Practical Guide for Sustaining Open Source Software in the Research Enterprise

A Business Process Workflow and Decision Framework for Universities, Research Teams, and OSS Communities

Grounded in Findings from the Sustaining Open Source Software in the Research Enterprise Workshop

May 27, 2026

apereo

ITHAKA S+R

This guide builds off a report authored by Ithaka S+R and the Apereo Foundation:

McCracken, Chelsea, Dylan Ruediger, Mark McBride, Patrick Masson, Josh Baron. "Sustaining Open Source Software in the Research Enterprise: Findings from a One-Day Workshop." *Ithaka S+R*. March 25, 2026. https://doi.org/10.18665/sr.325189.

This material is based upon work supported by the National Science Foundation under Grant. No. 2512157. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

This material is based upon work supported by Alfred P. Sloan Foundation under Grant No. G-2025-25223.

This material is based upon work supported by a gift from the Chan Zuckerberg Initiative.

Purpose of the guide

Open Source Research Software (OSRS) is now central to discovery, collaboration, reproducibility, and interdisciplinary innovation across the research enterprise. Yet OSRS remains fragile: maintenance is undervalued, incentives misaligned, discoverability inconsistent, and institutional responsibility unclear. Many projects disappear within their first few years due to a lack of sustainability planning.

This practical guide provides:

- **A workflow** for researchers and institutions to identify sustainability needs and select appropriate models.
- **A logic model/decision tree** that visually maps sustainability pathways.
- **Concrete actions** for researchers, administrators, OSRS developers and communities, Open Source Program Offices (OSPOs), Tech Transfer Offices (TTOs), libraries, consortia, and funders.
- **Embedded insights, examples, and quotations** from the workshop's findings.

Summary of key challenges the guide addresses

Five challenges were identified in the workshop:

1. What is the value proposition for universities? Why should they prioritize OSRS?
2. How can we know who is or could be using an OSRS product?
3. How can we overcome researchers' cultural barriers to open culture?
4. Who should be responsible for sustaining OSRS?
5. What extra-institutional support is needed to sustain OSRS?

This guide offers a decision process and sustainability models that directly respond to each challenge.

Core sustainability dimensions

Workshop participants surfaced four interdependent dimensions that together define OSRS sustainability:

1. **Motivational Sustainability**

   Sustainability begins with **the goals of the project and the goals of the people** involved.

   - Sustainability only has meaning in the context of the software's goal.
   - Many OSRS tools are "purpose-built Q-tips"—not meant to be sustained beyond the immediate scientific need. Software should not be sustained simply for its own sake.
   - Motivation varies: fun, profit, learning, academic rewards, or real-world impact. Each carries different sustainability risks.

2. **Infrastructural Sustainability**

   OSRS is **core research infrastructure**—as essential as labs, buildings, and utilities.

   - Many researchers use OSS without ever being aware that someone needs to maintain it.
   - Infrastructure requires continuous, reliable funding streams—not episodic grants.

3. **Archival Sustainability**

   Even if code is not actively maintained, it must be **citable, discoverable, and preserved**.

   - A citation is required so that software can fit into existing systems for academic incentives. Also, according to the principles of open science, preservation of the code that produced certain findings is similarly important to preservation of the data.
   - Discoverability is critical for the reproducibility of science.

- Preservation and discoverability enable future researchers to identify relevant/useful code sources that can be adopted and modified for new research endeavors.
- Preservation in the scholarly record includes documentation, provenance, licensing, and archiving in an accessible location.

4. Relational Sustainability

OSRS communities must be **governed, connected, and supported**.

- People are the most essential feature for ensuring project sustainability.
- Theoretically, OSRS could be sustained as a by-product of other institutional work (ongoing research, as a teaching and learning methodology, upstream/downstream of administrative computing).
- Communication, onboarding, mentoring, shared governance, and attention to culture are all central.

