage's Concept Grants program funds innovative products and tools aimed at enhancing social science education and research. One class of awards is for Doing Research: Developing advanced research software tailored for academic research in the social sciences (Sage Research Methods Community, 2024).

The Digital Technology Development Awards (Climate-Sensitive Infectious Disease Modelling) program, run by the UK-based Wellcome Trust philanthropy, is designed to fund the development of digital tools that will improve climate-sensitive infectious disease modelling and support their maintenance later on. This funding is aimed at supporting the delivery of open-source digital technology that improves the impact of research relevant to this theme of climate-sensitive infectious disease modelling.

Table 1. Illustrative examples of research software funding categories.

Funding Program	Funder	Funder Type	Disciplinary Focus	Stage of Research Software Supported	Funding Mechanism
Building Sustainable Software Tools for Open Science	National Institutes of Health (NIH)	National Government - USA	Biomedical research	Enhancement; infrastructure	Standalone grants
Concept Grants	Sage	Industry	Social science	Prototype; enhancement	Standalone grants
Digital Technology Development Awards (Climate-Sensitive Infectious Disease Modelling)	Wellcome Trust	Philanthropic	Health	Prototype; enhancement	Standalone grants
Research Software - Quality Assured and Reusable	German Research Foundation (DFG)	National Government - Germany	Interdisciplinary	Enhancement; infrastructure	Standalone grants
Support for Open-Source Tools, Frameworks, and Libraries	NASA	National Government - USA	Space science	Infrastructure	Standalone grants
Supplement for Open-Source Science	NASA	National Government - USA	Space and earth science	Prototype; enhancement	Supplement to existing award

An example of national cross-disciplinary funding is the German Research Foundation's (DFG's) program, Research Software: Quality Assured and Reusable. This is an interdisciplinary call for proposals from German organizations that pursues three sub-goals, usability and impact, quality assurance, and further development, which together seek to raise the maturity level of research software so that it can also be used by researchers other than those developing it, and to simplify its further development.

An example of a national and disciplinary-focused program that supports existing research software is NASA's Support for Open-Source Tools, Frameworks, and Libraries (OSTFL, ROSES-24 F.7). This supports the improvement and sustainment of existing software that has significantly impacted the NASA Science Mission Directorate (SMD) science community. The program element seeks to support two types of awards. 1) Foundational awards: cooperative agreements for up to five years for open-source tools, frameworks, and libraries that have a significant impact on two or more divisions of the SMD and 2) Sustainment awards: grants or cooperative agreements of up to three years in duration for open-source tools, frameworks, and libraries that have significant impact in one or more divisions of the SMD.

An example of a nationally and disciplinary-focused program that is only available to existing grantees is NASA's Supplement for Open-Source Science opportunity (ROSES-24 F.8) NASA supports adding an open science component to an existing "parent" ROSES (NASA's overall Research Opportunities in Space and Earth Sciences funding program) award. The goal of this program element is to increase the accessibility, inclusivity, and reproducibility of the science from the parent award, to contribute back to the open-science communities relevant to the parent award, and/or provide cloud credits to further support or expand the parent award. This could include taking software developed in the project supported by the parent award and further developing it for additional usage.

3.1 Funder type

The funder type (and the funding model) category differentiates between the kinds of various organizations that financially support research software projects. A key funding source is governmental funding, usually national. Government agencies provide substantial resources to advance research and development nationally. Governmental funding is often driven either top-down by political priorities and strategic initiatives or bottom-up by research community requests and expert decision-making. In some countries, similar funding structures exist at the sub-national level, for example, in Brazil with the São Paulo Research Foundation (FAPESP) and the Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ) in the states of São Paulo and Rio de Janeiro respectively. More unusual are supranational funding structures, such as those administered by the European Commission on behalf of European Union Member States and Associated Countries.

Another source is philanthropic funding. Philanthropic funders such as the Chan Zuckerberg Initiative (CZI) may focus on specific areas of interest or innovative approaches that align with their mission and values. They often aim to complement or fill gaps left by governmental funding programs, and sometimes collaborate directly with governmental funders. A third category is industry funding, provided by companies.

3.2 Disciplinary focus

This category addresses the scope of research fields targeted by funding programs.

Some programs have a *specific discipline/field* focus, aiming to advance software development within particular scientific areas such as space sciences or biomedical research. For example, NASA's programs are tailored to support software that benefits space and Earth sciences, while NIH and CZI focus on research software for biomedical science. A limited disciplinary range can limit options for coordination between funders.

Conversely, *inter/multidisciplinary* programs are open to multiple disciplines or encourage cross-disciplinary collaboration.

It is also relevant to consider the stages of software supported.

This category considers the development phase of the research software projects that funding programs support. Here, we use a three-stage classification:

1. *Prototype development* (Stage 1) refers to creating early-stage software, where initial concepts are transformed into functional prototypes.

2. *Enhancement of existing software* (Stage 2) involves improving or adding new features to already developed software. This could include optimizing performance, expanding functionality, or adapting the software for

additional use cases. Likewise, *maintenance and sustainment* (Stage 2) focus on the ongoing support required to keep software operational, secure, and up-to-date. This includes activities like bug fixes, updates, and user support.

3. *(Research) Infrastructure development* (Stage 3) pertains to building foundational systems or platforms that support other research software applications. Funding for infrastructure is designed to create robust environments and structures upon which other tools and applications can be developed and executed.

There have been many previous categorizations of research software, such as work by Hinsen (2019) on the layers of software above the operating system, from 1, non-scientific infrastructure such as gcc and Python, and R, to 2, scientific infrastructure such as BLAS, HDF5, SciPy, and the tidyverse, to 3, domain-specific tools such as GROMACS and MMTK, to 4, project-specific code such as scripts, notebooks, and workflows. Hasselbring *et al.* (2025) offer a multi-axis approach, where one axis is "role in research", which includes proof-of-concept software and research infrastructure software as categories. Maturity is another axis for Hasselbring *et al.*, 2025, with categories of research data processes, novel methods and models, and accepted methods and models. This maturity axis was originally proposed for the Australian Research Data Commons (ARDC) by Honeyman and Treloar (2021), who highlight that different activities are needed at each level: increased sharing for research data processes, application of best practices to novel methods and models, and increased maintenance for accepted methods and models.

