SUSTAINABILITY GUIDE FOR FLOSS TOOLS

Internews
Local voices. Global change.

PURPOSE OF THE GUIDE

The "Sustainability Guide for FLOSS Tools" is a straightforward and brief guide made for developers who work with Free/Libre/Open Source Software (FLOSS). It gives useful tips and ideas on how developers can secure the enduring success of their tools.

This guide identifies key areas that are essential for the sustainability of FLOSS tools. It brings together knowledge and advice from different areas to give a full picture of what sustainability means in the context of FLOSS and outlines the necessary steps to address these areas, guiding developers towards a variety of reliable online resources that offer more in-depth information. This is particularly important as the landscape of free and open source software continues to change and grow.

The guide does not just help with immediate needs but also adds to the broader discussion about keeping FLOSS projects healthy and viable. By providing developers with necessary knowledge and tools, this guide aims to play a key role in supporting a strong and thriving free and open source community.

This guide was developed as part of Internews' two-year initiative to support six Free/Libre/Open Source Software (FLOSS) tool teams in their efforts to develop and implement sustainability action plans. Building on the lessons learned from this project, desk research was used to compile the information and resources provided throughout this report, using credible sources from experts in the field. Due to the rapidly developing nature of the FLOSS environment, this report prioritizes recently published materials, and older sources are referenced only where they remain applicable. This guide was originally published in January 2024 and has been updated with additional content based on community feedback and lessons from the remainder of the project.

This report was written by Amira Galal, Cecilia Mwende Maundu, Joey Salazar, Skyler Sallick, and Marc Shaffer of the Internews Internet Freedom and Resilience team and designed by Skyler Sallick.

June 2024

This content is available as a CC-BY 4.0 International license.

Table of Contents

FORWARD

In a relentlessly digital world, open source software is a testament to the incredible power of collaboration, inclusivity, and shared knowledge on a global scale. There are countless examples of how free, libre, and open source development has spearheaded innovation and, at times, provided solutions to problems when commercial applications could not. The success of projects like Linux in the realm of operating systems showcases how open source can offer not only free alternatives but also transparent and highly customizable solutions for users. Similarly, Mozilla Firefox revitalized browser competition, prioritizing user privacy and offering a faster, more secure browsing experience – giving rise to Tor Browser. And OpenSSL has played a crucial role in enhancing online security by providing a widely adopted solution for secure communication on the Internet.

Amid today's ever-increasing push across industries to innovate and digitize seemingly all aspects of our lives and identities, there is a renewed urgency for open source software to rise to the occasion– not just solving global challenges, but also defending people's interests: from safeguarding digital privacy, freedom of expression, and access to information, to giving rise to decentralized financial systems and emerging currencies to name just a few. We are seeing, in real time, the rise of new transformative technologies such as AI, alongside alarming geo-political events that expose the vulnerability of users to exponentially more powerful providers in an overly centralized digital ecosystem. At a moment's notice, corporate and government interests can change, and open source software is the last line of defence offering alternatives that give power back to users. Open source projects have an incredible track record of tackling complex issues with collective expertise.