Business process workflow: Practical guide for sustaining OSRS

STEP 1 — Define purpose and intended impact of the software

Key Question: *What is this software trying to be?*

Actions:

1. Identify the archetype of the project (e.g., research tool, platform, library, infrastructure).
    a. One resource identified by SOSSRE workshop participants to assess/align purpose and projects was "Open Source Archetypes: A Framework for Purposeful Open Source (version 2.0)," Open Tech Strategies and Mozilla, 2019, https://opentechstrategies.com/archetypes-files/open-source-archetypes-v2.pdf.
2. Assess expected user community size: single-use, niche, discipline-wide, or cross-disciplinary. While formal tools specific to market interest may not exist, several "desktop research" efforts using industry-recognized resources may help assess potential applicability and interest.
    a. Stack Overflow Trends and Developer Surveys
        i. Stack Overflow Developer Survey (https://survey.stackoverflow.co/)
        ii. Stack Exchange Data Explorer (https://data.stackexchange.com/)
        iii. Search tags related to your idea (e.g., rust, bioinformatics, GraphQL); assessing:
            1. Question volume over time
            2. Unanswered questions (indicates unmet need)
        iv. Signal strength:
            1. High volume = large ecosystem
            2. Low but consistent = niche

3. Very low = likely single-use, early, or experimental

b. GitHub Inspiration and Metrics
   i. Github Innovation Graph (https://innovationgraph.github.com/)
   ii. Github Topics (search directly, e.g., github.com/topics/<your-topic>)
   iii. Consider stars, forks, contributors of similar projects, and growth trends (stars over time).
   iv. Signal strength:
      1. 10k+ stars across multiple projects = large ecosystem
      2. 1k - 10k = niche
      3. <100 = likely single-use, early, or experimental

c. Google Search and Trends
   i. Advanced Search (https://www.google.com/advanced_search). Or search directly, e.g., "<your topic 1>" "<your topic 2>" site:.edu
   ii. Trends (https://trends.google.com/trends/)
   iii. Identifies related projects, communities, and organizations.
   iv. Measures search interest over time
   v. Signal strength:
      1. Multiple projects/communities, active through time = large ecosystem
      2. Some projects, active through time = niche
      3. Some projects, inconsistently active through time = likely single-use, early, or experimental

3. Determine desired lifespan: short-term (one project cycle), medium-term, long-term infrastructure.
4. Clarify motivation: scientific impact, novelty, funding, market viability, student training, etc.

Decision Point → **Select a Sustainability Path and go to STEP 2**

- **Archive-only Path** (Q-tip software)
- **Stewardship Path** (moderate maintenance for transparency and reproducibility)
- **Community Path** (active multi-user ecosystem)
- **Infrastructure Path** (mission-critical or widely adopted)

STEP 2 — Assess human capacity and governance needs

Key Question: *Who supports this software?*

Actions:

1. Identify potential/current core maintainers and contributors, their roles (e.g., faculty, students, postdocs, RSEs, external developers), and affiliations (e.g., campus, institution, company).
    a. Data from STEP 1 can help identify peers, colleagues, and other relevant potential contributors.
    b. Signal strength:
        i. Multiple local/affiliated contributors = strong support environment
        ii. Some local, potential affiliated contributors = limited support environment
        iii. Few local, unknown affiliated contributors = low/no support environment
2. Identify resources (infrastructure, funding, non-technical collaborators, existing code/projects/communities, intellectual property, aligned non-technical organizations, etc.)
    a. Again, STEP 1 may provide relevant leads for identifying non-staffing resources.
    b. Signal strength:
        i. Multiple local/affiliated resources = strong support environment
        ii. Some local, potential affiliated resources = limited support environment
        iii. Few local, unknown affiliated resources = poor support environment
3. Evaluate turnover risks, “continuous cycle of new people” (e.g., students) in academia.
    a. Assess the roles of your potential community contributors and collaborators:
        i. Signal strength:
            1. Primarily long-term faculty and staff (local and affiliated FTE) = strong support environment
            2. Few long-term faculty and staff (local and affiliated FTE), primarily students (local and affiliated, enrolled or grant-funded) and fixed-

term faculty and staff = limited support environment
3. Primarily students (local and affiliated, enrolled or grant-funded) = poor support environment