An alternative model was developed by the German Aerospace organization, Deutsches Zentrum für Luft- und Raumfahrt (DLR) (Schlauch *et al.,* 2018). It includes 4 classes, from 0) personal use and small scope and not planned for distribution outside of the local organization, to 1) possible for non-developers to use and open to extension/development by external people, to 2) having a defined development process and a plan for long-term development and maintenance, to 3) critical software with product characteristics. The European Virtual Institute for Research Software Excellence (EVERSE) has released a Research Software Quality Kit (RSQkit) (https://everse.software/RSQKit) which includes a three-tier view (EVERSE, 2024). This is based on the ARDC model, consisting of analysis code, prototype tools, and research software infrastructure, though this also roughly corresponds to DLR's class 0, 1–2, and 2–3, respectively.

Our proposed categorization above follows from this previous work, most closely following ARDC's model, but with slight changes to better fit this more general context.

3.3 Funding mechanism

The funding mechanism category describes how funds are allocated to research software projects. A *supplement to an existing award* is additional funding to augment current projects. This mechanism allows researchers to expand the scope of their work, integrate new components, or address unforeseen challenges without securing a completely new grant. A supplement to a research grant that supported software development for that research might be used to generalize the software to support the research of others.

In contrast, *standalone grants* are independent funding opportunities that do not need to be tied to existing research projects. These grants enable researchers to initiate new projects or pursue innovative ideas independently of their current work. Within this category are (a) standalone *grants for research that include research software projects* and (b) dedicated funding specifically for research software per se.

4. Existing coordination of funding organizations

A range of models for funding programs designed to address shared international priorities collaboratively exists. These models typically involve funding organizations either pooling their resources to be centrally distributed or agreeing to work together under specific conditions, with funders only providing resources to their local teams selected for funding through a shared process. Here, we present some examples of such funding models, highlighting the potential to draw inspiration from such established structures.

Few examples exist of philanthropic funders coordinating their efforts and running shared programs specifically focusing on research software. We focus here on four examples that illustrate both philanthropic and government funder collaborations, and pooled funding that is administered either by a single centralized process or through separate but coordinated processes, as shown in Table 2.

Another example of multi-funder coordination that provides a different model, albeit one not yet seen with a focus on research software, is the Belmont Forum, which coordinates both national government and philanthropic funders.

Table 2. Illustrative examples of research software funding collaborations.

Funders	Program	Funder Type	Administration
CZI, Kavli Foundation, Wellcome Trust	Essential Open Source Software for Science (EOSS)	Philanthropic	Separate, coordinated processes
Gordon and Betty Moore Foundation, Footprint Coalition, Schmidt Futures	Low-Cost Tools for Science	Philanthropic +	Centralized
CHIST-ERA consortium members	Open and Re-usable Research Data and Software	Government	Separate, coordinated
European Commission	Horizon Europe Framework Programme (HORIZON): Digital and emerging technologies for competitiveness and fit for the Green Deal: Fundamentals of Software Engineering	Government	Centralized

4.1 CZI, Kavli Foundation, and Wellcome Trust: EOSS

These three international philanthropic funders collaborated on the sixth cycle of the EOSS program in 2024. The program supports open-source software projects that are essential to biomedical research, specifically the software maintenance, growth, development, and community engagement for these critical tools. It seeks projects from all over the world that have previously demonstrated impact, currently show potential for continued improvement, and are expected to deliver added value to the biomedical research community through the proposed activities.

While CZI was the sole funder in the first five rounds of the EOSS opportunity, the Kavli Foundation and Wellcome Trust joined CZI in the sixth. CZI managed the program and the peer-review process. After that, each funder chose the software they wanted to support, and the funders discussed this, leading to more coordinated funding decisions. This improves the situation for the software projects funded, compared to them having to apply independently to these three funders, followed by the funders running separate evaluation processes and making independent decisions. While it reduces the work for software projects and leads to more coordinated decisions, it does not include many other funders, so the projects cannot rely on only this program (Hertweck *et al.*, 2024).

4.2 Gordon and Betty Moore Foundation, Footprint Coalition, Schmidt Futures: Low-cost tools for science

This 2023 program supported projects that aim to build and/or test new low-cost tools for science. Low-cost science tools can increase access to scientific experimentation and inquiry for individuals and organizations that may not have the resources to purchase expensive laboratory equipment and supplies. This can help close the gap between those with access to science resources and those without access. Low-cost science tools can also help to encourage innovation by providing individuals and organizations with the resources they need to develop and test new ideas and theories. This can lead to discoveries and advancements in scientific knowledge. This small, fast grant program aimed to get new ideas off the ground quickly, backing eligible projects up to USD$10,000. Funds were distributed on a first-come, first-served basis (Experiment, 2023).

4.3 CHIST-ERA: Open and re-usable research data and software

CHIST-ERA is a consortium of research funding organisations in Europe and beyond supporting use-inspired basic research in Information and Communication Technologies (ICT) or at the interface between ICT and other domains. This collaborative funding structure is limited to the European Research Area. It has core funding from the European Commission and brings national funders together to create a transnational call for proposals. Each year, CHIST-ERA's participating funding organizations identify emerging scientific domains that enable European researchers to undertake high-risk, high-impact initiatives to advance knowledge. In pursuit of this goal, they issue transnational calls for proposals focused on two priority ICT topics. These calls are formulated and published within six months of selecting topics, drawing on the research community's expertise and a strategic long-term vision to maintain their relevance in rapidly evolving research fields.

In 2023, one of CHIST-ERA's funding calls was focused on: Open and Re-usable Research Data and Software. This call tackles the challenge of open research data and software from the perspective of their possible reuse. While there was a total funding pool of 6 million euros, each partner in the successful research projects is funded separately by their respective national/regional research funding organization. This is an example of the model where each funder provided

resources to the research teams that are within their typical geographical scope; each research project has to include a minimum of 3 partners requesting funding from participating funding organizations from at least 3 of the following countries: Belgium, Brazil, Czech Republic, France, Latvia, Lithuania, Luxembourg, Poland, Romania, Slovakia, Switzerland, Turkey, United Kingdom (CHIST-ERA, 2024).

