

EXECUTIVE SUMMARY

Keep the software alive. Let the science grow.

A foundation for the long-term care of scientific software

Scientific software carries methods, assumptions and practical knowledge from one research team to the next. Yet its useful life can extend far beyond the grant, contract or career stage that created it. Open Science Continuity proposes a nonprofit institution to take responsibility for that longer life.



Illustrative continuity metaphor, adapted from the accompanying pitch deck. The foundation would bridge project funding and continued research use.

The proposal

Combine supervised AI agents with research software engineers, scientific reviewers and institutional agreements. Agents would inspect code, prepare repairs and assemble test evidence. Humans would judge engineering quality and scientific consequences; maintainers would retain release authority. The service would continue monitoring supported tools and plan handover or retirement when appropriate.

Why now, why here

The Netherlands is already investing in research-software sustainability, while European policy promotes verifiable, reusable research outputs. These developments create a practical setting for an operating foundation that turns support into continuing maintenance. The ambition is a Dutch base, European cooperation and global access, with research communities retaining ownership and scientific control. [7-11]

Care for one tool. Strengthen many discoveries.

01 / THE SCIENTIFIC CASE

Shared tools. Shared scientific exposure.

Published code needs a continuing path back to research use.

A numerical library, imaging package or analysis workflow can support work across institutions and disciplines. The SciPy paper reported more than 110,000 dependent GitHub repositories and 6,500 dependent packages in its February 2019 snapshot. This historical example illustrates shared infrastructure; it is not a current adoption count or a measure of abandonment. [5]

Evidence from published research code

Study and population	Measured finding	Interpretation
Trisovic et al., 2022 [3] 2,109 Harvard Dataverse replication packages; 9,078 R files, deposited 2010-July 2020.	74% of R files initially failed to complete without error; 56% failed after automated code cleaning.	Automation repaired some practical barriers. Re-execution remains distinct from scientific validity. This was not an LLM experiment.
Samuel & Mietchen, 2024 [4] Jupyter notebooks linked to biomedical publications.	Of 15,817 Python notebooks with declared dependencies attempted, 10,388 had dependencies installed; 1,203 ran without errors and 879 reproduced identical recorded results.	These are notebook-level outcomes in a selected corpus under an automated protocol. They cannot be extrapolated to all biomedical research.

From breakdown to a service obligation

These studies measure execution barriers, not a global abandonment rate. Inactivity can describe stable software, while an active repository can still fail users. The practical targets are environment setup, dependencies, paths, missing inputs and execution errors. The foundation would assess these barriers and document what remains uncertain.

For each supported repository, continuity should mean a known maintainer, a supported environment, meaningful tests, accessible documentation and a recorded route for reviewing changes. Availability alone is insufficient: an archived repository can preserve code while leaving a new user unable to execute it.

The outcome is continued research use, not simply more code online.

02 / THE INVESTMENT CASE

Protect the value already created.

Maintenance is part of the cost of usable research infrastructure.

The research enterprise is large. CBS reports EUR 26.438 billion in Dutch in-house R&D expenditure for 2024, including EUR 6.429 billion in higher education; those figures are revised provisional. Eurostat reports EUR 408 billion in EU R&D expenditure for 2024 in its September 2026 release. These totals include sectors and activities beyond academic research and software. [1,2]

Investment signal	Amount and period	What it supports
Netherlands: research-software sustainability [7]	EUR 5.4 million; announced January 2026.	A programme spanning technical maintenance, funding models, governance and recognition.
Netherlands: OSS 2026 call [8]	Eight planned projects, up to EUR 250,000 each over up to two years.	Primarily in-kind RSE time: EUR 225,000, about 2,000 hours per project. Final decisions were scheduled for December 2026.
Europe: EVERSE [9]	EUR 6.789 million EU contribution; March 2024-February 2027.	Research-software quality, sustainability, recognition and a European network; not solely maintenance delivery.
International: Open Source for Science Fund [15]	2026 inaugural call ceilings: USD 250,000 for domain tools; USD 1 million for foundational efforts.	Grant ceilings, not total spending or awarded funding. A further signal of demand for scientific open-source support.

The gap is continuity, visibility and accountability

Dedicated support exists, and software work also happens inside ordinary grants, institutional salaries and infrastructure budgets. The sources reviewed here do not establish what fraction of all research spending pays for software creation or long-term care. The amounts above have different scopes and periods and must not be divided to invent a maintenance-spending percentage.

