

Research Software Discovery: How do we Want to Search Research Software and Where do we Want to Find it?

A deRSE25 workshop report

Ronny Gey¹, Oliver Karras², Daniel Mietchen³, Alexander Struck⁴, and Tim Wittenborg⁵

¹Helmholtz Center for Environmental Research - UFZ

²TIB – Leibniz Information Centre for Science and Technology and University Library

³FIZ Karlsruhe — Leibniz Institute for Information Infrastructure

⁴Cluster of Excellence Matters of Activity. Image Space Material at Humboldt-Universität zu Berlin

⁵L3S Research Center

The workshop aimed to identify current challenges in research software discovery and propose actionable improvements. Using a World Café discussion format, participants explored topics such as the effectiveness of current discovery systems, barriers to software discovery, the role of social networks and professional communities, and a vision of the future of software discovery. The workshop generated concrete recommendations for enhancing research software discovery, benefiting developers, researchers, and institutions.

1 Introduction

The discovery of research software is a critical component of modern scientific research. However, the existing landscape for finding, classifying, and hosting research software is fragmented and lacks comprehensive solutions [Gey+25]. This workshop aimed to bring together researchers, developers, and infrastructure providers to discuss the challenges and propose improvements. On 27th of February 2025, 14 participants collaborated for two hours towards the objectives described in the following chapter.

The workshop was conducted using a **World Café** format, allowing for structured yet flexible discussions. Participants were divided into small groups, each addressing a specific topic related to research software discovery. Discussions were summarized and presented to all attendees for further refinement. The schedule is depicted in table 1, discussion topics in table 2.

15 min	Introduction
5 min	Grouping
40 min	World Café (4 × 10 min)
20 min	Break (Time for table chairs to sum up the discussions)
30 min	Presentation plus open discussion for each table/topic (5-8 min each)
20 min	Outro (what are we going to do with the results)






Table 1: Schedule of the world cafe discussions

#	Topic	Description
1.	Effectiveness of Current Discovery Systems	Evaluating the strengths and weaknesses of existing platforms and tools.
2.	User Experience and Accessibility	Identifying barriers and ways to make discovery systems more user-friendly.
3.	Collaborative Networks and Social Platforms	Leveraging social networks and professional communities for software discovery.
4.	Envisioning the Future	How would you like to discover relevant software and evaluate its trustworthiness and usefulness?

Table 2: Discussion topics of the world cafe

2 Workshop Participants

Of the 16 participants, the following have agreed to be listed by name:

Andrew Webb, Camilla Lummerzheim , Gerasimos Chourdakis , Jean-Noël Grad , Julian Gethmann , Safial Islam Ayon .

The selected topics and discussion results are described in section 3. We then synthesize the topics in a result discussion in section 4

3 Results

3.1 Effectiveness of Current Discovery Systems

Participants were unaware of dedicated software discovery solutions and usually turn to general purpose search engines. But these platforms lack information that could be used to evaluate results. Researchers would appreciate advice and context to evaluate software search results. This is one of the key messages that emerged from the table. Regarding LLMs, it was mentioned that they do at times provide relevant results, but they should also provide help in evaluating them. General purpose search engines tend to provide relevant results, but do not rank the results relevant for software. Recall is usually high and users certainly need evaluation assistance there. Evaluation indicators that would be desirable are, for example, size of community, last commit, GitHub stars, citations of software/paper, or FAIRness indicator of software. As an alternative to general purpose search engines, the social network (e.g. colleagues) was often mentioned, which provides more trustworthy references based on relevant experience - advice and context that other discovery approaches often miss.

Many participants referred to package managers as sufficient discovery tools, but these are limited by the domain or application context. They can act as a comprehensive software directory for the language or domain (for Python e.g. [PyPi](#) and [Conda](#); for R [CRAN](#) and [bioconductor](#)) and provide extensive metadata and the link to the corresponding software repository of the packaged software.

During the analysis of existing discovery systems, the desire for a “Software-focused Google Scholar” was also expressed. Users want a search engine specialized in software. Since a lot of software is developed and made available on GitHub, its restricted search functionality was also discussed. GitHub search is used but users report irrelevant results because search functionality lacks relevant thematic filters like ‘limit search to README.md’ or on the functionalities that a software offers.