4.4 European Commission - Horizon Europe Framework Programme (HORIZON), Digital and emerging technologies for competitiveness and fit for the Green Deal: Fundamentals of Software Engineering

This call aims to progress the state of the art in responsible software engineering methods, tools, and best practices leveraging, among others, novel AI and data technologies to accelerate the development and maintenance of software, including for multi-architecture systems, addressing in particular efficient and agile modelling, verification, and validation, as well as vulnerability assessment and mitigation. Projects are expected to demonstrate their developments in at least three industrial or societal use cases. Implementing responsible software engineering, the use-cases should address functional and non-functional requirements and principles like optimising energy usage, reducing the environmental footprint, security-by-design, and data protection (European Commission, 2022).

4.5 Belmont Forum

The Belmont Forum is an international partnership that mobilizes funding of environmental change research. Since its formation in 2009, the Forum has successfully led 22 calls for proposals, supporting 181 projects in over 90 countries. Each of the Belmont Forum's funding programs comprises an ad hoc coalition gathered to address a shared focus called a Collaborative Research Action. The Forum works to bring prospective partners into the scoping phase at the earliest juncture so they can co-produce the developing Collaborative Research Action agenda. Yet this process remains fluid enough for organizations, whether they offer monetary or in-kind support, to join at any point, reflecting the realities of collaborative engagement between research funders that operate on different timescales. Each participant's contributions, core thematic interests, eligibility parameters, and primary program contacts are documented in a publicly available document linked to the Collaborative Research Action funding call. Awards are made through coordinated funding from individual organizations, that is, the funding is not administered directly by the Belmont Forum, but by the collaborating organizations to the research projects within their remit (which is often geographic but can also be international) (Belmont Forum, 2017).

5. Options to democratize funding decision-making and coordination

We propose three main categories of decision-making for research software funding: (1) funder-driven; (2) user-driven; (3) developer-driven (Table 3).

The models we have discussed so far are all funder-driven, though with differences in how the decisions are actually made, and in possible coordination mechanisms between funders. Typically, public (e.g., government) research funders make decisions based on peer review that evaluates potential projects, and this is also used by some philanthropic funders and industry. Some funders make decisions based on their own knowledge of needs, perhaps informed by internal and/or external feedback on one or more projects selected for potential funding.

In the other two categories, funders still provide resources, but the community determines how to allocate them. This could be done at two different levels: software projects and researchers/users. The funder-driven category can be viewed as top-down, while the software developer-driven category can be viewed as bottom-up, with the user-driven model in the middle.

Table 3. Locus of decision-making in research software funding.

Decision- making Locus	Definition
Funder-driven	Research software funders take into account strategic priorities, perceived research community needs and other factors to determine the guidelines and priorities for funding. Allocation decision-making may be implemented through formal peer review processes, expert panels, or funder representatives.
User-driven	Researchers' needs are expressed through their usage of specific software, and this drives funding allocation decisions.
Developer-driven	Software developers analyze the needs of the research community, usage patterns, software needs, and other variables, and collectively use this to determine where to allocate resources.

At the software project level, the Sustainable Research Software Institute proposal (Watson *et al.*, 2023) proposed a system where similar software projects talk about what they need to improve their sustainability. Then representatives of those groups work together to make an overall allocation of funding that matches what funders provide. The Sustainable Research Software Institute proposal was a predecessor to what is now the CORSA project (https://corsa.center) within the US Department of Energy-funded Consortium for the Advancement of Scientific Software (CASS, https://cass.community/). CASS partially implements this with software projects that were formerly funded by DOE's Exascale Computing Project (ECP, https://www.exascaleproject.org), which have been grouped into CASS projects that share some common discipline, intent, or practices. Each CASS project has an internal process where the software projects within it propose the sustainability funding they need, and the CASS project collectively decides the funding split. The funding given to each CASS project is currently determined by the funder (DOE), but an extension of the current system could allow the CASS projects to decide on this split collectively.

At the user level, funding could be distributed to projects based on their usage by researchers funded by (or of importance to) particular funders. A mechanism such as Drips (Decentralized Research Infrastructure for Pursuing Sustainability, https://www.drips.network/app) could track the usage of software by such research projects, and report to the funders of the projects, who could then use this information to distribute funding to the software developers. The funders would determine how this funding would be allocated, for example, based on the number of their researchers who use the software, the relative importance of the researchers to the funders' goals, or another factor.

No matter what decision-making mechanism is used (or combined), coordination between funders is generally beneficial, as at the least, shared knowledge between funders can lead to better decisions, where funders can make decisions with awareness and in response to those of other funders (e.g., funder A is aware that funder B is supporting project X and so chooses to support project Y instead of also supporting project X). Active shared decision making can be even more efficient, where multiple funders coordinate their decisions (e.g., funders A and B collectively decide that projects X and Y should be funded, and how funding should be most effectively split and distributed).

6. Conclusion

Research software underpins much of contemporary scholarship, yet the funding pillars supporting it are not well-coordinated and aligned to maximize efficiency and impact. Multiple agencies and organizations—governmental, philanthropic, and industry—provide grants with limited coordination, often creating duplicative efforts for both funders and recipients. A central issue lies in the mismatch between software projects' ongoing resource needs and time-limited grant cycles, leading to funding gaps, maintenance bottlenecks, and gaps in support for critical tools. For example, a recent study of international research software funders describes how funding strategies are evolving in response to sustainability and coordination pressures (Jensen & Katz, 2025c).

New frameworks for more efficient, coordinated funding, drawing on cross-organization communication and infrastructure sharing, are emerging in the international research funding landscape. Existing collaborative structures like CHIST-ERA, the Belmont Forum, and philanthropic partnerships from CZI illustrate how shared goals can unify diverse funders and reduce inefficiencies. Pooled resources, co-funded calls, and community-driven allocation can help identify software tools with broad applicability and critical dependencies across research fields. In addition to the models provided above, other useful inputs include the Amsterdam Declaration on Funding Research Software Sustainability (ADORE.software), which aims to raise awareness of the role of funding practice in the sustainability of research software and to improve that practice. The Declaration includes a set of recommendations and an accompanying ADORE.software Toolkit, which have gained some traction with research software funders (Jensen & Katz, 2025a, 2025b).

Several options for funding models and procedures have emerged to distribute resources and make decisions between competing options and priorities. Traditional peer review remains prevalent as an input for funder-driven decisions. Other approaches place decision-making power with users (viz., via usage-based funding allocations) or software developers (e.g., through collective negotiations of funding pools). A mix of processes may be effective to align funding decisions with both real-world needs and strategic considerations.

