



Electronic Communications of the EASST
Volume 86 (1) Year 2026

deRSE26 - Selected Contributions of the 5th Conference for Research Software Engineering in Germany (Part 1)

Edited by Bernd Flemisch, René Caspart, Bernadette Fritzsch, Florian Goth, Jean-Noël Grad, Florian Thiery

Tools and services for research software quality

Elena Breitmoser, Shraddha Bajare, Faruk Diblen

DOI: 10.14279/eceasst.v86.1.2795

License: This article is licensed under a CC-BY 4.0 License.

Electronic Communications of the EASST (https://eceasst.org).
Published by **Berlin Universities Publishing**
(https://www.berlin-universities-publishing.de/)

Tools and services for research software quality

Elena Breitmoser [1], **Shraddha Bajare** [2], **Faruk Diblen** [3]

EPCC, University of Edinburgh, Edinburgh, UK[1], Square Kilometer Array Observatory, Cheshire, UK[2], Netherlands eScience Center: Amsterdam, NL[3]

Abstract: One of the objectives of the EU-funded EVERSE [eve26] project is to create a quality framework for research software and code excellence within the research communities.

Within this, our goal is to leverage existing resources and build services on top of them to assess the quality dimensions, such as FAIRness, Maintainability, Sustainability and to link these assessments to research community best practices that support the improvement of software quality.

To this end, we collected metrics commonly used by research communities and created the Research Software quality indicators catalog [ind26]. These are implemented through a list of tools, that assess different research software quality dimensions, by checking for indicators - like the software having a license, a unique persistent identifier, descriptive metadata, etc.

In this article, we present three components, the Technology Radar(TechRadar) [VDG+26], the Quality Pipeline (*resqui*) [GE26] and the EVERSE Dashboard (DashVERSE) [DE26], which together will take the Research Software Engineer (RSE) through the journey of assessing and improving the quality of their software. These tools complement the [Research Software Quality Toolkit RSQKit [rsq26a], a knowledge base developed by the EVERSE community, designed to inform researchers and RSEs about good practices in research software engineering.

1. The TechRadar is a curated, interactive catalogue of external tools and services designed to facilitate software improvement. Organized by quality dimensions (e.g., FAIRness), the TechRadar empowers users to discover and select the most appropriate solutions to address the specific issues identified in the assessment phase, thus closing the loop from analysis to action.

2. The Quality Pipeline *resqui* is a flexible to reuses existing tools that automatically assesses a software repository against a configurable set of quality indicators. Available as both a command-line interface for local analysis and a GitHub Action for seamless integration into CI/CD workflows, *resqui* generates a detailed, machine-readable report of the software's current quality status.

3. The DashVERSE is a visual analytics platform that ingests and displays the results from the *resqui*. This dashboard provides an intuitive interface to interpret assessment outcomes, identify specific quality deficits, and track the evolution of their software's quality over time.

Keywords: research software, research software quality, FAIR principles, research software engineering, RSE, EVERSE, RSQKit, automated assessment

1 Introduction

Research software is a cornerstone of modern science, yet ensuring its quality, sustainability, and adherence to community standards remains a significant challenge for developers and Research Software Engineers (RSEs) who look for guidance.

The **E**uropean **V**irtual **I**nstitute for **R**esearch **S**oftware **E**xcellence **(EVERSE)** is a three-year project (2024-2027), with 18 consortia across the European Union and the UK. Funded by the European Union's Horizon Europe Program, EVERSE is working to create a framework for research software and code excellence across Europe, with the purpose of setting the foundations of a future virtual institute for research software excellence, inspired by the UK Software Sustainability Institute SSI [ssi26].

EVERSE is informed by previous work on software quality assessment through a variety of past software maturity assessment approaches. The previous EOSC-Synergy project [EOS], [BOD+24] created a first quality software framework. The main difference between EOSC-Synergy and EVERSE is that the former is driven by an exclusively technical perspective, whereas EVERSE is more community-driven as is demonstrated by the support of the EOSC Clusters. The LFX Insights tool by the Linux Foundation [LFX] offers advanced dashboards for analytics. Their software quality metrics are focused on quality assurance with a grounding in security best practice. Also similar to the EVERSE dashboard is the UC Santa Cruz [UC-] repository browser. [CHA] provides various implementations of software to assess quality-related aspects of open source software. Projects like FAIR-IMPACT [fai25] extended the basis for the quality dimension of FAIRness for Research Software as well as the investigation and development of automated assessment tools. Overall, EVERSE is more focused on research software and takes a broader approach by providing an overall framework that integrates research software quality assessment via pipelines and dashboards as well as training and recognition.

In this article, we present work that focuses on the objective to leverage existing open-source tools and resources to support the, where possible automated, evaluation, verification, and improvement of research software code quality. This is done based on existing good practices and standards that can already be found in research communities.

For EVERSE these communities are represented by the five EOSC Science Clusters, covering Astronomy and Particle Physics, Environmental Sciences, Life Sciences, Photon and Neutron Science, and Social Sciences and Humanities.

In the following sections, we will provide more details about the EVERSE services and overall built-in quality framework.

2 Overview of the EVERSE quality framework

There are many stages in the development of research software, each with different challenges. To support this, EVERSE has identified the need for various tools and services as entry points into the process of assessing and improving research software.

Figure 1 summarizes the complete journey of EVERSE's research software quality framework development. The suite consists of five main components, the Research Software Quality Toolkit (RSQKit), the TechRadar, the *resqui* Quality Pipelines, the DashVERSE dashboard and

the underlying concept of software quality dimensions and indicators.