4. Create or adopt governance models:
    a. Decision-making process, see:
        i. Fogel, Karl. "Social and Political Infrastructure." In *Producing Open Source Software: How to Run a Successful Free Software Project* (Second Edition). 26 September 2024, https://producingoss.com/en/producingoss.html#social-infrastructure.
        ii. Bacon, Jono. “Governance.” In *The Art of Community* (Second Edition). O’Reilly, 2012, https://www.jonobacon.com/wp-content/uploads/2019/01/jonobacon-art_of_community_second_edition.pdf.
        iii. Neary, Dave, Josh Berkus, and Bryan Behrenshausen. “Project and Community Governance.” In “*The Open Source Way*. Edited by Brian Proffitt , et al. Red Hat, 2025, https://producingoss.com/en/producingoss.html#social-infrastructure.
    b. Succession planning
        i. “Practitioner Guide: Getting Started with Contributor Sustainability” CHAOSS. Linux Foundation, 2026, https://chaoss.community/practitioner-guide-contributor-sustainability/.
    c. Codes of conduct
        i. Fogel, Karl. "Codes of Conduct." In *Producing Open Source Software: How to Run a Successful Free Software Project* (Second Edition). 26 September 2024,“ https://producingoss.com/en/producingoss.html#code-of-conduct.
        ii. Maffeo, Lauren. “Step Three: Create and Enforce a Clear Code of Conduct.” In “*The Open Source Way.* Edited by, Brian Proffitt , et al. Red Hat, 2025, https://guidebook.theopensourceway.org/attracting-users/building-diverse-open-source-communities-by-

making-them-includive-first#step-three-create-and-enforce-a-clear-code-of-conduct.

Decision Point → **Is the team stable enough to maintain the project?**

- If **Yes** → proceed to Step 3.
- If **No (Undetermined)** → seek support:
  - Campus resources: local OSPO (if one exists), campus library, Tech Transfer Office, etc.
  - Open source software foundations related to your project's scope, core technology(ies), academic domain, etc.: CURIOSS (https://curioss.org/), SustainOSS: Open Source in Academia Map (https://sustainoss.org/academic-map/intro.html), FOSS Foundations Directory (https://fossfoundation.info/listing), Google Supported Organizations (https://opensource.google/organizations-we-support).
  - External community incubator or consultants

STEP 3 — Document, license, cite, and make discoverable

Key Question: *Are your practices mature, can new contributors engage?*

Actions:

1. Ensure code is findable:
   a. Package Registries: Publishing in these and similar ensures projects show up in developer workflows (search, installs, dependency graphs). Examples include Npm (https://www.npmjs.com/), PyPI (https://pypi.org/), RubyGems (https://rubygems.org/), crates.io (http://crates.io), Docker Hub (https://hub.docker.com/).
   b. Code repositories: Publicly accessible code repositories provide environments to expose code, issues, and roadmaps to peers in technology and/or disciplines: Examples include ASF Bitbox, Bitbucket, Codeberg, Forgejo, Gitea, GitHub, Gitlab, invent.kde.org, Launchpad, Savannah, SourceForge, etc.

    c. Your choice should align with your vision for the project; different project priorities (e.g., maximum visibility, enterprise workflows, open source values/nonprofit, self-hosting, decentralization, legacy/distribution, and advanced dev workflows) align with different code repositories.
2. Ensure code is citable:
    a. Publish software in venues that recognize code as scholarship (in addition to discipline-specific publications). For example:
        i. Journal of Open Source Software
        ii. SoftwareX
        iii. Journal of Open Research Software
    b. Practices
        i. Include a CITATION.cff file in your repository.
        ii. Assign a DOI to releases (Zenodo, Figshare)
        iii. Adopt clear versioning such as v1.2.0
        iv. Add a “How to Cite” section in the project’s README.md
        v. License clearly and appropriately: OSI Approved Open Source License (code), Creative Commons Licenses (non-code, text, graphics)
3. Document purpose, governance, and usage.
    a. Fogel, Karl. "Documentation." In *Producing Open Source Software: How to Run a Successful Free Software Project* (Second Edition). 26 September 2024, https://producingoss.com/en/producingoss.html#documentation.
    b. Hawthorne, Leslie. “Step Three: Communication Norms in Open Source Projects.” In In “*The Open Source Way.* Edited by Paula Dickerson, et al. Red Hat, 2025, https://guidebook.theopensourceway.org/attracting-users/communication-norms-in-open-source-software-projects.
4. Select an appropriate open source license. Due diligence dictates that founders of projects initiated through/with an institutional relationship consult the appropriate offices, for example, the campus’s Office of the General Counsel and/or the Technology Transfer Office/IP Transfer Office. In order to prepare for such discussions, a review of potential open source software licenses would be helpful.