Groups such as the ReSA-led Research Software Funders Forum and the Global Research Council (GRC) Multilateral Engagement Working Group are also a focus for useful discussions. Bringing philanthropic and governmental funders into collaborative initiatives can help research software projects gain consistent support to evolve from prototype to sustainable infrastructure. Such alignment reduces duplication and strengthens the software ecosystem via more efficient resource allocation, ensuring that tools vital for advancing the frontiers of knowledge are robust, widely accessible, and continuously improved.

Finally, as stated by reviewers of the first version of this paper and audience members where we have spoken, the incentive system in research performing institutions remains an underlying issue. Multiple groups are working to address this, including Coalition for Advancing Research Assessment (CoARA) in its Agreement on Reforming Research Assessment published in July 2022 (Arentoft *et al.*, 2022), as well as the ReSA/RDA Policies in Research Organisations for Research Software (PRO4RS) working group (Barker *et al.*, 2025).

Ethics and consent

Ethical approval and consent were not required.

Disclaimer

The views expressed in this article are those of the authors. Publication in Open Research Europe does not imply endorsement by the Alfred P. Sloan Foundation or the European Commission.

Data availability

All materials underlying the article are contained within the text.

References

Arentoft M, Berghmans S, Borrell-Damian L, *et al.*: **Agreement on Reforming Research Assessment.** *Zenodo.* 2022.
Publisher Full Text

Barker M, Cohen J, Serrano PH, *et al.*: **Institutional Policy Pathways for Supporting Research Software: Global Trends and Local Practices.** *J Open Res Softw.* 2025; **13**(1).
Publisher Full Text

Belmont Forum: About. 2017; Accessed 24 March 2025.
Reference Source

CHIST-ERA: **Projects of call ORD**. 2024; Accessed 24 March 2025.
Reference Source

Chue Hong NP, Katz DS, Barker M, *et al.*: **FAIR Principles for Research Software (FAIR4RS Principles) (1.0).** *Zenodo.* 2022.
Publisher Full Text

CZI: How open source software is transforming biomedicine. 2024; Accessed 24 March 2025.
Reference Source

European Commission: Fundamentals of software engineering. 2022; Accessed 24 March 2025.
Reference Source

EVERSE: Three-tier model. 2024; Accessed 24 March 2025.
Reference Source

Experiment: Low-cost tools for science. 2023; Accessed 24 March 2025.
Reference Source

Gruenpeter M, Katz DS, Lamprecht AL, *et al.*: **Defining Research Software: a controversial discussion (Version 1).** *Zenodo.* 2021.
Publisher Full Text

Hasselbring W, Druskat S, Bernoth J, *et al.*: **Multi-Dimensional Research Software Categorization.** *Comput Sci Eng.* 2025; **27**: 59–68.
Publisher Full Text

Hertweck K, Strasser C, Taraborelli D: **Insights and impact from five cycles of essential open source software for science.** *Zenodo.* 2024;
Publisher Full Text

Hinsen K: **Dealing with software collapse.** *Comput Sci Eng.* 2019; **21**(3): 104–108.
Publisher Full Text

Honeyman T, Treloar A: **A national agenda for research software (0.9).** Zenodo. 2021.
Publisher Full Text

Jensen EA, Katz DS: **Strategic priorities and challenges in research software funding: results from an international survey [version 2; peer review: 2 approved, 2 approved with reservations, 1 not approved].** *F1000Res.* 2025a; **13**: 1447.
Publisher Full Text | **PubMed Abstract** | **Free Full Text**

Jensen EA, Katz DS: **Awareness of FAIR and FAIR4RS among international research software funders.** *Sci Data.* 2025b; **12**(1): 627.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

Jensen EA, Katz DS: **Evolving funding strategies for research software: Insights from an international survey of research funders.** *PLoS One.* 2025c; **20**(8): e0329833.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

Research Software Alliance: Research software funding opportunities. 2025; Accessed 24 March 2025.
Reference Source

Sage Research Methods Community: Concept grants from Sage. 2024; Accessed 24 March 2025.
Reference Source

Schlauch T, Meinel M, Haupt C: **DLR software engineering guidelines (1.0.0).** *Zenodo.* 2018;
Publisher Full Text

Watson GR, Malviya-Thakur A, Katz DS, *et al.*: **An open community-driven model for sustainable research software: sustainable research Software Institute.** ArXiv. 2023.
Publisher Full Text

Yehudi Y, Cashman M, Felderer M, *et al.*: **Research Software Stages.** *ECEASST.* 2025; **85**.
Publisher Full Text

Open Peer Review

Current Peer Review Status: ✔ ? ? ? ✔

Version 2

Reviewer Report 12 May 2026

https://doi.org/10.21956/openreseurope.25049.r74018

© **2026 Ruohonen J.** This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

✔ **Jukka Ruohonen**
University of Southern Denmark, Sønderborg, Denmark

I had eight concerns. All were adequately addressed in the revision. I further acknowledge that the open letter format caused some confusion for the previous review.

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Computer science, software engineering, cyber security, technology policy, European Union, among other things

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Reviewer Report 12 May 2026

https://doi.org/10.21956/openreseurope.25049.r74017

© **2026 Waltman L.** This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

✔ **Ludo Waltman**
Leiden University, Leiden, The Netherlands

No further comments.

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: open science; scholarly communication; peer review; research assessment;

scientometrics

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Version 1

Reviewer Report 26 December 2025

https://doi.org/10.21956/openreseurope.21867.r64286

© 2025 Waltman L. This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Ludo Waltman
[1] Leiden University, Leiden, The Netherlands
[2] Leiden University, Leiden, The Netherlands

In this 'open letter' the authors argue there is a lack of coordination in funding for research software. According to the authors, this lack of coordination leads to inefficiencies, in particular "duplicated effort and suboptimal resource distribution". The authors argue that coordination between funders of research software needs to be strengthened, and they provide examples of ways in which such coordination can be organized.