In principle, users can start this journey at any of the these five points and see what is most useful for them to support them in their goal to measure and improve the quality of their research software. The RSQKit can be viewed as the gateway with the most detailed introduction, background and discussion of best practices.

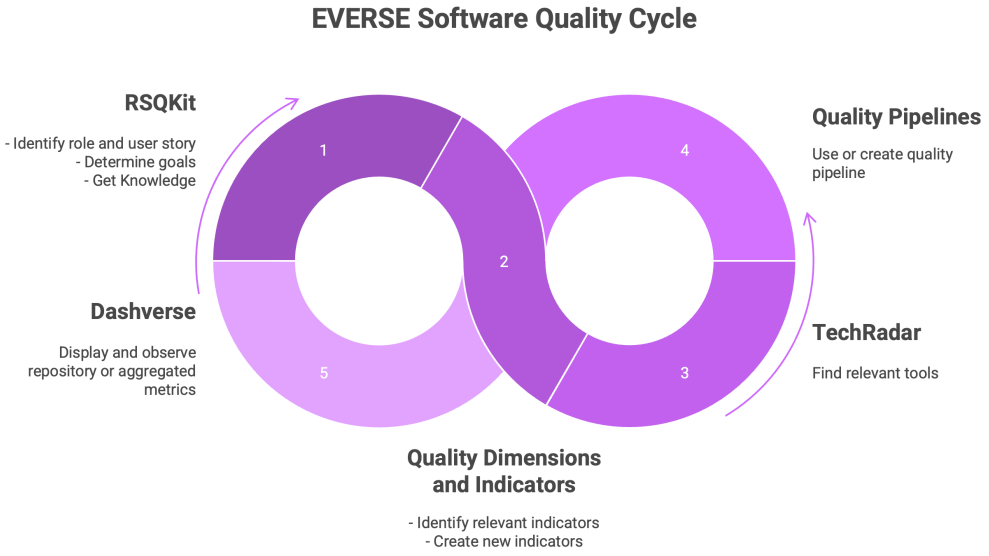


Figure 1: The EVERSE software quality tools and services cycle.

The purpose and workings of each of these five components are described in the following sections.

2.1 The concept of indicators and dimensions

The concept of research software quality indicators and dimensions in EVERSE - together with the actual tools as a third component - is crucial because it provides a standard with measurable criteria for assessing the quality of research software, and ensuring that it meets these standards for reliability, accessibility, and interoperability, which are so important for effective scientific research. The concept of indicators and dimensions is also used as a basis across the various EVERSE tools and helps to link them together.

- A **research software quality dimension** represents a collection of metrics relevant for assessing software quality. Currently, EVERSE uses twelve quality dimensions: Compatibility, FAIRness, Flexibility, Functional Suitability, Interaction Capability, Maintainability, Open Source Software, Performance Efficiency, Reliability, Safety, Security and Sustainability; most of which are based on the ISO/IEC 25010 standard [iso24], whereas e.g., FAIRness is used as defined according to [BNKa22]. The idea is to not re-invent the wheel but to use dimensions which are both already well known and have proven to be

useful across the research software communities and to extend the ISO standard-based, technical ones by a few others to ensure that the unique requirements of research software development are taken into account [CDGa26].

- A **research software quality indicator** represents a specific software aspect that can be measured [ind26], e.g., for the quality dimension of FAIRness some of the most intuitive and most easy to measure indicators are: does the software have a license, documentation, a persistent and unique identifier and similar measures.

EVERSE has compiled a complete list of quality dimensions and indicators [ind26], including a brief description for each item. A more detailed description and motivation can be found in EVERSE's Reference Framework document [CDGa26], a living document that gets updated as the project progresses.

2.2 The Research Software Quality Toolkit: RSQKit

The Research Software Quality Toolkit (RSQKit) [rsq26a] provides a comprehensive knowledge base (see Figure **??**) designed to gather and curate expertise, to enhance software and code quality across various fields.

By integrating the RSQKit and its services into the EOSC Science Clusters, EVERSE showcases improvements in research software quality to maximize its reuse. This will lead to standardized development practices and sustainable research software. The RSQKit brings all ideas and concepts of EVERSE together: Best practices, which are sourced both internally and externally and developed under the guidance of EVERSE; dimensions and indicators; tools from the TechRadar as RSQKit shares sub-set of tools and resources for research software quality from TechRadar service [rsq26b]; collected training resources that are available through the EVERSE Training Portal and information about Research Software Clusters and Infrastructures and related Research Software Stories. The five roles of policy maker, principal investigator, RSE, researcher who codes and trainer[rsq26c] are taken up again in the DashVERSE. And finally, the *Tasks*[rsq26d] provide an alphabetically ordered, detailed list of concepts related to research software development, deployment, sharing and maintenance if a deeper understanding of each of the topics is needed by the user.

2.3 The EVERSE Technology Radar: TechRadar

The EVERSE Technology Radar (TechRadar) is a curated, interactive platform developed within the EVERSE project to support the assessment and improvement of research software quality. It serves as both a catalogue of tools and services and a visual dashboard, enabling users to explore relevant solutions in an accessible and structured way. Figure 2 shows the TechRadar service website.

