BioHackathon series:
BioHackathon Europe 2024
Barcelona, Spain, 2024
Project 5

Submitted: 10 Jun 2025

Published by BioHackrXiv.org

Software Quality Indicators: extraction, categorisation and recommendations from canonical sources

Eva Martin del Pico [1], **Fotis Psomopoulos** [2], **Laura Portell Silva** [1], **Renato Alves** [3], **Sébastien Moretti** [4], **Shoaib Sufi** [5], **Adel Kamel Eddine Bouhraoua** [6], **David Steinberg** [7], **Gavin Farrell** [8], **Maria Tsontaki** [9], **Rafael Andrade Buono** [9], **Sebastian Beier** [10], **Daniel Garijo** [11], **José María Fernández** [1], **Mariia Steeghs-Turchina** [12], **Mihail Anton** [13], **and Magnus Palmblad** [12]

1 Barcelona Supercomputing Center **2** Aristotle University of Thessaloniki **3** European Molecular Biology Laboratory (EMBL) **4** SIB Swiss Institute of Bioinformatics **5** University of Manchester **6** University of Padova **7** University of California, Santa Cruz **8** European Bioinformatics Institute **9** Vlaams Instituut voor Biotechnologie **10** Forschungszentrum Jülich **11** Universidad Politécnica de Madrid **12** Leiden University Medical Center (LUMC) **13** ELIXIR Hub, Chalmers University of Technology

Abstract

Research software plays a central role in modern science, and its quality is increasingly recognized as essential for reproducibility, sustainability, and trust. Numerous initiatives have proposed indicators to guide quality assessment, yet these indicators are dispersed across domains and vary in scope, terminology, and practical use. This work presents a curated catalogue of software quality indicators tailored to the needs of research software. Developed during BioHackathon Europe 2024 and refined in collaboration with the ELIXIR Tools Platform and EVERSE project, the catalogue consolidates and structures indicators from a range of authoritative sources.

Over 300 indicators were gathered and systematically reviewed for relevance, clarity, and implementation feasibility. Each was classified into thematic categories—such as Documentation, Security, Usability, and Sustainability—and annotated with target applicability, ease of evaluation, and recommended actions. Redundant, overly abstract, or narrowly scoped indicators were excluded or flagged, while additional tags highlighted cross-cutting concerns such as licensing, testing, and community practices.

The resulting open dataset, available as a structured spreadsheet, includes detailed metadata and decision criteria to support reuse, adaptation, and extension. The catalogue offers a foundation for context-specific assessment frameworks. Intended users include research software developers and maintainers, evaluators, and developers of quality-focused tools and guidelines.

1. Introduction

The primary goal of this project is to perform a comprehensive extraction,categorization and recommendations of existing indicators related to research software quality, resulting in a curated catalogue that reflects shared understanding and practical usability. This work was carried out as part of BioHackathon Europe 2024 (Project 5), with a focus on supporting the ELIXIR Research Software Ecosystem while aiming for applicability across the broader Life Sciences community.

Assessing the quality of research software remains a recognized challenge, with significant variability in the maturity and adoption of indicators across domains. Our goal was to address this by collecting, comparing, and consolidating existing indicators into a structured format, removing redundancy, and highlighting relevance to common use cases in scientific software evaluation.

The resulting catalogue is designed to:

- Raise awareness of current best practices and services in research software.
- Clarify the requirements and expectations of these services.
- Help users identify the most appropriate quality assessment approach for their specific context.
- Reveal gaps in community-specific practices or support tools.
- Offer a foundation for adopting common indicators across different scientific domains.

Through this work, we aimed to support a shared vocabulary around software quality, enable meaningful comparisons across tools and frameworks, and facilitate a more coherent ecosystem for sustainable, trustworthy research software.

Project Scope and Outcome

Research software plays a vital role in modern scientific work, and assessing its quality is essential for reproducibility, sustainability, and trust. The goal of this project was to create a curated, non-redundant set of indicators that are relevant, feasible to implement, and applicable to the assessment of research software best practices.

Our objectives were to:

- Gather indicators from established resources.
- Remove indicators that are out of scope, redundant, or overly philosophical (too abstract for practical use).
- Assign supergroups and tags to support practical use.
- Produce a usable spreadsheet of indicators annotated with implementation guidance and decisions.

The final outcome is a structured dataset of quality indicators, accompanied by documentation of decision criteria and rationale. While the list does not include prioritization or ranking, it can be adapted for specific use cases. Future users may wish to introduce weighting schemes based on context.

Expected Users

This resource is intended for:

- Research software developers and maintainers.
- Projects (e.g., EVERSE, ELIXIR-STEERS)
- Guides (e.g., RSQKit)
- Anyone seeking to assess research software against quality best practices.