Software reviews are relevant to certain areas, especially papers and blog posts, but their coverage is often incomplete. Conference and lightning talks were mentioned but not further evaluated. Stickers were mentioned as a marketing strategy that led to relevant discoveries for some participants.

- [PyPi](#): is a good system to search, but you need to know it and how to use it
- Keywords are often insufficient for a search
- It is often difficult to understand a project by looking only at the repository

However, some **positive experiences** with different services were also reported:

- [Betty’s Research Engine](#) to collect metadata based on keyword search
- Humans as a “discovery system” are a good source as they give you often correct and detailed information
- [PyPi](#): is a good system to search
- Use of ChaptGPT is helpful, by asking what software do I need for my problem. You can e.g. do same kind of Tool “Chain-of-Thought” by stepwise investigating if one tool fits or you need to look at the next suggestion

In addition to the experiences, **general suggestions** were also made as to how discovery systems can be made more user-friendly:

- Software Archive/Directory and Community should be combined in one platform
- Integration in one-system with a programmable API, probably REST-API
- “Google Software” as a search platform for software similar to Google Scholar with overviews of statistics, e.g., usage, stars, etc. An integration in mainstream tools such as Google or Google Scholar should be also possible.
- Tool for analyzing a software project, e.g., on GitHub, to propose similar software projects

Furthermore, **improvements** to the search and the search process were suggested to make the discovery systems more user-friendly:

- Consideration of GitHub Tags for better search results
- Assessment of FAIRness, Developement status, Reusability, Quality and Documentation of a software
- Better result order independent from the aspect of being the “most prominent” page on a topic as done by Google
- Description of the purpose of a software to understand for what it can be used
- Use CodeMeta for describing a software
- Support filtering by Domain (as tools can share the same name but are for different domains only usable), Number of dependencies, Number of commits, Number of unit tests, Number of citing papers, Inputs/Outputs for specific workflows, Licence



Figure 2: Results for Table 2: User Experience and Accessibility

3.3 Collaborative Networks and Social Platforms

The conversations were mainly shaped by the places where social interactions happen in which a search for research software can be triggered. Here, a distinction could be made between **physical** and **virtual places**. In **physical places**, the following were mentioned: conferences, the library, lunch or coffee breaks, while workshops, lectures both as participant or host, as well as during professional advice on topics of research data management and software use.

In **virtual places**, on the other hand, social networks were mentioned, with the scientific community on Twitter/X being particularly mourned. In the meantime, however, there are also important alternatives here (Mastodon, BlueSky). Scientific communities also exchange ideas and information on LinkedIn or ResearchGate, where it is also possible to get advice on research software. The Slack instance of the international RSE community was mentioned, as was the [HPC Marketplace](#), a social media platform for HPC practitioners, institutions, and infrastructure providers.

Wikipedia lists on software, institutional newsletters (SW of the month¹), or software reviews (as a systematic analysis of available software with a certain functionality, similar to systematic literature reviews [KDJ04]), which were also mentioned in the discussions, are not social networks in the narrower sense. Both Wikipedia lists and institutional newsletters can be understood as the output of underlying social networks (Wikipedia, research institution). Although there is no direct connection to social networks, a systematic software review is an extremely exciting idea and will hopefully be taken up soon.

In addition to the places for social interactions, **success factors** that a social network needs in order to be relevant for a search for software were also reported. Fundamentally, a critical mass of participants is required for interactions to take place. This differs from the topics around which a social network is grouped. It was mentioned that the interdisciplinarity of the participants is conducive to an exchange. Accessibility was also briefly addressed in the discussions, insofar as people feel excluded, for example, if their language level is not sufficient to be able to follow discussions at the lunch table, etc., which indirectly also inhibits inquiries about software topics or the search for software.