At its core, TechRadar[VDG+26] organises tools according to a set of software quality dimensions, providing a systematic overview of how different tools contribute to improving research software. These dimensions include areas such as FAIRness and many others as described in section 2.1. The platform does not attempt to list every available tool; instead, it applies selection criteria to highlight those that are most relevant and useful for the research community, particularly

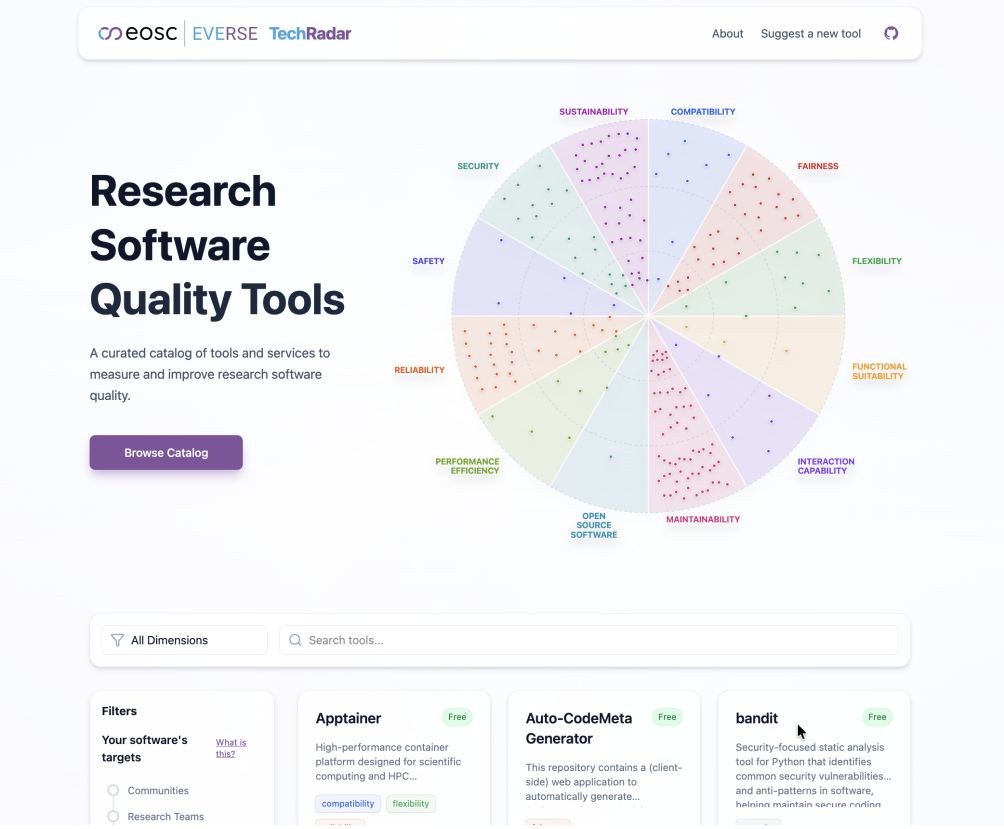


Figure 2: The EVERSE Technology Radar.

within the EOSC science clusters. This curated approach helps researchers and research software engineers quickly identify appropriate tools to address specific quality challenges. Overall, EVERSE TechRadar acts as a bridge between best practices and practical implementation by helping users navigate the landscape of research software quality tools and make informed decisions about improving their software. Users can explore the platform either visually through the radar interface or via the searchable catalogue of tools and services available on the website.

The platform provides several filtering and search mechanisms that allow users to identify tools relevant to a specific software quality problem. Tools can be filtered according to software quality dimensions such as FAIRness, Maintainability, Security, Performance Efficiency etc. In addition, users can search by keywords, tool names, categories, programming languages, or EVERSE quality indicators. Each tool entry contains a short description, links to external resources, and metadata describing the purpose and maturity of the tool, helping users quickly evaluate whether it is suitable for their use case.

This filtering approach enables researchers and RSEs to efficiently discover tools that address particular quality deficits identified during software assessment and supports the practical implementation of research software best practices.

2.3.1 TechRadar: Tools Selection and Catalogue Creation

The TechRadar adopts a structured contribution and review process to ensure that all catalogue entries are relevant, accurate, and consistent. New tools and services are submitted using a standardised metadata schema [DGV+25] that captures essential information, including the tool's purpose, official URL, licence, supported quality dimensions and indicators, usage modes, and the research software tiers for which it is applicable. Every tools submission must satisfy a set of predefined inclusion criteria. Specifically, tools and services must be designed to measure and/or improve software quality, be actively maintained and used within the research software community, support recognised research software engineering standards and best practices, and provide capabilities that improve software quality across one or more stages of the research software lifecycle, from development to long-term sustainability. The metadata schema promotes consistency across the catalogue by requiring contributors to map tools to the EVERSE Quality Dimensions and Quality Indicators. This enables users to discover and compare tools based on the specific software quality aspects they measure or improve. Submitted entries are reviewed for completeness, metadata quality, and compliance with the inclusion criteria before being incorporated into the TechRadar, ensuring a reliable and well curated catalogue for the research software community.