I don't consider myself to be an expert on the specific issue of funding for research software. However, on the basis of my broader expertise on issues related to research funding and open science, I found the paper to be clear and informative. Based on the information available on the Open Research Europe platform, my understanding is that an 'open letter' should discuss "policies relevant to a broad research community", that it "should usually represent the views of an eligible project or group of projects", and that it "should not include new data and analysis". To me the paper appears to meet the criteria for an 'open letter'. I don't have any specific suggestions for improvements.

Is the rationale for the Open Letter provided in sufficient detail? (Please consider whether existing challenges in the field are outlined clearly and whether the purpose of the letter is explained)
Yes

Does the article adequately reference differing views and opinions?
Yes

Are all factual statements correct, and are statements and arguments made adequately supported by citations?

Yes

Is the Open Letter written in accessible language? (Please consider whether all subject-specific terms, concepts and abbreviations are explained)
Yes

Where applicable, are recommendations and next steps explained clearly for others to follow? (Please consider whether others in the research community would be able to implement guidelines or recommendations and/or constructively engage in the debate)
Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: open science; scholarly communication; peer review; research assessment; scientometrics

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Author Response 10 Feb 2026

Eric Jensen

Dear Ludo Waltman, Thank you for taking the time to review this and for providing your positive feedback. We hope you will view the changes we have made in response to other reviewers as making the paper stronger.

Competing Interests: No competing interests were disclosed.

Reviewer Report 18 December 2025

https://doi.org/10.21956/openreseurope.21867.r64287

© 2025 Kamaludin M. This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

?

Mahirah Kamaludin
[1] Universiti Malaysia Terengganu, Kuala Terengganu, Terengganu, Malaysia
[2] Universiti Malaysia Terengganu, Kuala Terengganu, Terengganu, Malaysia

This Open Letter provides a clear and timely overview of the fragmented landscape of research software funding and makes a persuasive case that better coordination among funders could improve sustainability and efficiency. The article successfully maps existing funder types, funding mechanisms, and collaborative models, using international examples that help readers understand how joint decision-making and pooled approaches can work in practice. The

manuscript is accessible, logically structured, and well suited to its intended policy-oriented audience. It is particularly strong as a primer for both software developers and funders who are unfamiliar with the current funding ecosystem, and it appropriately highlights the growing role of international coordination efforts in strengthening research software as critical infrastructure.

However, to improve the manuscript, the authors should strengthen the evidential basis of their main claims and better align the abstract, body, and conclusion. Several key statements about inefficiencies, project dependency, and the benefits of coordination require clearer empirical support or more explicit framing as informed observations rather than established facts. The definitions of “research software” and the proposed software stages would benefit from tighter consistency and brief justification of how these categories were constructed. The conclusion should be more specific about how proposed coordination and community-driven models would operate in practice, and should acknowledge structural issues beyond funding, such as academic incentive systems and the high long-term costs of software maintenance. Adding short clarifying sentences on these points would significantly strengthen the manuscript’s coherence and policy relevance.

Is the rationale for the Open Letter provided in sufficient detail? (Please consider whether existing challenges in the field are outlined clearly and whether the purpose of the letter is explained)
Yes

Does the article adequately reference differing views and opinions?
Partly

Are all factual statements correct, and are statements and arguments made adequately supported by citations?
Yes

Is the Open Letter written in accessible language? (Please consider whether all subject-specific terms, concepts and abbreviations are explained)
Yes

Where applicable, are recommendations and next steps explained clearly for others to follow? (Please consider whether others in the research community would be able to implement guidelines or recommendations and/or constructively engage in the debate)
Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: -

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 10 Feb 2026

Eric Jensen

Response to Reviewer 4 Dear Mahirah Kamaludin, Thank you for taking the time to review this and for providing your feedback. Here, we respond to your points that bring up issues, including a brief summary of each point: *Original point: ... to improve the manuscript, the authors should strengthen the evidential basis of their main claims and better align the abstract, body, and conclusion. Several key statements about inefficiencies, project dependency, and the benefits of coordination require clearer empirical support or more explicit framing as informed observations rather than established facts.* This open letter (which is not a research paper) is based in part on our experiences (as funders and in discussions with funders while running events for research funders such as the ReSA Funders Forum and its three conferences). We've added text in Section 2.2 to make this clearer, which we feel is the primary place that contains our own opinions, unsupported by external references. We've also added a few words at the start of Section 5 to make clear that this section is a proposed model. *Original point: The definitions of "research software" and the proposed software stages would benefit from tighter consistency and brief justification of how these categories were constructed.* The definition of research software is external and based on a previous community discussion, so we view it as more or less fixed at this time. We have added a sentence at the end of the software stages part of Section 3 to better contextualize our proposed stages. *Original point: The conclusion should be more specific about how proposed coordination and community-driven models would operate in practice, and should acknowledge structural issues beyond funding, such as academic incentive systems and the high long-term costs of software maintenance.* We don't feel that we are ready to make specific recommendations, but rather, that this is the point where more discussion is needed, which we hope to have contributed to. Ideally, this discussion will lead to further research and piloting, and then recommendations from ourselves and others. We have added a paragraph at the end of the conclusions that discusses the academic incentive system as a concern. Additionally, we updated the Yehudi et al. reference from a poster to a paper that was published after our original paper was submitted.

Competing Interests: No competing interests were disclosed.

Reviewer Report 18 December 2025

https://doi.org/10.21956/openreseurope.21867.r64289

© 2025 Canterino F. This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

?

Filomena Canterino
[1] Polytechnic University of Milan, Milan, Italy
[2] Polytechnic University of Milan, Milan, Italy

This Open Letter addresses the fragmented landscape of research software funding and advocates for stronger coordination among European and international research funders. The authors argue that uncoordinated funding mechanisms create inefficiencies for software developers, funders, and the broader research ecosystem. The letter provides an overview of existing grant-funding programs across governmental, philanthropic, and industry sectors to illustrate the variety of current approaches. It also presents examples of existing coordination models among funding organizations, highlighting both centralised and distributed methods for collaborative decision-making. Finally, the authors propose three options to democratize funding decisions, moving beyond traditional funder-driven models to include user-driven and developer-driven mechanisms as alternative pathways for allocating resources more equitably and effectively.
I believe this piece is relevant and timely, and it offers useful insights for the research community. With the intent of providing constructive feedback for improvement, below I provide detailed comments about the evaluation criteria expressed above

1. The rationale is clearly stated and appropriate for the Open Letter format. The authors articulate the challenges arising from fragmented and uncoordinated research software funding and convincingly explain why improved coordination is needed. The motivation is grounded in concrete challenges faced by software maintainers, funders, and the broader research system.