2. Method

2.1 Sources Included

We reviewed indicators from a variety of authoritative sources:

- EOSC Task Force 3
- FLOSS Best Practices Criteria
- RSMD Guidelines
- FAIRIMPACTD5.2–MetricsforAutomatedFAIRSoftwareAssessment
- Helmholtz Quality Indicator
- ISO/IEC 25010
- SWEBOK v4 – Chapter 12: Software Quality

2.2 Sources Excluded

Several sources were not included due to either duplication, minimal added value, or being overly narrow:

- STEERS scoreboard
- Software Citation Workshop 2022 indicators
- Reproducible builds (focused on a single criterion)
- Bioinformatics SE practices (descriptive rather than indicator-focused)

2.3 Workflow

The work was initiated at BioHackathon Europe 2024 (Project 5) and refined through a collaborative review process:

1. **Collection**: Over 300 indicators were gathered from selected sources.

2. **Grouping**: Indicators were assigned to thematic "supergroups" based on their primary focus (e.g., Documentation, Security, Sustainability).

3. **Parallel Evaluation of Each Indicator**:
   For each indicator, three types of review were carried out in parallel:

   - **Target**: We classified whether the indicator applied to software, services, governance, or architecture/design.
   - **Implementation Feasibility**: We assessed how easily a person (rather than a machine) could evaluate whether the indicator was being met. This was not a measure of how well a particular indicator was implemented but of how easy it was to assess whether this indicator was being met, e.g. it was not a measure of how fit for purpose an open license was but a measure of whether it was easy to acertain that there was an open license in place.
   - **Action Decision**: We assigned an action:
     - **Keep**: The indicator was considered relevant and feasible.
     - **Maybe**: The value of the indicator was uncertain or context-dependent.
     - **Discard**: The indicator was redundant, too vague, or philosophical (e.g., "immutable infrastructure").
   - **Tagging**: We also added supplementary tags to help identify indicators relevant to cross-cutting themes such as testing, findability, or community practices.
   - **Redundancy Checks**: We identified indicators that duplicated the intent or content of others — whether within the same supergroup or across different ones — and marked them as redundant. These were either discarded or annotated in the comments for transparency, especially when an alternative indicator was preferred.

4. **Definition Verification**: During the BioHackathon, some indicator descriptions were manually shortened in the spreadsheet to facilitate the review process. As a final step, these descriptions were cross-checked against their original sources. If a shortened version had been used, it was replaced with the literal wording from the source whenever available, to ensure fidelity to the original text.

3. Challenges

- **Volume and Overlap**: 303 indicators were initially gathered. Redundancies and overlaps were frequent.
- **Granularity**: Indicators varied significantly in scope — some were fine-grained metrics, others broad statements.
- **AI Support**: Attempts to use AI to identify redundancies failed. Human curation proved essential.
- **Knowledge Gaps**: The team had to consult documentation and resources in unfamiliar areas.

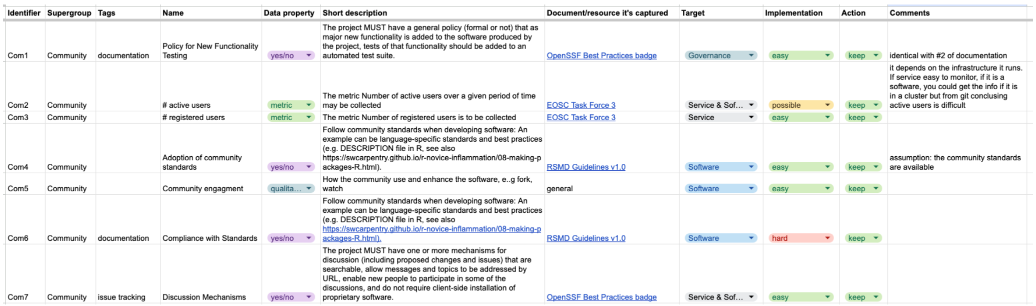
- **Feasibility Gaps**: Few sources provided tools or metrics to measure the indicators directly.
- **Ambiguity**: Some indicators required a "keep if applicable" classification, reflecting contextual relevance.

4. Results

The outcome is a structured spreadsheet available at ZenodoDOI:10.5281/zenodo.14834245, containing:

- A cleaned and annotated list of quality indicators.
- Classification into supergroups and assignment of tags.
- Notes on source, feasibility, and rationale for inclusion or exclusion.
- Internal identifiers (e.g., `Se1`) for reference consistency.

An excerpt of the spreadsheet is shown in **Figure 1**, illustrating how indicators in the *Community* supergroup are categorized and annotated with target, implementation feasibility, and action status.