2.4 The EVERSE quality pipeline: *resqui*

The EVERSE quality pipeline for REsearch Software QUality Indicators (*resqui*) [GE26] offers a suite of tools for the automated measurement of research software quality. The user has the option to either use the suite locally from the command-line interface or via GitHub Actions. The command-line tool requires Python v3.9 or above as well as Docker. Currently, the *resqui* can only assess GitHub repositories, but the aim is to support GitLab and potentially other forges in the future if time permits. The users has the flexibility to configure which quality indicators to measure and with which tools, e.g. if a certain quality indicator can be assessed by multiple tools. It is the user's responsibility to decide which assessment tools to use, to understand what each tool does and does not assess as well as how it does the assessment, e.g., to understand where a tool looks for license files as well as for which formats of license files it looks. We want to stress here that EVERSE provides a service built on top of already existing tool and hence does not develop any individual software quality assessment tools. For a better understanding, the user can consult the Task rubric of RSQKit and then move on to the TechRadar before applying a specific tool in the quality pipeline assessment. In addition, the RSEs can develop and add their own plugins to measure any other indicators they need to have assessed. Detailed documentation [res26b] for every-day usage as well as instructions of how to build your own plug-in are provided.

The v0.2.0 version of *resqui* measures twenty-six indicators, such as license, citation, etc. These python scripts are publicly available on github [res26a] and can be used - together with the detailed, general documentation - as examples for developers. Any plugins developed by the external contibutors can be added to the pipeline via a GitHub Pull Request that will be reviewed by the EVERSE team before becoming publicly available to the community.

The *resqui* checks indicators for a user's software repository as shown in Figure 3 and gener-

ates an assessment report. The generated report is ingested to the DashVERSE via REST API as the next step. Figure 3 gives an example in which the GitHub Action was checking for the quality indicators 'has_license', 'has_citation', 'has_ci_tests', etc. with the automated assessment tools 'HowFairIs' ([Dib26b]), 'CFFConvert' [cff] and 'OpenSSFScorecard' [ope], and that all of these three tests failed for the specific repository investigated, whereas the test for the indicator 'has_no_security_leak' passed using the 'Gitleaks' tool [git].

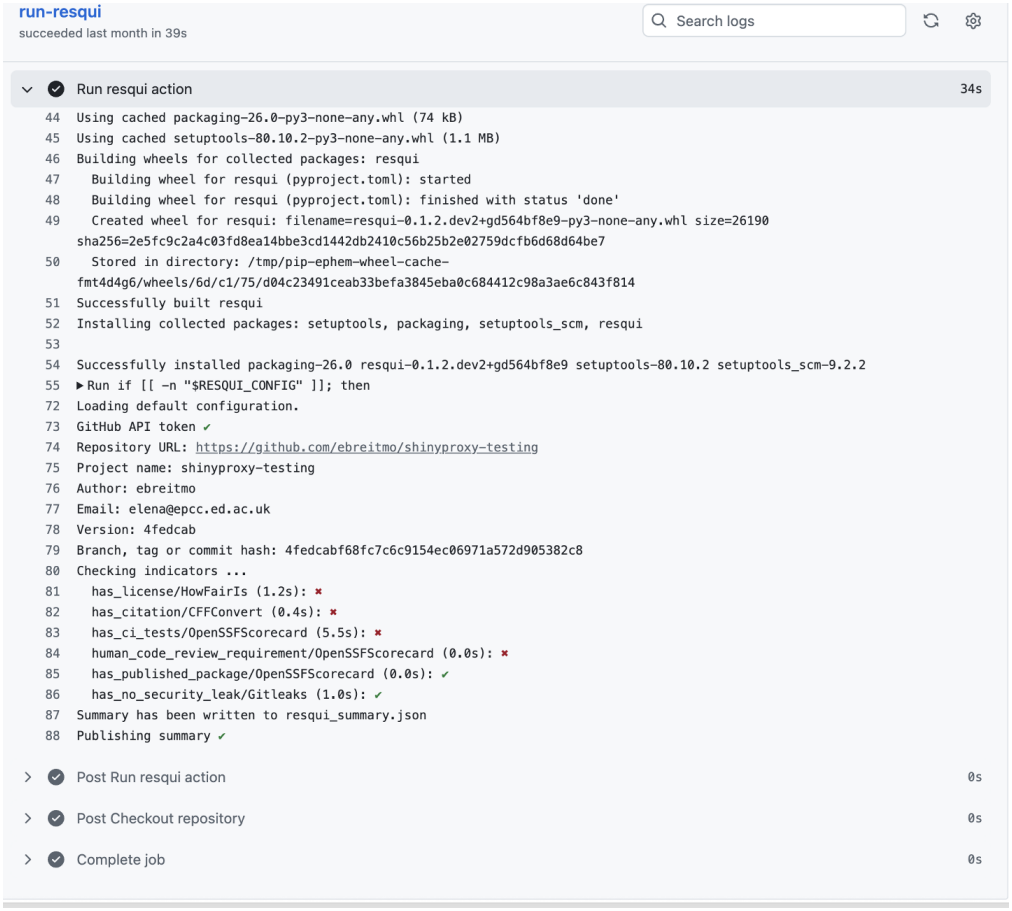


Figure 3: The *resqui* pipeline output when run on an example GitHub repository via GitHub Actions.

2.5 The EVERSE dashboard: DashVERSE

The DashVERSE [DE26] is the central visual analytics and observation platform of the EVERSE project. It serves as a unified hub designed to simplify the interpretation of research software quality assessments by translating complex technical checks of indicators into actionable insights for a diverse range of stakeholders. The dashboard acts as the reporting layer of the EVERSE quality framework, bridging the gap between automated technical assessments and

human-centered decision-making.

A public-facing Demonstration Portal [Dib26a] has been established to facilitate immediate engagement and validation by the Science Clusters and the broader community. Figure 4 shows a snapshot of the application.