An international funder survey describes the tension between short-term, novelty-oriented funding and software needs such as maintenance, community building and stable engineering careers. It also records emerging dedicated and embedded support. The foundation would offer a continuing delivery mechanism alongside these reforms. [6]

03 / THE GEOGRAPHIC AMBITION

Dutch base. European reach. Global access.

Build locally, coordinate across borders, share the benefit.



Illustrative ambition from the pitch deck. Institutions depicted are symbolic; no partnership or endorsement is implied.

Netherlands: a practical starting point

Start with a coalition of research institutions, software communities and funders that can nominate tools and share engineering responsibility. The Dutch sustainability programme provides a relevant policy and delivery setting. A foundation could add reusable agent workflows, continuing service agreements and transparent evidence of what remains supported, subject to collaboration being agreed. [7,8]

Europe: shared capability, distributed expertise

European research crosses borders; its software support can do the same. The proposed foundation would work toward a federated network in which domain experts and RSE teams remain distributed while sharing maintenance procedures, evaluation tasks and continuity records. This aligns with European open-science objectives and could draw on EVERSE quality practices rather than inventing competing standards. [9-11]

World: access with local authority

Global access should be part of the operating model from the start. Partners with fewer resources could receive supported maintenance, practical documentation and local maintainer development. Local teams would choose priorities and retain ownership. Adaptation for modest hardware or limited connectivity would be pursued where technically feasible. This gives practical form to UNESCO principles of equitable access and collective benefit. [20]

Expansion should follow demonstrated capacity and partner demand. A European network and global programme are ambitions to build, not claims about an existing federation.

04 / THE ECOSYSTEM

Connect the strengths already in place.

A continuing operator within a growing research-software community

The foundation enters an active ecosystem. Preservation, discovery, quality guidance, funding and engineering support all matter. Its proposed contribution is to combine these resources into an accountable, continuing service for selected software portfolios.

Existing contribution	Examples	Proposed interface
Engineering and national sustainability	Netherlands eScience Center; Open Science NL [7,8]	Work with institutional expertise; share maintenance evidence and extend continuity beyond individual interventions.
Quality practices and community	EVERSE / RSQKit [9,10]	Use and contribute to shared quality guidance; adapt it to monitored maintenance workflows.
Discovery and research context	Research Software Directory [12]	Connect supported repositories to existing discovery, impact and reproducibility information.
Source preservation	Software Heritage [13]	Preserve code and provenance while maintaining supported executable environments.
Funding scientific open source	Open Source for Science Fund [14,15]	Provide a potential delivery capability for funded portfolios, without implying eligibility or a funding relationship.

What makes the proposal distinctive

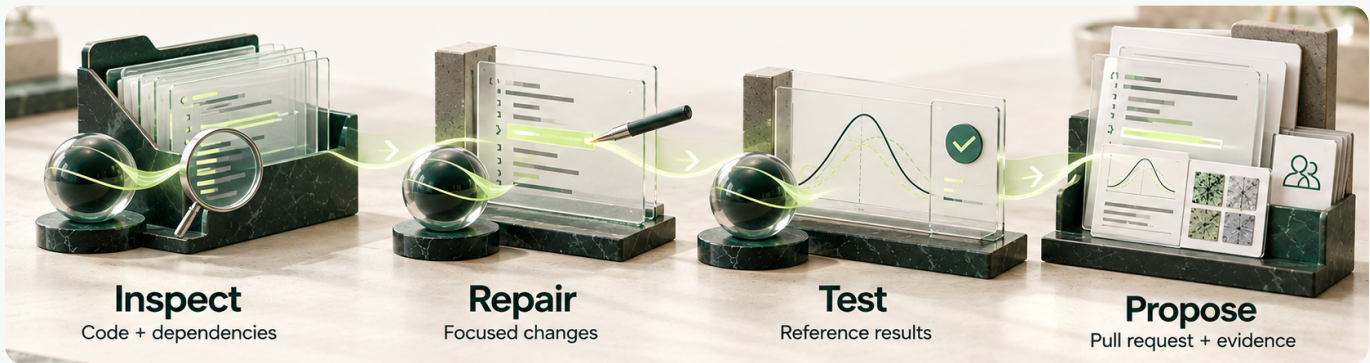
A generic coding agent can propose a patch. The proposed foundation would accept an agreed service responsibility: understand the scientific purpose, establish an acceptance baseline, organise review, monitor supported environments and maintain a continuity plan. The institution, its evidence and its relationships would persist when a model, employee or grant changes.