2. The letter presents a strong case for coordinated funding. A suggestion could be to expand a bit alternative viewpoints. For example, authors could include explicit discussion of potential reservations regarding coordination (e.g., autonomy of national funders, risk of over-centralisation, disciplinary differences), or they could reference existing debates from research infrastructure policy literature about collaborative governance.

3. The descriptions of funding programs and collaborative mechanisms are generally accurate, but I suggest to support general claims about inefficiencies, funding gaps, or the scarcity of research-software-specific programs with explicit references (in the text) to studies, reports, or community surveys.

4. The letter is readable and clear for audiences familiar with research software and research infrastructure, but accessibility can be further improved. Here two specific points of attention: (i) define specialised terms (e.g., FAIR principles, pooled funding, research infrastructure), (ii) Ensure all acronyms are defined at first use (e.g., CASS, ECP, DRIPS); (iii) break up long paragraphs in Sections 3 and 4 to enhance readability.

5. The letter offers valuable insights into possible coordination mechanisms but would benefit from more explicit, actionable recommendations. For example, authors could: discuss more explicit a recommendations section, outlining concrete steps funders could implement (e.g., shared review frameworks, exploratory coordinated calls, data-sharing mechanisms for software usage); provide examples of how European funders or research organisations could engage with the proposed user-driven and developer-driven models.

Is the rationale for the Open Letter provided in sufficient detail? (Please consider whether existing challenges in the field are outlined clearly and whether the purpose of the letter is explained)

Yes

Does the article adequately reference differing views and opinions?
Yes

Are all factual statements correct, and are statements and arguments made adequately supported by citations?
Partly

Is the Open Letter written in accessible language? (Please consider whether all subject-specific terms, concepts and abbreviations are explained)
Partly

Where applicable, are recommendations and next steps explained clearly for others to follow? (Please consider whether others in the research community would be able to implement guidelines or recommendations and/or constructively engage in the debate)
Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Collaborative Research; research ecosystems

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 10 Feb 2026

Eric Jensen

Dear Filomena Canterino,
Thank you for taking the time to review this and for providing your feedback.
Here, we respond to your points that bring up issues, including a brief summary of each point:
Original point 2: The letter presents a strong case for coordinated funding. A suggestion could be to expand a bit alternative viewpoints. For example, authors could include explicit discussion of potential reservations regarding coordination (e.g., autonomy of national funders, risk of over-centralisation, disciplinary differences), or they could reference existing debates from research infrastructure policy literature about collaborative governance. We believe that the second set of points in Section 2.2 (the challenges) sufficiently discuss these reservations.

Original point 3: The descriptions of funding programs and collaborative mechanisms are generally accurate, but I suggest to support general claims about inefficiencies, funding gaps, or the scarcity of research-software-specific programs with explicit references (in the text) to studies, reports, or community surveys. We agree with this suggestion in theory and thought about it when writing the original draft, but were unable to find such

references.

Original point 4: Here two specific points of attention: (i) define specialised terms (e.g., FAIR principles, pooled funding, research infrastructure), (ii) Ensure all acronyms are defined at first use (e.g., CASS, ECP, DRIPS); (iii) break up long paragraphs in Sections 3 and 4 to enhance readability. We have made changes in the text in response, though we note that some of the identified acronyms were already defined when first used.

Original point 5: The letter offers valuable insights into possible coordination mechanisms but would benefit from more explicit, actionable recommendations. ... We don't feel that we are ready to make such recommendations, but rather, that this is the point where more discussion is needed between a range of relevant parties, which we hope to contribute to. Ideally, this discussion will lead to future research and then recommendations from ourselves and others. In the meantime, we have added references to relevant prior empirical research with funders. Additionally, we updated the Yehudi et al. reference from a poster to a paper that was published after our original paper was submitted.

Competing Interests: No competing interests were disclosed.

Reviewer Report 22 August 2025

https://doi.org/10.21956/openreseurope.21867.r57224

© 2025 Skinner K. This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

?

Katherine Skinner
[1] invest in open infrastructure, USA, USA
[2] invest in open infrastructure, USA, USA

I admit not fully understanding the Open Letter format and its norms; I've tried to rely on the description provided (https://open-research-europe.ec.europa.eu/for-authors/article-guidelines/social-sciences/preparing-an-open-letter/) in order to fairly evaluate this writing. The Open Letter format is intended for, "discussing policies relevant to a broad research community, presenting guidelines or white papers, or announcing new initiatives. An Open Letter should usually represent the views of an eligible project or group of projects; publication does not imply endorsement by the European Commission. An Open Letter should not include new data and analysis..."

Based on this description, I see this Open Letter as a limited review of current public/private funder practices specifically in the funding area of research software.

I see value in the general topic of the Open Letter, which provides a brief, provocative, position around research software public and private funding opportunities, stating that there is a need for

more coordination and shared systems for grant makers to cure what it diagnoses as "inefficiencies for research software projects seeking funding, the funding organizations, and the overall research software field.". The letter provides little by way of evidence of concrete benefit that would come from such coordination, though; it also provides little evidence of explicit harm that results from the lack of coordination.

The authors' hypothesis and methods were not provided as far as I could tell (which may be common for an Open Letter format).

The letter makes a few steps that I question below, but overall, it is a good review of some of the practices, challenges, and barriers in a particular type of grant-making. It is a solid primer for prospective research software grantees without a clear idea of how these funding opportunities work or where they are; it also provides a good overview for funders in this discipline/topical/functional area on some of the practices and known issues in private/public grant-making as they are experienced from a grantee's point of view.

Other than a few specific questions I raise below, this seems to be a wholly appropriate Open Letter, with clear recommendations and requests that the target audience should understand.

A few specific things that I'll call attention to:

1. The Abstract ambitiously claims that "Encouraging both bottom-up user-driven and top-down coordination mechanisms ultimately supports more robust, widely accessible research software, improving global research outcomes." This statement is unsubstantiated (and perhaps not actually explored) by the Open Letter. Supporting it would require different evidence than this Open Letter provides.