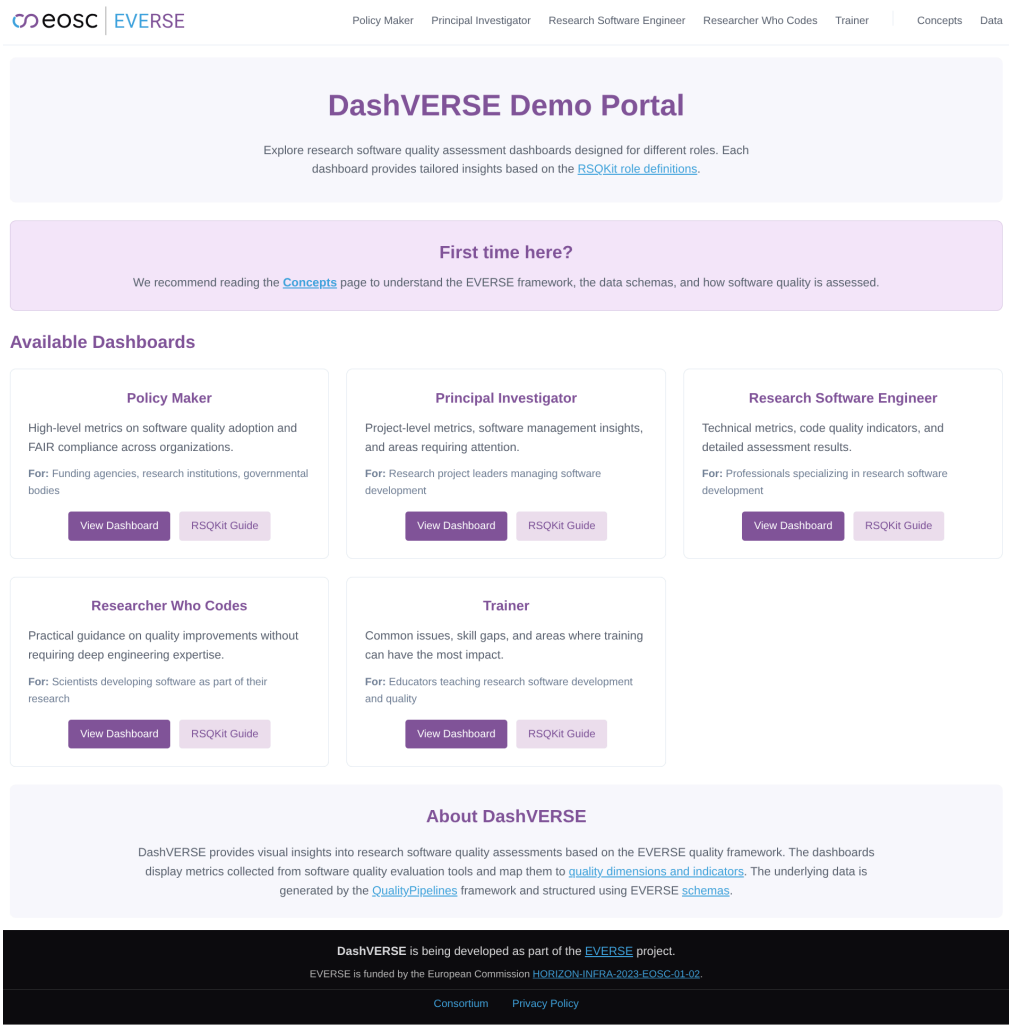


Figure 4: The DashVERSE demonstration portal version 0.0.2.

2.5.1 Role-Based Visual Analytics

A core feature of DashVERSE is its alignment with the five key personas identified in the RSQKit: Policy Makers, Principal Investigators (PIs), Research Software Engineers (RSEs), Researchers, and Trainers. This role-based approach ensures that data is presented in a contextually relevant manner:

- **Policy Makers** receive high-level overviews of software quality adoption and compliance across entire portfolios or organizations.
- **Principal Investigators** can monitor the sustainability and long-term health of their research software projects.
- **Research Software Engineers** have access to detailed technical indicator breakdowns, facilitating debugging and targeted code improvements.
- **Researchers** gain practical insights into the reliability and reproducibility of their scientific outputs.
- **Trainers** can identify common pitfalls and quality gaps within specific scientific communities to develop targeted educational resources.

These personas were actively used during the development of DashVERSE.

The platform currently provides distinct charts and detailed data views, including quality heatmaps, dimension coverage radars, and assessment trends over time. Figure 5 illustrates an example of these specialized, role-based dashboard prototype for RSEs.

2.5.2 Integration with the EVERSE Ecosystem

DashVERSE is seamlessly integrated with the broader EVERSE ecosystem. It natively supports the EVERSE Research Software Quality Indicator vocabulary and dynamically synchronizes with the central indicator repository. The dashboard provides a secure ingestion endpoint for JSON-LD assessment reports generated by the *resqui* quality pipelines. This integration completes the EVERSE "Software Quality Cycle," where results from CI/CD-based testing are automatically surfaced in the user-facing dashboard, providing near real-time visibility into the quality status of research software.

DashVERSE is engineered for versatility and local adoption. Individual research organizations or institutions are encouraged to deploy their own local instances to manage quality assessment data internally. The choice of Apache Superset as the presentation layer allows these organizations to highly customize the platform, tailoring existing dashboards or creating entirely new visualizations to meet their specific local requirements and scientific workflows.

The public prototype and demonstration portal are accessible at [Dib26a], providing a low-barrier entry point for the community to explore the capabilities of the EVERSE quality framework.