This differentiation must be demonstrated in practice. The Open Source for Science Fund already explicitly addresses automation and autonomous workflows, so using AI alone does not establish novelty. The case rests on accountable execution and scientifically reviewed continuity. [14]

05 / THE OPERATING MODEL

From repository to a tested repair.

Agents prepare. Experts decide.



Proposed workflow. Test results support review; scientific validity requires expert judgement.

An agent combines a language model with tools such as repository search, a shell, package managers and test runners. It can inspect a failure, act through several steps and revise a proposed fix. The foundation would define its scope and permissions and run the work in controlled environments.

Stage	Work and evidence	Accountability
Intake and baseline	Record ownership, licence, supported environments, priority uses and reference outputs.	Maintainer and scientific lead define acceptance criteria.
Inspect and repair	Reproduce issues, prepare focused changes and document the reasoning and changed files.	Agents work within an approved task; RSEs review engineering quality.
Test and propose	Run regression and reference analyses; compare outputs to agreed tolerances. Submit a pull request with evidence.	Experts review scientific effects. Maintainers decide release.
Monitor and hand over	Schedule checks, triage issues and record changes, escalation and handover or retirement.	The foundation delivers the service agreed with each partner.

Baseline failures should be reported openly; tests must not be weakened to accept a patch. Where essential data cannot be shared, partners could run reference checks locally and return agreed evidence. Tests of scientific outputs remain separate from checks that code executes.

Better models. A stronger continuity service.

Model progress can expand capability; the institution makes it usable.

The opportunity is broader than faster code generation. Better reasoning, tool use and repository understanding could help agents handle more complex migrations, reproduce difficult failures and prepare clearer review evidence. If those improvements reduce total maintenance effort, the foundation could support more tools or offer deeper service with the same resources. This is a proposed benefit to measure, not a guaranteed forecast.

The evidence supports ambition and disciplined evaluation

SWE-bench established a reproducible benchmark for resolving real repository issues. METR's January 2026 evaluation update reports continuing increases in the human-equivalent length of software tasks models can complete, while also noting wide uncertainty and sensitivity to task composition. Neither establishes that a scientific algorithm remains valid after a repair. [16,17]

Real-world productivity can differ from benchmark performance. METR's early-2025 randomised study involved 16 experienced developers and 246 tasks and found that AI access increased completion time by 19% in that setting. Its February 2026 follow-up described likely improvement but substantial selection effects that made the size of any gain uncertain. These results support measuring reviewed maintenance outcomes and costs rather than promising an automatic speed multiplier. [18,19]

How the foundation would capture model progress

- 1. Keep models replaceable.** Use interchangeable model providers and, where suitable, open models. Preserve task records, environments, reference results and decision rules independently of any single model. Record model and tool versions so a completed run remains auditable.
- 2. Evaluate upgrades on real maintenance work.** Compare a candidate model with the current system on held-out repository tasks and independent reference checks. Require evidence of acceptable scientific outcomes, review effort and cost before adopting it more broadly.
- 3. Compound reusable assets.** Every supported tool can add verified setup instructions, tests, reference data, migration examples and maintainer knowledge. Better models can use this structured context; better context can also make current models more effective. This is an operational feedback loop to test.
- 4. Reinvest demonstrated gains.** When total cost falls or useful throughput rises, reinvest in neglected tools, stronger validation and global-access support. Broader capability may increase rather than reduce expert-review demand, so staffing must follow measured workloads.

07 / WHY A FOUNDATION

Shared infrastructure. Shared responsibility.

Finance continuing care and preserve scientific authority.

The proposed foundation would be an operating organisation with a public-benefit mandate. Its durable assets would include supported software portfolios, scientific reference tests, transparent maintenance records, reusable agent workflows and a community of maintainers and reviewers. This mandate can survive changes in vendors and technology.

A mixed funding model

Institutional maintenance agreements would provide recurring income for defined portfolios and service levels. Research and infrastructure grants could fund onboarding, evaluation, scientific validation and shared capability. Philanthropy and sponsorship could support common dependencies, neglected tools and equitable access. Training and fellowships could strengthen local maintenance capacity.