a. Open Source Initiative's Approved License List (https://opensource.org/licenses)
b. Creative Commons, Choose a License for your work (https://creativecommons.org/chooser/)
c. Carnegie Mellon University, Open Source License Comparison Grid (https://www.cmu.edu/cttec/forms/opensourcelicensegridv1.pdf)
d. Choose a License (https://choosealicense.com/)
e. Github License Selector (https://ufal.github.io/public-license-selector/)

5. Deposit code in a discoverable, persistent archive.
    a. Software Heritage (https://www.softwareheritage.org/) *referenced by workshop attendees*
    b. OpenAIRE (https://www.openaire.eu/)
    c. Re3data (https://www.re3data.org/)

Decision Point → **Does the project require active maintenance?**

- If **Yes** → proceed to Step 4.
- If **No** → Follow **Archival Sustainability Path**.

STEP 4 — Evaluate funding pathways

Key Question: *Does the project need funding, and where might funding be obtained?*

Actions:

1. Identify required maintenance levels (minimal, moderate, high). Several tools are available to assess a new project's demands based on existing examples/references from peer projects.
    - CHAOSS
        i. Metrics and Metrics Models (https://chaoss.community/kb-metrics-and-metrics-models/)

        ii. Software, eg., Augur, GrimoireLab (https://chaoss.community/software/)
    - Multiple commercial options (https://github.com/topics/analytics)
2. Determine if the project is a "by-product" of other funded operations.
    - When an open source software project emerges as a by-product of broader, funded university operations, it often does not require dedicated funding of its own. Instead, the resources needed to sustain and evolve the project—such as developer time, infrastructure, and maintenance—are naturally supported through existing budgets tied to administrative computing, academic programs, student services, and research initiatives. In this model, the software benefits from ongoing institutional investment in these areas, allowing it to grow organically without the overhead of securing separate funding streams, while still delivering value both within and beyond the university.
3. Determine if funding is available/possible for all aspects of the project.
    - Apereo Foundation OSS Grant Funding Prioritization Framework
    - Apereo OSS Funding Tracker
4. Consider alternatives to traditional grant-driven funding. "Business Models for Free and Open Source Software, "by Chitu Okoli and An Nguyen (SSRN, 2015, https://dx.doi.org/10.2139/ssrn.2568185) includes a variety of revenue models and, critically, their impact on both the project, developer, and end-user communities.
    - Funding OSS as infrastructure (Institutional Cost-Sharing/ Pooled Funding)
    - Service-Based Revenue (Support, Hosting, SLAs)
    - Membership & Consortium Models (Core + Tiered)
    - Corporate Partnerships & Sponsorships
    - Subscription-Based "Open Core Services" (Without Closing the Core)
    - Training, Certification, and Workforce Development
5. Choose from multiple organizational models:
    - Institutional (e.g., campus OSPO, library, department, research office)
    - Consortial (e.g., multi-institutional or discipline, academic or working groups)

- Foundational: (e.g., create or join a 501c3 or c6)
- Endowment/Collectives (new/emerging models as of 2026)
- Industry partnerships
- National or state agencies

Decision Point → **Is there long-term funding?**

- If **Yes** → proceed to Step 5.
- If **No** → shift to hybrid sustainability, reduce scope, or move toward archival path.