2.6 Example Use Cases

The tools in EVERSE have been designed so that a user can start their journey at any of the five entrypoints in the software cycle (See Figure 1).

EVERSE is currently developing a demonstrator [GDMG] (that we expect to be available on Zenodo in due course). Its aim is to show how the EVERSE framework has been adopted by existing services and by different communities. The EVERSE framework comprises the research

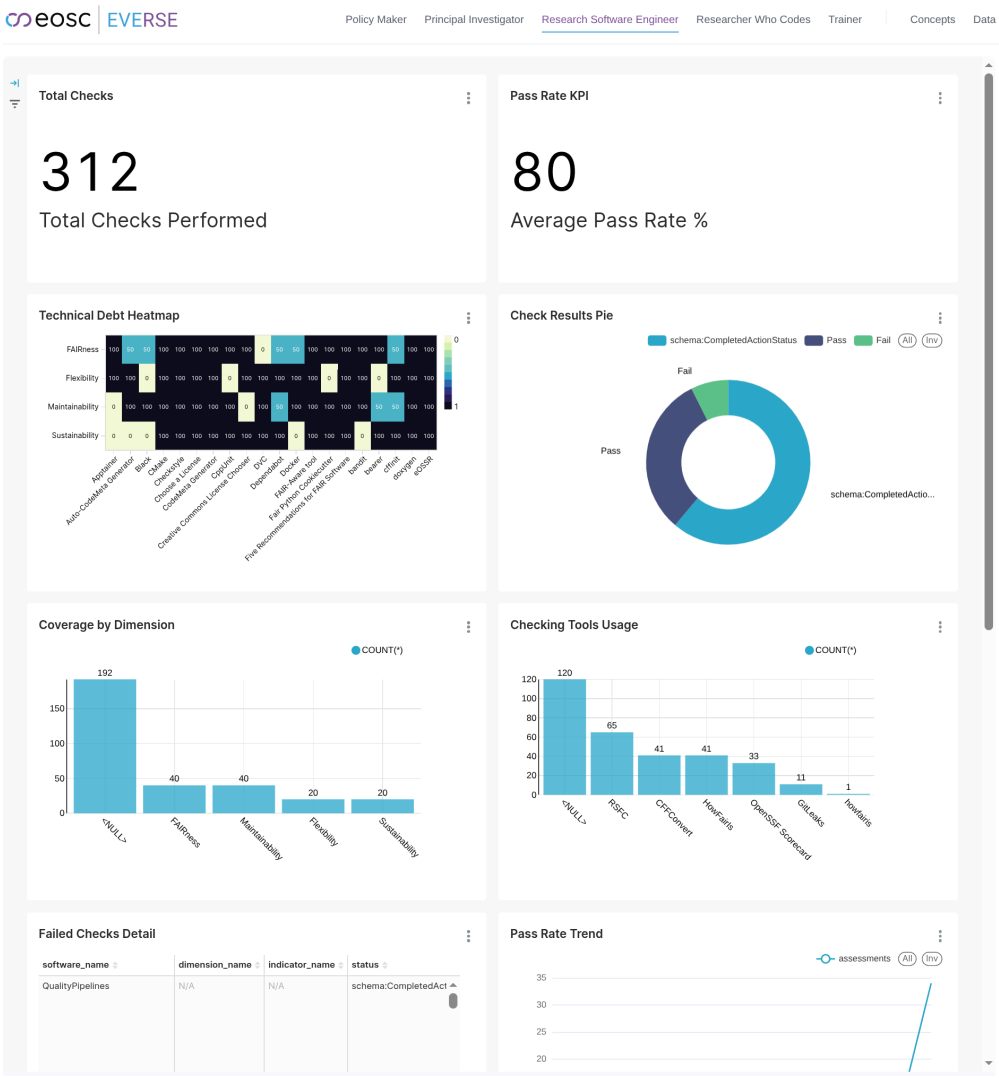


Figure 5: The Research Software Engineer Dashboard (DashVERSE version 0.0.2).

software quality indicators and dimensions and the EVERSE Ecosystem of tools and services (RSQKit, *resqui*, DashVerse and TechRadar).

This study involved several communities across the European Science Clusters: ELIXIR [eli26] (Life Sciences), ENVRI [env26] (Environmental Sciences), Square Kilometre Array Observatory [ska26] and the Open-source Scientific Software Repository (OSSR) [oss26] (Astronomy and Astrophysics) and the High Energy Physics (HEP) [hep26] and three different levels of adoption of the EVERSE environment:

- Level1: Best practices - research quality indicators and dimensions are adopted and aligned with the best practices guidelines proposed by a scientific community or organization. For example, the ELIXIR community provides an analysis of software quality indicators and

dimensions, which are aligned with the ones proposed by EVERSE.

- Level2: Assessment Pipelines - the target community or organization has contributed to the development of plugins to assess their selected indicators, or used the EVERSE *resqui* pipelines to assess the software of their own institutions.
- Level3: Monitoring - a demonstration service exists to monitor the quality indicators required by the target community or organization. Alternatively, the target community or organization has adopted and deployed DASHVerse to monitor their indicators.

Collaborators from the Universidad Politécnica de Madrid (UPM) started integrating EVERSE functionalities to the software catalog of their organization [upm26] based on the indicators and FAIR4RS [BNKa22]. This illustrates how any community will be able to install and run their own instance of e.g. the dashboard independently. The UPM adoption provides information on software quality and project health to the user. This is work in progress and other *resqui* tools will be added.