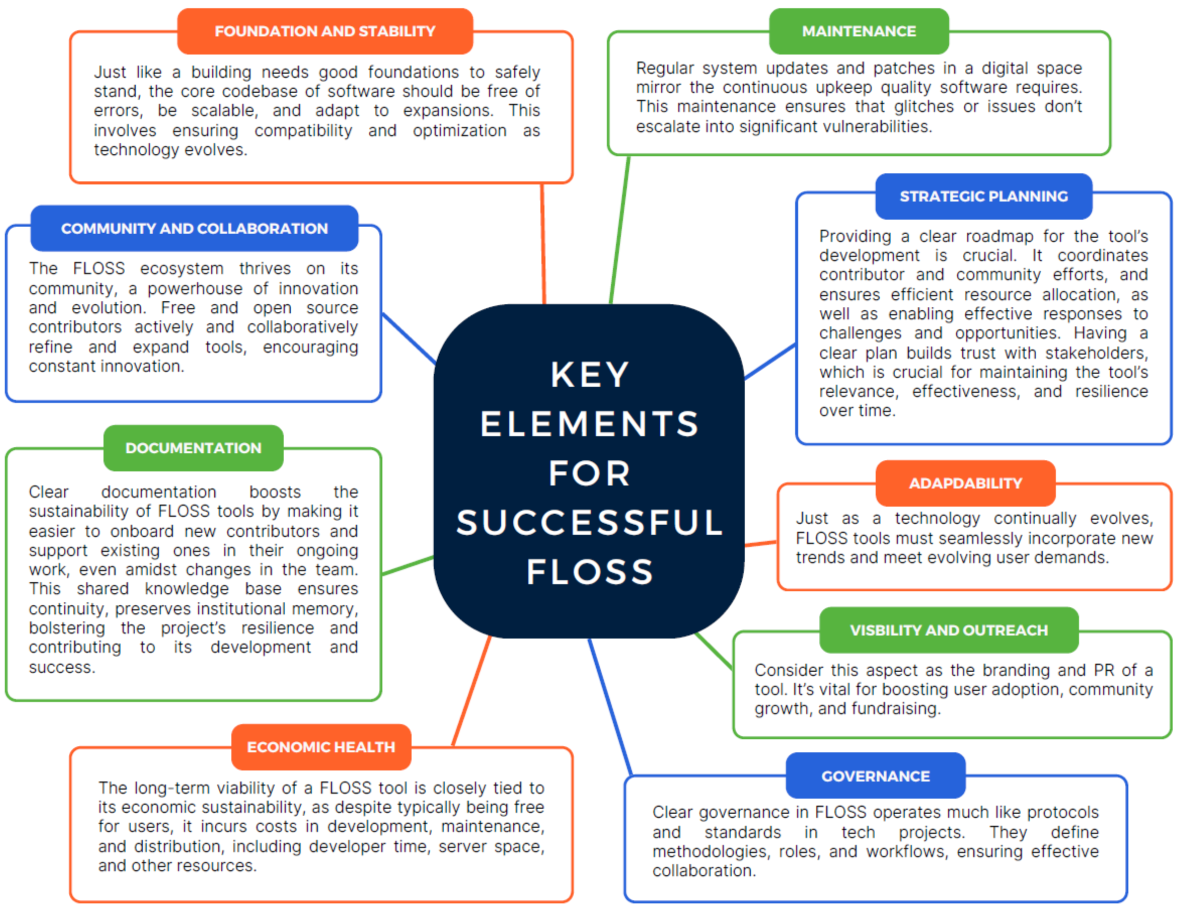
At the Tor Project, we know from first-hand experience that better digital tools are built with a collective vision, effort, and power. We are also looking back on more than two decades of maturation as an organization. This Sustainability Guide for FLOSS Tools manages to condense a lot of this kind of practical experience into a concise and easy-to-digest document. Laying out critical organizational elements to sustaining open source tech ecosystems and operations, it serves both as a practical-oriented cheat sheet for developers and maintainers, and as an advocacy tool for contributors and community members. We are certain that this guide will contribute to supporting future innovation in the FLOSS community and sustain and grow the impact of this technology in people's lives across the globe.

The Tor Project

Section 1: Defining FLOSS Sustainability

Sustainability in FLOSS Explained

In the complex and diverse landscape of Free/Libre/Open Source Software (FLOSS), grasping the long-term viability of each FLOSS tool is crucial for both the tool's sustainability and the ongoing stability of the broader community. Each tool, be it an expansive data storage solution or a complex network configuration, plays a pivotal role in the ecosystem. Their sustainability ensures that these tools not only meet current needs but can also adapt and evolve with emerging technological trends. As many organizations and individual developers rely on these FLOSS tools, their long-term viability and adaptability become essential for continuous innovation, reliability, and overall progression of the free and open source movement.


KEY ELEMENTS FOR SUCCESSFUL FLOSS
FOUNDATION AND STABILITY
Just like a building needs good foundations to safely stand, the core codebase of software should be free of errors, be scalable, and adapt to expansions. This involves ensuring compatibility and optimization as technology evolves.
MAINTENANCE
Regular system updates and patches in a digital space mirror the continuous upkeep quality software requires. This maintenance ensures that glitches or issues don't escalate into significant vulnerabilities.
COMMUNITY AND COLLABORATION
The FLOSS ecosystem thrives on its community, a powerhouse of innovation and evolution. Free and open source contributors actively and collaboratively refine and expand tools, encouraging constant innovation.
STRATEGIC PLANNING
Providing a clear roadmap for the tool's development is crucial. It coordinates contributor and community efforts, and ensures efficient resource allocation, as well as enabling effective responses to challenges and opportunities. Having a clear plan builds trust with stakeholders, which is crucial for maintaining the tool's relevance, effectiveness, and resilience over time.
DOCUMENTATION
Clear documentation boosts the sustainability of FLOSS tools by making it easier to onboard new contributors and support existing ones in their ongoing work, even amidst changes in the team. This shared knowledge base ensures continuity, preserves institutional memory, bolstering the project's resilience and contributing to its development and success.
ADAPDABILITY
Just as a technology continually evolves, FLOSS tools must seamlessly incorporate new trends and meet evolving user demands.
VISBILITY AND OUTREACH
Consider this aspect as the branding and PR of a tool. It's vital for boosting user adoption, community growth, and fundraising.
ECONOMIC HEALTH
The long-term viability of a FLOSS tool is closely tied to its economic sustainability, as despite typically being free for users, it incurs costs in development, maintenance, and distribution, including developer time, server space, and other resources.
GOVERNANCE
Clear governance in FLOSS operates much like protocols and standards in tech projects. They define methodologies, roles, and workflows, ensuring effective collaboration.