STEP 5 — Understand usage and impact

Key Question: *How will the project identify users and measure impact?*

Actions:

1. Use indirect measures where direct measurement (e.g., software and user tracking) is incompatible with OSS values:
    - Publications citing the software
    - Dependency analysis
    - Code repository metrics (e.g., downloads, stars, pull requests, issues submitted, contributors, forks, etc.)
    - Stories and use cases from the community
    - Community engagement (discussion forums, emails)
    - Participation (new committers, attendees at events/activities)
2. Adopt tools such as:
    - As suggested by participants in the SOSSRE Workshops
        i. BuiltWith
        ii. Scarf (privacy-preserving usage analytics)
    - Software Bill of Materials (SBOM) to map dependencies
    - Voluntary, inner, institution-wide port scans (in partnership with IT)

Decision Point → **Is there evidence of meaningful adoption?**

- If **Yes** → justify investment and move to Step 6.
- If **No** → consider repositioning, outreach, or winding down the project.

STEP 6 — Integrate project into institutional structures

Key Question: How will project sponsors/maintainers institutionalize the project and community?

Actions:

1. Determine which institutional unit(s) own stewardship responsibilities.
2. Determine which institutional unit(s) are dependent on or contribute to the OSRS.
    - Note: contributors from other institutions involved in the development and/or maintenance of the OSRS should be included.
3. Provide awareness and onboarding materials (as well as marketing and promotional) for faculty, students, staff, and administrators of current and potential adopting organizations.
4. Integrate OSRS into:
    - Research compliance workflows
    - Student training and work-study programs
    - Institutional PR and reputational strategies
5. Ensure alignment with institutional mission, i.e., research, teaching, public service.

Decision Point → **Are the institution(s) or sufficient units prepared to steward OSRS?**

- If **Yes** → proceed to Step 7.
- If **No** → escalate to research and or enterprise-level support.

STEP 7 — Connect to external ecosystems

Key Question: *What external relationships can be developed to increase resilience?*

Actions:

1. Identify multi-institution consortia, societies, and foundations aligned with your software domain.
2. Explore **Foundation-as-a-Service** models such as Code for Science & Society or Lyrasis.
3. Encourage participation in global OSS communities and standards groups.
4. Develop interoperability with other tools to increase value and reach.

Decision Point → **Is the project positioned within a sustainable ecosystem?**

- If **Yes** → Probability for long-term sustainability maximized.
- If **No** → Revisit earlier steps (funding, governance, adoption, archiving).

Logic model / Decision tree for OSRS sustainability

Key Question: *Should this project be actively sustained, minimally maintained, or archived?*

Decision tree structure

1. **What is the project's purpose?**
    - Single-use → **Archive Path**
    - Multi-user or foundational to ongoing research → go to step 2

2. **Who are the maintainers?**
    - Stable team with governance → go to step 3
    - High turnover or unclear roles → **Institutional Support Path**

3. **Is the software discoverable, documented, and citable?**
    - Yes → go to step 4
    - No → **Archival Preparation Path**

4. **Is there sustained funding or institutional commitment?**
    - Yes → go to step 5
    - No → **Hybrid/Minimal Maintenance Path**

5. **Are usage metrics or community signals strong?**
    - Yes → **Infrastructure / Community Sustainability Path**
    - No → **Reassessment Path**

6. **Does the institution recognize OSRS as mission-aligned?**
    - Yes → **Institutional Integration Path**
    - No → **Value Proposition Advocacy Path**

7. **Are there external ecosystems to integrate with?**
    - Yes → **Consortium / Foundation Path**
    - No → **Develop Partnerships Path**

Sustainability models for OSRS

A. Archival sustainability (minimum viable model)

a. Document, license, archive.
b. Achieves transparency and reproducibility.
c. For Q-tip or one-time-use software.

B. Stewardship model (low/moderate maintenance)

a. Department, lab, OSPO, library, or other unit provides light support.
b. Works for tools with small but consistent user bases.

C. Community co-maintenance model

a. Distributed contributions; governance shared.
b. Best when community engagement is strong.

D. Institutionally supported infrastructure model

a. OSPO, TTO, central IT, or library anchors sustainability.
b. Ideal for widely used or high-impact tools.