These income streams serve different purposes. A short grant can finance a repair programme; a service agreement can support continued checks and response. The foundation would need multi-year planning, a diversified income base and explicit responsibilities when support expires. Open licensing would be preserved in line with each project's licence; institutions would pay for service and continuity, not exclusive access to the code.

Governance that connects money to responsibility

The proposed board and advisory structure should include scientific, engineering, institutional and community perspectives. Portfolio selection would consider scientific usefulness, dependency reach, unmet need, feasible validation and maintainer engagement. Funding terms and conflicts would be transparent; sponsors would not obtain scientific release authority.

Each supported project would have an agreed mandate: who owns the code, who can approve a release, which environments are supported, what response is expected and how handover or retirement works. Continuing responsibility requires budget for governance, coordination and reviewer time alongside technical work.

Measure the economics of validated maintenance

The relevant cost includes engineering, scientific review, inference, test compute and stewardship. Useful throughput means changes accepted with adequate evidence and continuing user benefit. Cheap patch generation can still be expensive if it creates review burden or regressions. A funding case should therefore report fully loaded costs and outcomes, separating routine tasks from scientifically consequential work.

Keep care continuous. Let research flourish.

Access means usable tools and lasting local capacity.



Illustrative metaphor from the pitch deck: ongoing care keeps software useful to researchers. It is not measured evidence of a particular growth effect.

An open repository does not automatically provide a usable research tool. A team may lack the time, engineering support or compute needed to repair an installation or adapt a workflow. The access programme would address these practical barriers while keeping scientific priorities with the people doing the research.

What support could include

Maintain locally important tools that fall outside the interests of large institutions; improve installation and documentation; help establish reference analyses; and adapt workflows for available hardware where feasible. Fellowships, mentoring and collaborative maintenance would build local expertise rather than make teams permanently dependent on an external service.

How access would be funded and governed

Institutional income and designated grants or sponsorship could finance supported access for partners with fewer resources. The foundation would publish how resources are allocated and involve those partners in priority setting. Subsidised support would carry the same scientific-review expectations as other work, with scope adapted transparently to the available budget.

The goal is reciprocal participation. Local teams can contribute domain knowledge, tests, user experience and maintenance expertise; they should retain ownership and authority. This proposed approach follows the collective-benefit and equity values in UNESCO's Recommendation on Open Science. [20]

09 / FROM AMBITION TO EVIDENCE

Build the institution. Prove the benefit.

A staged route from a Dutch portfolio to wider cooperation

Stage	Work to undertake	Evidence needed to expand
Netherlands: establish	Agree the founding mandate and anchor support. Build a representative multi-institutional software portfolio and baseline its condition.	Clear authority; funded service commitments; accepted changes with reference evidence; fully loaded delivery costs.
Europe: federate	Connect domain and RSE teams, share procedures and evaluation tasks, and establish cross-institutional service agreements.	Repeatable onboarding; reliable handovers; effective distributed review; demand beyond the initial institutions.
Global: widen access	Build partner-led access programmes, local maintainer development and practical support for under-resourced teams.	Useful research workflows; fair allocation; local ownership; sustained capacity and affordable support.

A compact outcome framework

Reliability: installation and reference-test success in specified environments, with denominators.

Scientific assurance: output comparisons, tolerances and expert decisions. **Continuity:** response, succession and handover against agreed responsibilities.

Efficiency: engineering and review time, inference and compute costs, accepted changes and regressions. **Research benefit:** partner use of supported workflows and downstream reuse. **Equity:** partner participation, supported access and local maintenance capability. Targets would follow agreed baselines and resources.

The decision for founding partners

The immediate task is to agree a mandate, assemble a funded portfolio and demonstrate that supervised agents plus expert review can deliver dependable continuity. Institutions can nominate tools and contribute engineering or scientific expertise. Funders can support the shared platform, rigorous evaluation and access programme. Software communities can shape priorities, acceptance criteria and governance.

The grant ends. The software should not.

REFERENCES AND EVIDENCE SUMMARIES

Sources 01-10

Verified through 2 October 2026. Links open the original publications or official pages.

[1] CBS (2026). Research and development; key figures per implementation sector, table 84644ENG. Updated 20 July 2026.

2024 totals and sector spending; revised provisional. Higher-education methodology changed in 2024, limiting comparisons with earlier years.

[Read source](#)

[2] Eurostat (2026). R&D expenditure, Statistics Explained. Data extracted September 2026.