2. The Open Letter defines "research software" and briefly describes a set of funder models and programs. It then outlines "Funding challenges" in 2.2, making a number of claims that are not substantiated, including "Research software projects typically begin under a grant...the project needs to find new resources to continue to develop and/or maintain the software when the grant ends." While this is true of certain types of software projects (and the authors do specify that they are focused on research software that is developed with the intent that others beyond the developer(s) use the software), there are many other ways that research software projects begin, including as part of student or faculty work that is voluntary or supported by their institutional home or as part of industry work that is voluntary or supported by a commercial entity. Not all of these projects need new resources beyond the developers' volunteerism to continue, though many do.

3. The Funding challenges listed in this section are all believable, but again not well substantiated or given clear context. I wonder if the authors could provide clearer signaling via stating these as challenges they themselves have seen or experienced, challenges that others have shared with them, or challenges documented in the scholarly record? Again that may be asking for more than is appropriate in an Open Letter.

4. I wanted more details on the categories in the ReSA public database of current/past research software-specific funding opportunities (a terrific resource), specifically how these categories were determined and who provides these designations. Of particular interest, the "stage of software

supported" gives a three-stage classification that feels academically driven and might leave out a lot of projects. The specificity of the letter's focus on research software developed with the intention of external use (by someone other than the developer) might give enough context to make these three stages make sense, but I can imagine a lot of exceptions to these stages that could be funded, including use of the software by someone other than the developer.

5. Similarly, I would have loved to see more about how the three main categories of decision-making for research software funding were constructed. In my experience, there are a lot of blended models that do not fit easily into these three categories or their combination; knowing more about who created them and how they fit their purpose within the database would be helpful context.

6. I found the arguments about coordination and governance interesting, but underdeveloped. The conclusion didn't match the abstract or introduction well. Claims were made in the conclusion that were often not clearly substantiated, either by the examples provided in the letter or through citations. Again, this may be a product of the "Open Letter" format.

Is the rationale for the Open Letter provided in sufficient detail? (Please consider whether existing challenges in the field are outlined clearly and whether the purpose of the letter is explained)
Yes

Does the article adequately reference differing views and opinions?
Partly

Are all factual statements correct, and are statements and arguments made adequately supported by citations?
Partly

Is the Open Letter written in accessible language? (Please consider whether all subject-specific terms, concepts and abbreviations are explained)
Yes

Where applicable, are recommendations and next steps explained clearly for others to follow? (Please consider whether others in the research community would be able to implement guidelines or recommendations and/or constructively engage in the debate)
Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Open Infrastructure funding models, Sociology of Business, Sociology of Culture, Business lifecycles, Nonprofit management, etc.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 10 Feb 2026

Eric Jensen

Dear Katherine Skinner,
Thank you for taking the time to review this and for providing your feedback.
Here, we respond to your general and numbered points, including a brief summary of each point:
A. General point: The letter provides little by way of evidence of concrete benefit that would come from such coordination, though; it also provides little evidence of explicit harm that results from the lack of coordination. We agree with this; however, we do not have such evidence to prove the point, but we feel the issue will be accepted by almost all developers who have sought grants, and we also believe that many funders have discussed this. Both our experiences (as funders and in discussions with funders while running in ReSA Funders Forum and its three conferences) and the few coordinated programs that exist are evidence that those funders agree with this point. Also see response point 3 below.

1. Original point: the open letter claims that "Encouraging both bottom-up user-driven and top-down coordination mechanisms ultimately supports more robust, widely accessible research software, improving global research outcomes." This statement is unsubstantiated (and perhaps not actually explored) by the Open Letter. Supporting it would require different evidence than this Open Letter provides. We agree, but also believe that the point of the open letter is to propose this idea, not to demonstrate or prove it, which could be the aim of a future research paper.

2. Original point: some reusable research software does not need grant funding and can be supported by volunteers. We agree with this point, but our experience indicates that this is somewhat rare for the type of software we are talking about. However, we've added a sentence in the first paragraph of Section 2.1 to make this situation explicit.

3. Original point: The Funding challenges listed in this section are all believable, but again not well substantiated or given clear context. I wonder if the authors could provide clearer signaling via stating these as challenges they themselves have seen or experienced, challenges that others have shared with them, or challenges documented in the scholarly record? In response to this reasonable comment, we've added text before and after the second list in Section 2.2, and also point out that the first list is referenced by a funder (CZI, 2024), which is the source of those questions.

4. Original point: I wanted more details on the categories in the ReSA public database of current/past research software-specific funding opportunities (a terrific resource), specifically how these categories were determined and who provides these designations. Of particular interest, the "stage of software supported" gives a three-stage classification that feels academically driven and might leave out a lot of projects. ... The categories were determined by the authors based on our own analysis of the database contents. While we would welcome further work by others in this area, which might include different categories, we believe these categories are a reasonable representation, and where we are aware of other work (such as the stages of research software), we've attempted to explain our choices in that context.

5. Original point: Similarly, I would have loved to see more about how the three main categories of decision-making for research software funding were constructed. In my experience, there are a lot of blended models that do not fit easily into these three categories or their combination; knowing more about who created them and how they fit their purpose within the database would be helpful context. We agree that blended models are possible, but our point was more to say that funder-directed (top-down) decisions were not the only option and to suggest alternatives. In our experience as funders and based on the funders we have worked with, other options are unlikely to have been considered and do not appear to have been used, except in rare experiments.

6. Original point: I found the arguments about coordination and governance interesting, but underdeveloped. The conclusion didn't match the abstract or introduction well. Claims were made in the conclusion that were often not clearly substantiated, either by the examples provided in the letter or through citations. Again, this may be a product of the "Open Letter" format. As suggested by the reviewer, this is an open letter, not an evidence-based research paper, intended to open discussion that could lead to future empirical research. However, we have added references to prior relevant empirical research with research software funders. Additionally, we updated the Yehudi et al. reference from a poster to a paper that was published after our original paper was submitted.

Competing Interests: No competing interests were disclosed.

Reviewer Report 14 August 2025

https://doi.org/10.21956/openreseurope.21867.r57225

© 2025 Ruohonen J. This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

?