3 Limitations

There are limitations to both the EVERSE framework and the automated assessment of research software in general, of which any responsible user should be aware of.

The assessment of research software quality is rather complex. The automation of assessment tools might make the assessment less subjective, but as already touched upon in section 2.4, the user has to be aware what exactly each tool is measuring and where it is looking for the evidence, relying on transparency provided by the tool developers. While some criteria are rather straightforward to measure and check for, others are more complex and harder to automate. Basing the software quality dimensions on ISO standards guarantee their consistency. Moreover, the automated pipelines rely on third party tools. This may affect the reliability of the plugins in case the the third party tool has some development or sustainability issues. Another challenge is to ensure the continuing development and sustainability of the tools to reflect their relevance and usability for the assessment of research software. Another consideration, is the setting of thresholds, which define if either an overall or individual tests passes or fails.

4 Sustainability and Future Outlook

Sustainability of research software projects is often challenging and the EVERSE teamEVERSE team is still working on their sustainability plan. To ensure the sustainability of the project's activities beyond its current funding cycle, the consortium is investigating various options, which include establishing a fully legal entity (the Virtual Institute), signing a formal Memorandum of Understanding that defines the EVERSE consortium's ongoing cooperation, as well as positioning within the overall EOSC Ecosystem, including potential targeted funded options [Con26].

The uptake of the EOSC Science Clusters of the EVERSE tools and their integration into their day-to-day workflows will be crucial to ensure the future of the idea underlying EVERSE, as well as ensuring that the tools and services will continue to be a living toolkit that remains relevant for

the communities. It is one of the project's main goals to work with the EOSC Science Clusters and collaborate on the integration process during its last project year.

Another pillar to guarantee the future sustainability of many tools and services is their potential automation via AI agents. E.g., for the TechRadar most processes can be automated via AI agents that can perform a first check on newly suggested tools by external contributors from the community and assess the consistency with the tools' websites thus reducing the review process for human reviewers. For already ingested tools, regular checks if all metadata are still up-to-date can be automated in the same way.

5 Conclusion

The EVERSE tools and services are being developed in close collaboration with research communities and five EOSC Science Clusters to ensure that they address real-world requirements and workflows to assess and improve the quality of research software. By building on existing open-source tools, integrating with established community initiatives, and fostering broad participation across scientific domains, EVERSE aims to establish a sustainable foundation for future research software quality services in Europe based on universally established and accepted standards.

A key challenge beyond the project lifetime will be the long-term sustainability and maintenance of the EVERSE ecosystem. Ensuring that the tools, indicators, and guidance materials remain up to date with evolving community standards, software practices, and technological developments will require continued engagement from both infrastructure providers and research communities.

The long-term success of EVERSE will ultimately depend on community adoption and continued collaboration. If the framework, tools, and practices become embedded in everyday research software development workflows and are supported by active user and contributor communities, EVERSE has the potential to continue evolving beyond the funded project period and contribute to a lasting culture of research software excellence.

RSE developers, researchers, community leaders, and RSE managers are encouraged to provide feedback throughout the project's lifetime via EVERSE's issue templates [fee26] in order to help refine the services and maximize their usefulness for the wider community including their own.

Acknowledgements: We acknowledge the hospitality of the university of Stuttgart for hosting the deRSE26 workshop (Stuttgart, Germany; 3rd-5th March 2026) and the organizers of deRSE26 for letting us present our work, and the participants of the workshop for their enthusiastic and thoughtful engagement with our talk.

EVERSE has received funding from the European Union's Horizon Europe Programme under GA 101129744 — EVERSE — HORIZON-INFRA-2023-EOSC-01-02. Support for E. Breitmoser's work has been provided by UKRI Innovate UK to the University of Edinburgh through its Horizon Europe Guarantee Extension scheme (Application Number: 10121158). This publication would not have been possible without the other team members of the EVERSE project, and in particular the Work Package 3 team for the technical development of the tools and services as

well as the team developing the RSQKit and the contributors to the EVERSE Reference Framework defining the research software quality dimensions. Particular thanks to Kirsty Pringle for proof-reading and discussions, Daniel Garijo-Verdejo , Fotis Psomopulos and Neil Chue-Hong for discussing the reviewers' feedback with the authors, as well as Daniel Garijo-Verdejo as the lead for the quality indicators and dimensions, Tamás Gál as the technical lead and developer of the *resqui*-pipeline and Thomas Vuillaume as the lead for the TechRadar.

We are grateful to the reviewers of this manuscript for their helpful comments, time and care in reviewing this article.

Declaration on the use of AI: No AI has been used for writing, language, style editing purposes nor for the creation of tables, figures or images or the creation of the bibliography of his report. The overall design of the TechRadar was developed and determined by the TechRadar curator team. AI tools were used to assist with the implementation of advanced user interface features. The TechRadar curators retained full responsibility for all design decisions, implementation, and the final content of the service.

The core functionality of the DashVERSE project was designed and implemented by the development team without the use of AI code generation tools. In the later stages of the project, AI-assisted development tools were selectively used as a helper for specific tasks such as documentation drafting, boilerplate generation, and code formatting.

Data availability: The data needed to create Figure 2 are all based on publicly available weblinks to the various software quality assessment tools. All data used for the creation of the dashboard in Figure 5 are based on open github repositories as example input.