E. Multi-institution consortium model

a. Shared governance, funding, and evangelism.
b. Enhances resilience and adoption.

F. Foundation-as-a-service model

a. Outsource governance, accounting, legal, and community operations.
b. Frees researchers to focus on science.

G. Endowment/continuous funding model

a. Percent of grants or institutional budget supports maintenance.
b. Provides strongest long-term stability.

Roles and responsibilities across the research enterprise

Researchers

- Define purpose and sustainability goals.
- Document, cite, archive; engage users.
- Advocate within departments.

Students (key workforce pipeline)

- Perform maintenance tasks; gain skills.
- Benefit from apprenticeship-like experiences.

Open Source Program Offices

- Provide governance, licensing, and sustainability planning.
- Serve as an institutional connector and community hub.

Libraries

- Steward archives, metadata, discovery systems.
- Provide training and preservation workflows.

Technology (IP) Transfer Offices

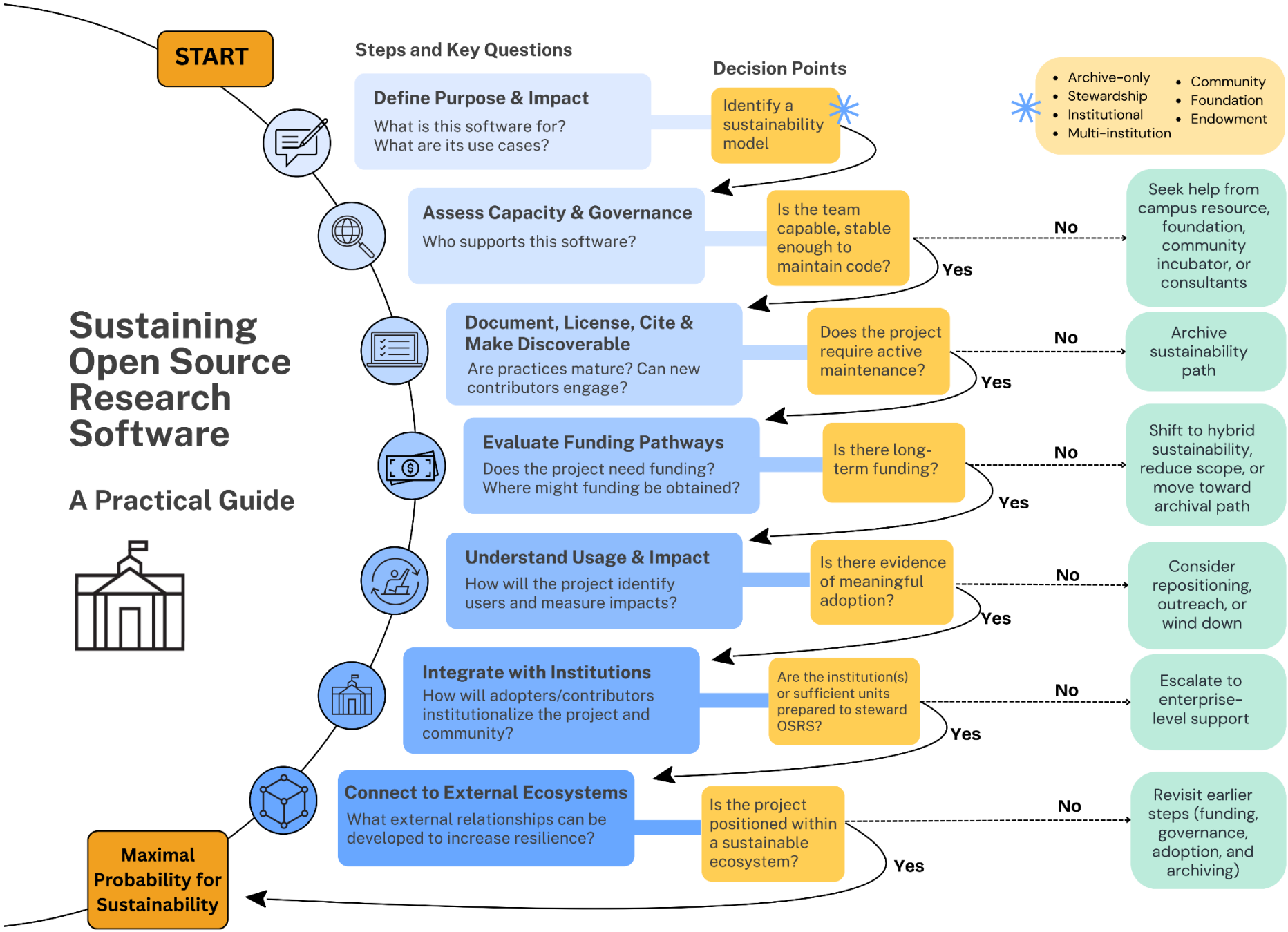
- Assist with legal, licensing, and business models.
- Adjust traditional commercialization frameworks to support OSRS.