Jukka Ruohonen
[1] University of Southern Denmark, Sønderborg, Denmark
[2] University of Southern Denmark, Sønderborg, Denmark

I applaud the authors for addressing the topic of funding for research software. The topic is important and extends well-beyond research software. Nevertheless, regarding criticism, my concerns are about a lack of research questions, the definitions used and framing done, and the general motivation. Thus, in no particular order of importance:

1. The questions presented in Section 2.2 are relevant but do not match exactly what is presented. Which one of these is examined in the paper specifically? Regarding the subsequent list of challenges, I suppose alignment is what the conclusion is about. If so, it should be made explicit in the concluding section. A clarification is needed also whether or not the other challenges are still present too.

2. The motivation should include a paragraph about software maintenance itself, which is costly. Therefore, most software projects have a fixed life cycle. It might take months, years, or decades, but most software is not meant to live forever. In many cases, once a software is released, it is not really an asset but a cost, as also concepts such as technical debt demonstrate. For instance, security updates are needed, and also for research software [1]. Against these backdrops, I would like to see a stronger argument somewhere, perhaps in the concluding section, about why software maintenance itself has supposedly not been a target for funding (?). While grant-based funding may certainly be a factor, maybe it is also because software maintenance fits poorly into the whole paradigm of research, development, and innovation?

3. Related to the previous point and regarding the software stages presented in Section 3, it would be relevant to have a cross-tabulation of these and the funding bodies.

4. The definition for research software in the beginning of Section 2 does not match well the software stages described in Section 3. By a reasonably well-grounded hypothesis, the overwhelming majority of research software, as per the definition provided, is about software needed to compute results for a single paper or for multiple papers in a research project. Practically all papers dealing with quantitative data nowadays contain such software. Yet, such software is not really even about prototypes (Stage 1). Nor is such software even meant to be maintainable. On one hand, the situation is unfortunate because it affects reproducibility and direct replications of existing results. On the other hand, it is also unreasonable to demand scientists to maintain such software. For better or worse, also scientific papers have a fixed live cycle in this regard.

5. Although SciPy (but not NumPy, PyTorch, and others) is mentioned, I am surprised to see no mention of R, which is the other *de facto* programming language for statistical computing in particular.

6. Regarding the stages, the categorization presented seems to lack the fundamental infrastructures used to develop, maintain, and distribute research software. Among these are PyPI for Python and CRAN for R. Furthermore, these language-specific repositories are closely tied to "native" package managers used in Linux distributions and other operating systems. Thus, all in all, the infrastructures are not only about building foundational systems. This point correlates with well known security issues currently plaguing open source software repositories.

7. Regarding the conclusion, I agree that better coordination and governance are needed. However, I would also like to see a sharper reflection about what the authors specifically have in mind regarding improvements. Although community-driven allocation is mentioned, how exactly would it work? There are also risks involved in using community-based metrics (cf. [2]), as some funding bodies for non-scientific open source software have done in the past. Given the grant-based funding and prioritization difficulties, perhaps it would make sense to have also larger software foundations involved in strategic consultations or even funding decisions? After all, they usually have a good bird's-eye view on their respective ecosystems.

8. Finally, I would also like to see a remark in the concluding section that the issues with research software are not only about funding but also related to the incentive systems in academia. Strengthening the creation *and* maintenance of research software in academic reward systems

would also help at tackling the issues. The recent Agreement on Reforming Research Assessment (CoARA) has already taken steps toward this direction.

[1] https://arxiv.org/pdf/2203.06502
[2] https://arxiv.org/abs/2412.13459

Is the rationale for the Open Letter provided in sufficient detail? (Please consider whether existing challenges in the field are outlined clearly and whether the purpose of the letter is explained)
Yes

Does the article adequately reference differing views and opinions?
Partly

Are all factual statements correct, and are statements and arguments made adequately supported by citations?
Yes

Is the Open Letter written in accessible language? (Please consider whether all subject-specific terms, concepts and abbreviations are explained)
Yes

Where applicable, are recommendations and next steps explained clearly for others to follow? (Please consider whether others in the research community would be able to implement guidelines or recommendations and/or constructively engage in the debate)
Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Computer science, software engineering, cyber security, technology policy, European Union, among other things

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 10 Feb 2026

Eric Jensen

Dear Jukka Ruohonen,
Thank you for taking the time to review this and for providing your feedback. Here, we respond to your numbered points below (labelled 'original point'), including a brief summary of each point:

1. Original point: questions in Sec. 2.2 are not aligned with the paper's content.

This is not a research paper, and these are not research questions we are addressing. Rather, they are motivation and context for the discussion of existing programs in Sec. 3, the existing coordination of programs in Sec. 4, and the discussion and suggestions in Sec. 5.

2. Original point: motivation should include a paragraph about software maintenance itself, which is costly. ... I would like to see a stronger argument somewhere, perhaps in the concluding section, about why software maintenance itself has supposedly not been a target for funding.
We've added a new paragraph in the introduction to Section 2 in response to this good point.

3. Original point: Related to the previous point and regarding the software stages presented in Section 3, it would be relevant to have a cross-tabulation of these and the funding bodies.
We believe Table 1 does this sufficiently for this paper.

4. Original point: Our focus isn't on the majority of research software, that which is needed to compute results for a single paper or for multiple papers in a research project.
We agree with you that we aren't focusing on the single use software you describe as most common, but on software that is intended for re-use. We believe we have made this point in the second paragraph of Section 2.

5. Original point: R is not mentioned with SciPy.
We've added a mention of R and of tidyverse, but we can never exhaustively mention all software or all programming languages.

6. Original point: package management systems are not discussed in the stages as part of infrastructure.
While we agree that package managers are a key component of software and programming today, they are not research software, which is where we are focusing. To clarify this, we've added the word research just before our list of stages.

7a. Original point: The review would like more about what we specifically have in mind regarding improvements.
We don't feel this is needed in this open letter, but agree that it would be useful in a more detailed article, perhaps to be written later.

7b. Original point: It would make sense to have larger software foundations involved in strategic consultations or even funding decisions.
We agree, and have slightly altered the third paragraph of the conclusions to try to make this more clear.

8. Original point: The issues with research software are not only about funding but also related to the incentive systems in academia. Strengthening the creation and maintenance of research software in academic reward systems would also help at tackling the issues.
We agree with this point, and have added a new final paragraph in the conclusions. Additionally, we updated the Yehudi et al. reference from a poster to a paper that was published after our original paper was submitted.

Competing Interests: No competing interests were disclosed.