In the discussions, another small thematic cluster developed around the topic of seeking,

¹e.g. JuRSE Code of the Month: <https://www.fz-juelich.de/en/rse/community-initiatives/jurse-code-of-the-month>

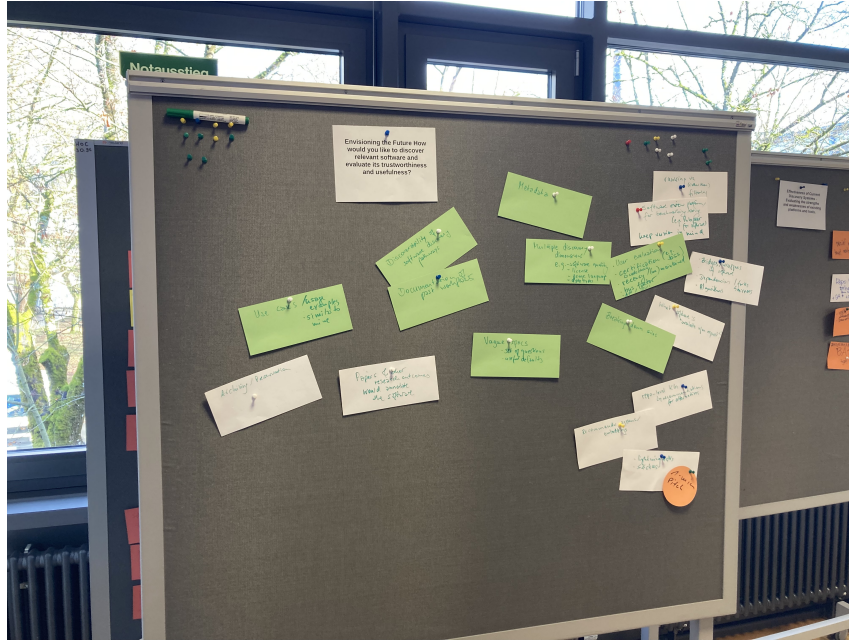


Figure 4: Results for Table 4: Envisioning the Future

- dependencies / forks/ mirrors
- data types (in/ out/ internal)
- filtering is not always preferable - sometimes ranking is, sometimes a combination
- use cases/ usage examples, ideally including some that are “similar to mine”
- striking a balance between standard metadata on the one hand and overly vague or overly precise search criteria on the user or system’s side - useful defaults
- algorithms used

Knowledge graphs were also discussed. These can cover any of the discovery dimensions mentioned, but can also be specific to a single software repository or any other resource. In the future, discovery systems should also integrate **recommendation systems** that contain recommendations for alternatives to software with similar functionality, for example. **Breaking down infrastructure silos** was also discussed. Many such silos exist in the form of institutional, national or domain-specific initiatives, but these in turn only cover the respective framework and sometimes restrict access. A plea was made for an open and globally accessible infrastructure, in general and for software discovery in particular. Another important point for the participants is to **annotate papers** and other research materials with respect to which and how this software was involved.

4 Discussion of the Results

After describing the results of the individual thematic tables of our World Café in section 3, we will now look at four different thematic clusters that emerged from the results.

4.1 Advice needed

Researchers like to rely on the advice of trustworthy third parties when searching for research software. To evaluate the search results provided by discovery systems, they want recommender systems that are integrated into the discovery system itself. Based

on the functionality of the software found or other criteria, these recommenders suggest alternatives with similar criteria features. To use these recommender systems efficiently, it is essential to capture further contextual features of the software (see section 4.2) and to annotate this information with the software in the form of metadata (see section 4.3).

A researcher’s social networks (colleagues, friends, digital communities) were frequently mentioned as a possible source for searching for research software. Social networks often provide trustworthy references based on relevant experience of peers. Even if humans as a “discovery system” are often difficult to reach or access – they are not ubiquitous like their digital alternatives, the collaborative information seeking strategy [Sha13] is worth the effort.

If we explore the reasons for this need for advice among researchers looking for research software, we find that software search is not a science, even among scientists [HG18]. This could be due to the fragmented state of available resources for research software discovery [HMY17] as well as cognitive barriers in information behavior [Sav15] and should be investigated in more detail in the future.

4.2 Gimme context

The key takeaway of the discussions around the effectiveness of current discovery systems (see section 3.1) was that researchers would appreciate advice *and* context to evaluate software search results. It is precisely because certain contextual factors of a research software are not sufficiently communicated that researchers often resort to advice and, above all, seek the advice of their social networks. They provide more trustworthy references based on relevant experience - advice and context - that other discovery approaches often miss. However, context can also be provided by specifying elaborate metadata.