Funders

- Require sustainability plans; support maintenance.
- Consider endowment-like approaches.

External partners (societies, foundations, NGOs)

- Provide training, consulting, matchmaking, tooling, evaluation.
- Help build sector-wide standards, e.g., “nutrition labels,” maturity models.


START
Steps and Key Questions
Decision Points
Define Purpose & Impact
What is this software for?
What are its use cases?
Identify a sustainability model
Archive-only
Stewardship
Institutional
Multi-institution
Community
Foundation
Endowment
Assess Capacity & Governance
Who supports this software?
Is the team capable, stable enough to maintain code?
No
Seek help from campus resource, foundation, community incubator, or consultants
Yes
Document, License, Cite & Make Discoverable
Are practices mature? Can new contributors engage?
Does the project require active maintenance?
No
Archive sustainability path
Yes
Evaluate Funding Pathways
Does the project need funding?
Where might funding be obtained?
Is there long-term funding?
No
Shift to hybrid sustainability, reduce scope, or move toward archival path
Yes
Understand Usage & Impact
How will the project identify users and measure impacts?
Is there evidence of meaningful adoption?
No
Consider repositioning, outreach, or wind down
Yes
Integrate with Institutions
How will adopters/contributors institutionalize the project and community?
Are the institution(s) or sufficient units prepared to steward OSRS?
No
Escalate to enterprise-level support
Yes
Connect to External Ecosystems
What external relationships can be developed to increase resilience?
Is the project positioned within a sustainable ecosystem?
No
Revisit earlier steps (funding, governance, adoption, and archiving)
Yes
Maximal Probability for Sustainability
Sustaining Open Source Research Software
A Practical Guide

Sustaining Open Source Research Software

Steps and Action Items

1 2 3 4 5 6 7

1 Define Purpose & Impact

- Identify the archtype of the project
- Assess expected user community size
- Determine desired lifespan
- Clarify motivation

2 Assess Capacity & Governance

- Identify potential/current core maintainers and contributors
- Identify resources
- Evaluate turnover risks
- Create or adapt governance model

3 Document, License, Cite & Make Discoverable

- Ensure code is findable
- Ensure code is citable
- Document purpose, governance, and usage
- Select an appropriate open source license
- Deposit code in a discoverable archive

4 Evaluate Funding Pathways

- Identify required maintenance levels
- Determine if the project is a 'by-product' of other funded operations
- Determine if funding is avilable
- Consider alternatives to traditional grant-driven funding
- Choose from multiple organizational models

5 Understand Usage & Impact

- Use indirect measures where direct measurement is incomptabile with OSS values
- Identify and adopt assessment tools

6 Integrate with Institutions

- Determine which institutional unit(s) own stewardship responsibilities
- Determine which institutional unit(s) are dependent on or contribute
- Provide awareness and onboarding materials
- Integrate OSRS into research compliance workflows, student programs, PR strategies
- Ensure alignment with institutional mission

7 Connect to External Ecosystems

- Identify consortia, societies, and foundations aligned with your software domain
- Explore Foundation-as-a-Service models
- Encourage participation in global OSS communities
- Develop interoperability with other tools to increase value and reach