The Importance of Longevity, Support, and Community in FLOSS

Free/Libre/Open-Source Software (FLOSS) thrives on the principles of collective collaboration and shared value. Three pillars are central to its success: longevity, support, and community.

- **Longevity**: Beyond just its temporal existence, longevity is about staying relevant, adaptable, and non-obsolete in a fast-evolving tech world. This involves ensuring ongoing value, open access, and rights to use, modify, and distribute software across generations.

- **Support**: Support in FLOSS extends beyond technical help to a continuous exchange of knowledge and expertise. This collaboration fosters innovation and keeps the ecosystem vibrant, benefiting both experienced developers and newcomers.

- **Community**: The FLOSS community is more than a user group; it is the movement's core. It promotes transparency, participatory development, and democratic values, countering monopolistic trends in tech by prioritizing innovation and ethics over profit.

FLOSS is not just about open access to software. It is a vibrant mix of long-lasting relevance (longevity), collaborative problem-solving (support), and a strong, active community. These pillars not only underpin the software but also uphold the philosophy that underpins the FLOSS movement, highlighting the importance of *freedom*, *active collaboration*, and *collective advancement* in tech.

Section 2: Long-Term Planning

Sustaining a FLOSS project over time requires careful planning and management. By setting clear roadmaps, managing deprecation and version control effectively, and considering backward compatibility, project maintainers can ensure a robust and enduring project. The key to sustainability lies in balancing progress with stability and community engagement with clear, transparent communication.

Project Roadmaps

A project roadmap is crucial to FLOSS sustainability as it provides a clear, strategic direction, ensuring that contributors and users are aligned towards common goals and milestones. It also helps in managing expectations, fostering trust and engagement within the community, which is vital for the long-term vitality and growth of the project.

Suggested Approaches

- **Vision and Goals**: Define the long-term vision and specific goals for your FLOSS project. This helps in aligning the team and the community towards a common objective.

- **Milestones and Timelines**: Break down the project into manageable milestones with realistic timelines. This provides a structured approach and helps in tracking progress.

- **Community Involvement**: Engage with the community regularly. Gather feedback and incorporate it into the roadmap. This ensures the project remains relevant and community driven.

- **Transparency**: Maintain transparency in the roadmap development process. Publicly share updates and changes to keep the community informed and involved.

Resources

- The Open Source Way Guidebook: A manual on managing open source projects, focusing on vision, governance, and leadership.
- Setting Open Source Strategies: An article on developing and measuring open source strategies, addressing a project's purpose, goals, and desired end state.

Sustainability Action Plans

In addition to the creation of a project roadmap, a Sustainability Action Plan (SAP) can help developers work together to accomplish their sustainability goals and ensure the long-term success of their tool. SAPs are developed for a tool team's internal use and serve to center sustainability within all aspects of the project to maintain a focus on sustainability in all decisions.

Suggested Approaches

- **Align SAP with Roadmap:** SAPs should complement the long-term goals identified in the project roadmap with an additional focus on sustainability.
- **Think Holistically:** As laid out in this guide, sustainability means more than financial security and technological development. SAPs should take all aspects of sustainability into account, moving beyond fundraising and grant opportunities. Areas of focus may include:
    - How to attract and retain core contributors over time,
    - Governance,
    - Licensing and intellectual property,
    - How to diversify the community,
    - Marketing strategies,
    - And potential options for legal structures to receive funds and protect the tool.
- **Collaborate with Peers, Mentors, and Experts:** Working with other FLOSS teams and mentors can ensure that identified goals and milestones are relevant, measurable, and obtainable.

SUSTAIN's SAP Development Process

As part of the SUSTAIN project, members of the tool team cohort collaborated with Internews and Code for Science & Society to develop and execute SAPs to address their sustainability goals. Tool teams engaged with several key processes that might be helpful for your tool to consider when developing your own sustainability plan.

1. **Conduct an internal brainstorm:** Tool teams began developing their SAPs by internally considering their sustainability goals and needs before collaborating with the rest of the project partners.
2. **Consider the bigger picture:** CS&S hosted workshops with the cohort that asked teams to consider important questions that go beyond daily maintenance tasks and crisis management, re-centering themselves and their relationship to their tool. Teams were encouraged to consider why they do this work, why their tool is important to themselves and the community it serves, and what would happen if the tool ceased to exist.