EU R&D expenditure of EUR 408 billion for 2024. Broad economic context, not spending on research software.

[Read source](#)

[3] Trisovic A. et al. (2022). A large-scale study on research code quality and execution. Scientific Data 9, 60.

R-file re-execution failures before and after automated cleaning in a specified Dataverse corpus; not a global abandonment estimate.

[Read source](#)

[4] Samuel S. & Mietchen D. (2024). Computational reproducibility of Jupyter notebooks from biomedical publications. GigaScience 13, giad113.

Automated reproduction outcomes for notebooks linked to biomedical publications; explicit dependency and execution denominators.

[Read source](#)

[5] Virtanen P. et al. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature Methods 17, 261-272.

Historical February 2019 dependency snapshot illustrates the reach of a shared scientific library.

[Read source](#)

[6] Jensen E. A. & Katz D. S. (2025). Evolving funding strategies for research software: Insights from an international survey of research funders. PLOS ONE 20(8), e0329833.

Reports evolving support and challenges in long-term funding, community building and RSE careers. An uneven respondent sample, not a spending census.

[Read source](#)

[7] Netherlands eScience Center (2026). New Research Software Sustainability Programme: Open Science NL and the eScience Center join forces. 22 January.

Announces EUR 5.4 million and a remit covering maintenance, funding models, governance and recognition.

[Read source](#)

[8] Netherlands eScience Center (2026). Call for Open and Sustainable Research Software 2026 (OSS 2026).

Eight planned projects, primarily in-kind RSE support. Award decisions were scheduled for December 2026, after this paper's evidence cutoff.

[Read source](#)

[9] European Commission, CORDIS. EVERSE project fact sheet, grant agreement 101129744.

EU contribution of EUR 6,789,468.75 for March 2024-February 2027. Quality and sustainability work; not wholly maintenance delivery.

[Read source](#)

[10] EVERSE. About RSQKit: Research Software Quality Kit.

A source of research-software quality guidance for a complementary maintenance service.

[Read source](#)

REFERENCES AND EVIDENCE SUMMARIES

Sources 11-20

Verified through 2 October 2026. Links open the original publications or official pages.

[11] European Commission. Open science: policy, practices and enablers.

European policy explicitly includes software, reproducibility, infrastructure and skills.

[Read source](#)

[12] Netherlands eScience Center (2026). Research Software Directory Receives Funding from Open Science NL. 4 March.

Shows investment in software discovery, impact, curation and reproducibility connections.

[Read source](#)

[13] Software Heritage. Mission.

Source-code preservation provides a complementary function to maintaining supported executable tools.

[Read source](#)

[14] Open Source for Science Fund. Purpose and activities.

Funds development, maintenance and evolution, including automation and autonomous workflows; AI use alone is not differentiation.

[Read source](#)

[15] Renaissance Philanthropy (2026). Open Source for Science Fund Launches to Power AI-Driven Discovery.

The inaugural request for applications specifies USD 250,000 and USD 1 million grant ceilings for two tracks, not aggregate awards.

[Read source](#)

[16] Jimenez C. E. et al. (2024). SWE-bench: Can Language Models Resolve Real-World GitHub Issues? ICLR. arXiv:2310.06770.

Benchmark for repository issue resolution. Passing issue tests does not establish scientific validity.

[Read source](#)

[17] METR (2026). Time Horizon 1.1. 29 January.

Reports software-task capability trends and evaluation revisions, with uncertainty and task-composition limits.

[Read source](#)

[18] Becker J., Rush N., Barnes B. & Rein D. (2025). Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity. METR; arXiv:2507.09089.

A randomised study found a 19% slowdown in a specific experienced-developer setting; not a universal estimate of AI productivity.

[Read source](#)

[19] METR (2026). We are Changing our Developer Productivity Experiment Design. 24 February.

Later results suggested possible improvement but selection effects weakened inference about its size.

[Read source](#)

[20] UNESCO (2021). Recommendation on Open Science; values and guiding principles.

Normative basis for collective benefit, equity, inclusiveness and reciprocal access. It is not empirical evidence of a subsidy's effectiveness.

[Read source](#)

Preparation note: this revision builds on the project's evidence-base report, investment-case report, original concept pitch and OSC pitch deck v1. External factual claims were checked against primary publications and official programme pages. Illustrations are conceptual, not statistical findings. The institutional design and future benefits are proposals.