Bibliography

[BNKa22] M. Barker, N.P.Chue-Hong, D. Katz, et al. Introducing the FAIR Principles for research software. Scii Data Journal 9(622), Oct. 2022.
doi:https://doi.org/10.1038/s41597-022-01710-x
https://www.nature.com/articles/s41597-022-01710-x

[BOD+24] S. Bernardo, P. Orviz, M. David, J. Gomes, D. Arce, D. Naranjo, I. Blanquer, I. Campos, G. Moltó, J. Pina. Software Quality Assurance as a Service: Encompassing the quality assessment of software and services. Future Generation Computer Systems 156:254–268, 2024.
doi:https://doi.org/10.1016/j.future.2024.03.024
https://www.sciencedirect.com/science/article/pii/S0167739X24000955

[CDGa26] N. Chue-Hong, F. Diblen, D. Garijo, et al. EVERSE Reference Framework. 2026. EVERSE Reference Framework v3.0.
https://zenodo.org/records/18861872

[cff] Generate your citation metadata files with ease.
https://citation-file-format.github.io/cff-initializer-javascript/#/

[CHA] CHAOSS Software.
https://www.chaoss.community/software/

[Con26] E. Consortium. EOSC EVERSE 2nd General Assembly Meeting Report. 2026.
https://doi.org/10.5281/zenodo.20406690

[DE26] F. Diblen, European Virtual Institute for Research Software Excellence (EVERSE). DashVERSE v0.2.0. Apr. 2026.
https://github.com/EVERSE-ResearchSoftware/DashVERSE

[DGV+25] F. Diblen, D. Garijo, T. Vuillaume, A. Montero, A. Nenadic. EVERSE-ResearchSoftware/schemas: 0.0.2. Nov. 2025.
doi:10.5281/zenodo.17736584
https://doi.org/10.5281/zenodo.17736584

[Dib26a] F. Diblen. DashVERSE. 2026.
https://www.dashverse.cloud/

[Dib26b] F. Diblen. Howfairis: Fully automated FAIR software assessment. 2026.
https://www.howfairis.com/

[eli26] Europe's distributed research infrastructure for life science data. 2026.
https://elixir-europe.org/

[env26] The ENVRI-FAIR project. 2026.
https://envri.eu/the-envri-fair-project/

[EOS] EOSC Synergy.
https://www.eosc-synergy.eu/

[eve26] European Virtual Institute for Research Software Excellence. 2026. EVERSE main web page.
https://everse.software/

[fai25] FAIR-IMPACT. 2025.
https://fair-impact.eu/

[fee26] Feedback EVERSE Services. 2026.
https://github.com/EVERSE-ResearchSoftware/tools-feedback/

[GDMG] D. Garijo-Verdejo, F. Diblen, J. Maason, K. Graf. Milestone 11: Demonstration of the outcome of the common framework to the research community, by showing software quality indicators in existing services (such as OpenEBench, WorkflowHub, Research Software Directory,ENVRIHUB, etc) based on the dashboard prototype.

[GE26] T. Gal, EVERSE. resqui. Apr. 2026.
https://github.com/EVERSE-ResearchSoftware/QualityPipelines

[git] Gitleaks.
https://github.com/gitleaks/gitleaks

[hep26] HSF - HEP Software Foundation. 2026.
https://hepsoftwarefoundation.org/

[ind26] EVERSE-research software quality indicators. 2026. EVERSE github repository, indicators and dimensions.
https://github.com/EVERSE-ResearchSoftware/indicators

[iso24] ISO/IEC 25010. 2024.
https://iso25000.com/index.php/en/iso-25000-standards/iso-25010

[LFX] LFX: Discover the world's most critical open source projects.
https://insights.linuxfoundation.org/

[ope] OpenSSF Scorecard.
https://github.com/ossf/scorecard

[oss26] Open-source Scientific Software Repository. 2026.
https://escape-ossr.gitlab.io/ossr-pages/

[res26a] EVERSE-ReserachSoftware Quality Pipelines. 2026.
https://github.com/EVERSE-ResearchSoftware/QualityPipelines/tree/main/src/resqui/plugins

[res26b] resqui. 2026.
https://everse.software/QualityPipelines/

[rsq26a] Research Software Quality Kit (RSQKit) (v1.4.0). Zenodo, 2026.
https://doi.org/10.5281/zenodo.18786768

[rsq26b] RSQKit: All tools and resources. 2026.
https://everse.software/RSQKit/all_tools_and_resources

[rsq26c] RSQKit: Roles. 2026.
https://everse.software/RSQKit/roles

[rsq26d] RSQKit: Tasks. 2026.
https://everse.software/RSQKit/tasks

[ska26] Standards applicable to derived software. 2026.
https://developer.skao.int/en/latest/policies/fundamental-sw-requirements.html#standards-applicable-to-derived-software

[ssi26] Software Sustainability Institute (SSI). 2026.
https://www.software.ac.uk/

[UC-] UC OSPO Network: Hidden in Plain Sight: Discovering the University of California Open Source Landscape.
https://ucospo.net/events/uc-open-2026/talks/hidden-in-plain-sight-uc-open-source-landscape/

[upm26] UPM - Software Catalogue. 2026.
https://software.oeg.fi.upm.es/portals/oeg-upm/

[VDG+26] T. Vuillaume, F. Diblen, A. Gamgami, S. Bajare, E. Breitmoser, N. Pechlivanis, K. Pringle, S. Ghosh, D. Garijo. EVERSE TechRadar (v2.0). Zenodo, 2026.
https://doi.org/10.5281/zenodo.19389279