Identifier	Supergroup	Tags	Name	Data property	Short description	Document/resource it's captured	Target	Implementation	Action	Comments
Com1	Community	documentation	Policy for New Functionality Testing	yes/no	The project MUST have a general policy (formal or not) that as major new functionality is added to the software produced by the project, tests of that functionality should be added to an automated test suite.	OpenSSF Best Practices badge	Governance	easy	keep	identical with #2 of documentation
Com2	Community		# active users	metric	The metric Number of active users over a given period of time may be collected	EOSC Task Force 3	Service & Sof…	possible	keep	it depends on the infrastructure it runs. If service easy to monitor, if it is a software, you could get the info if it is in a cluster but from git conclusing active users is difficult
Com3	Community		# registered users	metric	The metric Number of registered users is to be collected	EOSC Task Force 3	Service	easy	keep	
Com4	Community		Adoption of community standards	yes/no	Follow community standards when developing software: An example can be language-specific standards and best practices (e.g. DESCRIPTION file in R, see also https://swcarpentry.github.io/r-novice-inflammation/08-making-packages-R.html).	RSMD Guidelines v1.0	Software	easy	keep	assumption: the community standards are available
Com5	Community		Community engagment	qualita…	How the community use and enhance the software, e..g fork, watch	general	Software	easy	keep	
Com6	Community	documentation	Compliance with Standards	yes/no	Follow community standards when developing software: An example can be language-specific standards and best practices (e.g. DESCRIPTION file in R, see also https://swcarpentry.github.io/r-novice-inflammation/08-making-packages-R.html).	RSMD Guidelines v1.0	Software	hard	keep	
Com7	Community	issue tracking	Discussion Mechanisms	yes/no	The project MUST have one or more mechanisms for discussion (including proposed changes and issues) that are searchable, allow messages and topics to be addressed by URL, enable new people to participate in some of the discussions, and do not require client-side installation of proprietary software.	OpenSSF Best Practices badge	Service & Sof…	easy	keep	

Figure 1. Example of curated indicators in the 'Community' supergroup, with classification across targets, feasibility, and actions.

Supergroups

To structure the catalogue, indicators were grouped into a set of thematic supergroups. These were defined based on expert intuition and discussions during the BioHackathon, reflecting key dimensions of research software quality. Examples of supergroups include Documentation, Security, Usability, Sustainability, and Licensing. A small number of indicators that lacked sufficient description or were not clearly applicable to research software were excluded and labeled as "X - uncategorised."

Indicator Structure

Each indicator entry includes:

- identifier
  - An internal identifier comprising of a shortened version of the Supergroup name and a number (this is internal added to allow us to reference other identifiers in case there are duplicates or for other reasons) - this also allow sorting of the Supergroup tabs without damaging reference to other rows
- Supergroup
  - Supergroups are a named cluster of indicators on a particular topic (at the time of writing these are: Sustainability, Interoperability, Documentation, Accessibility, Fundability, Community, Security, Usability, Findability, Contribution, Issue tracking)
- Tags
  - These are other Supergroups or areas which this indicator also applies to
- Data type
  - This is the name of the indicator

- Data property
  - These are the types of values the indicator can take
- Short description
  - This is the description of the indicator - this is meant to be verbatim from the 'Document/resource it's capured' from
- Document/resource it's captured
  - The reference document/resource for the definition of the indicator
- Community it's used
  - This is an attempt to capture which communities this indicator tends to be used in
- Person who added this
  - This is the name of the person who captured this indicators - the original extractoin occured during the BioHackathon 2024
- Service/tool that provides this
  - This is the name of a tool or tools that can help you ascertain this indicator
- Target
  - This is what the indicator (aka Data type) applies to (at the time of writing this covers - software, service, governance, service & software)
- Implementation
  - What is the level of difficulty in acertaining the value of this indicator for a given software/service/project (not if it an easy technique to apply but if it's easy to assess whether this is being done)
- Action
  - Whether the BioHackathon25 Project 5 + ELIXIR Tools Platform WP3 (Software Best Practices) folk have said if this indicator should be kept, maybe kept or discarded when assessing software/services/projects - discard's mainly focus on duplicates and where indicators are more philosophical than widely recognised
- Comments
  - The reasons and thoughts around why a particular 'Action' was recommended

Limitations

- **Reusability is not a standalone supergroup**: While reusability is a core principle in software quality (and in frameworks like FAIR), it was not defined as its own supergroup in this catalogue. This is primarily because many aspects of reusability — such as documentation, licensing, and interoperability — are already captured across other groups. As a result, related indicators are distributed throughout the catalogue rather than concentrated under a single heading, which may require additional effort from users specifically focused on reusability.

- **No prioritization or scoring is offered**: The catalogue presents indicators in a neutral and unranked format. This decision was intentional, as relevance and importance can vary significantly across projects, domains, and assessment goals. Users are encouraged to apply their own prioritization schemes depending on context (e.g., compliance, funding, onboarding, sustainability audits).

- **Green computing indicators (e.g., energy efficiency) are not included**: Despite the growing importance of sustainable computing, metrics related to energy consumption, carbon footprint, or other ecological impacts were not included in this iteration. This was due to the lack of indicators in the reviewed sources pertaining to green computing.

6. Acknowledgments

This work was initiated at BioHackathon Europe 2024 (Project 5) and further developed in collaboration with members of the EVERSE WP3 and the broader ELIXIR community. We thank the organizers and participants of the BioHackathon for providing a productive and collaborative environment.