4.3 Filters and Metadata

A major limitation of existing research software directories is the lack of standardized metadata and classification systems. The discussions focused on:

- The need for fine-grained functional classifications of software.
- Implementing richer metadata to improve searchability (In section 3.4 we listed various criteria of metadata that would enhance the research software discovery process.)
- Leveraging AI to a) generate metadata from software documentation and usage patterns and b) to augment the information retrieval process in general (e.g. Retrieval-Augmented Generation (RAG))

A higher quality of existing metadata and a higher quantity of metadata in turn means that a better search can be offered, as there would be more filter options with which the search results could be contextualized and reduced.

4.4 No Place to Find Them All

One of the key challenges identified was the lack of a single, unified platform where researchers can find all relevant software. Current solutions include publication repositories, institutional catalogs, curated lists, and general search engines, each with its own limitations. Participants discussed possible solutions such as:

- Creating a federated search system that integrates multiple repositories (Betty’s Research Engine is a good example here).
- Enhancing metadata and interoperability between different platforms.

- Promoting the use of community-driven software registries.
- Creating a research software focused knowledge graph (e.g. use the Wikidata).

To what extent such a unified solution, a place to find them all, is technically feasible and even desirable as a concept, we cannot clarify at this point. In this context, the desire was also expressed to build a community around such a research software discovery solution that would jointly support and further develop the system.

5 Conclusion

The workshop results are limited by the following factors:

- The workshop participants do not represent the entire spectrum and diversity of the target group. The workshop was held at a computer science / RSE conference, so we had a over representation of scientist from the computer science domain.
- We did not double-check the report with the workshop participants, hence we do not know whether we described ideas and comments correctly.

Future work should focus on:

- Developing interoperable standards for software metadata and classification.
- Enhancing existing repositories with AI-powered discovery tools.
- Strengthening collaborations between research institutions, software developers, and indexing platforms.
- Continuing discussions at future conferences and through working groups.

The workshop provided valuable insights into the current state of research software discovery and identified actionable recommendations for improvement. We would like to thank the participants for the engaging discussions throughout the workshop. Also, we would like to say sorry for the noisy situation of the room the workshop took place in. We hope that the workshop results describe your comments and ideas correctly.

References

- [KDJ04] B.A. Kitchenham, Tore Dyba, and M. Jorgensen. “Evidence-based software engineering”. In: *Proceedings of the 26th International Conference on Software Engineering*. May 2004, pp. 273–281. DOI: [10.1109/ICSE.2004.1317449](https://doi.org/10.1109/ICSE.2004.1317449).
- [Sha13] Chirag Shah. “Collaborative information seeking”. In: *Journal of the Association for Information Science and Technology* 65.2 (Nov. 2013), pp. 215–236. DOI: [10.1002/asi.22977](https://doi.org/10.1002/asi.22977).
- [Sav15] Reijo Savolainen. “Cognitive barriers to information seeking: A conceptual analysis”. In: *Journal of Information Science* 41.5 (May 2015), pp. 613–623. DOI: [10.1177/0165551515587850](https://doi.org/10.1177/0165551515587850).
- [HMY17] Katherine Hepworth, Donica Mensing, and Gi Woong Yun. “Journalism Professors’ Information-Seeking Behaviors: Finding Online Tools for Teaching”. In: *Journalism & Mass Communication Educator* 73.3 (July 2017), pp. 255–270. DOI: [10.1177/1077695817718425](https://doi.org/10.1177/1077695817718425).
- [HG18] Michael Hucka and Matthew J. Graham. “Software Search Is Not a Science, Even among Scientists: A Survey of How Scientists and Engineers Find Software”. In: *Journal of Systems and Software*, 141:171-191, 2018 (May 2018). DOI: [10/ggc6j4](https://doi.org/10/ggc6j4).

- [Gey+25] Ronny Gey et al. *Seek and You Shall Find - or Not! Why Can't We Find the Research Software We Really Need?* Feb. 2025. DOI: [10.5281/zenodo.14878608](https://doi.org/10.5281/zenodo.14878608).