3. **Receive support to connect bigger picture ideas with internal brainstorm:** Tool teams then worked with CS&S to incorporate lessons learned through the workshops with the specific needs identified in the internal brainstorm.

4. **Draft the initial SAP:** Tool teams then developed customized SAPs that included priorities, milestones, and long term goals, as well as considered the potential challenges.

5. **Determine necessary resources:** Tool teams then worked with Internews to determine what would be needed to execute their SAP, including what outside expertise and material requirements would be necessary.

6. **Finalize the SAP:** After building their initial plan and determining the necessary resources, tool teams reviewed their draft SAPs and finalized their milestones and timeframe.

7. **Implement the SAP:** Using support provided by Internews and external consultants, tool teams executed their SAPs to accomplish their sustainability milestones.

8. **Utilize mentorship:** CS&S paired each tool team with a mentor that could provide advice on executing their SAP and help teams consider what sustainability might look like beyond the lifespan of the SUSTAIN project.

9. **Revise and update the SAP:** As tool teams executed their SAPs, they revisited the document to make any needed adjustments. SAPs are living documents, and teams were encouraged to continue adjusting their approach as they learned new lessons. Tool teams are able to build out their next series of milestones and sustainability goals.

Resources

- Code for Science & Society: CS&S is a non-profit organization, and SUSTAIN project partner, that works towards a future where research, data, and tech initiatives shift power to communities. They partner with the FLOSS community to advance equitable access to knowledge and education and to explore community governance, develop participatory funding programs, and explore new ways to center power within communities.
    - Under their Digital Infrastructure Incubator, CS&S created a cohort of tools, developing a shared curriculum that allows teams to develop sustainability practices with mentors and peers. Under the SUSTAIN project, Code for Science & Society facilitated group meetings with the cohort to develop their SAPs, including tool priorities, milestones, and long-term goals.
- A sample Sustainability Action Plan, developed under the SUSTAIN project in conjunction with Code for Science & Society, can be found in Appendix C.

Deprecation Strategies and Version Control

Deprecation strategies and version control are essential to FLOSS sustainability as they ensure smooth transitions between updates, maintaining system integrity and user trust. These practices also facilitate clear communication about changes and future plans, which is key to managing community expectations and fostering a stable development environment.

Suggested Approaches

- **Clear Communication**: Announce deprecation plans well in advance. Provide detailed explanations and timelines to prepare users for changes.
- **Versioning Protocol**: Adopt a consistent versioning system (like Semantic Versioning) to signal changes, especially breaking changes, to users.
- **Upgrade Path**: Offer a clear and manageable upgrade path for users. Provide detailed documentation and support to ease the transition.
- **Archiving Old Versions**: Ensure that older versions remain accessible for users unable to upgrade immediately. This includes maintaining documentation for these versions.

Resources

- Open Source Series: Version Management: An article that examines software versioning, especially semantic versioning (SemVer), and its importance in open-source software, including managing breaking changes.
- What Organizations Need to Know When Deprecating APIs: A blog post that outlines best practices for API deprecation, focusing on clear communication, transition periods, versioning, and user-impact minimization.

Backward Compatibility Considerations

Prioritizing backward compatibility is key. By ensuring that new updates seamlessly integrate with existing functionalities, this strategy reduces user disruption. This approach not only retains a dedicated user base but also fosters continuous community engagement, both of which are essential for the enduring growth and health of the project.

Suggested Approaches

- **Compatibility Goals**: Define your project's backward compatibility goals. Decide how many previous versions you intend to support.
- **Testing**: Implement rigorous testing for new releases to ensure they do not break backward compatibility. Automated testing can be particularly effective here.
- **Documentation**: Clearly document any backward-incompatible changes. Provide examples and guidance on how to adapt to these changes.
- **Gradual Transition**: Where possible, make backward-incompatible changes gradually. Introduce warnings in earlier versions before fully deprecating features.

Resources

- Ensuring backward compatibility in your applications: An article that discusses strategies for backward compatibility in software, emphasizing functionality maintenance and methods like coding, feature flags, and versioning.
- How can you ensure backward compatibility in developing new software?: This LinkedIn collaborative article provides a step-by-step guide on ensuring backward compatibility in software development, focusing on key strategies for consistent functionality.

Section 3: Governance and Leadership

Good project governance and leadership are crucial for ensuring project stability and sustainability, far beyond mere bureaucratic formalities. This involves clearly defining roles and responsibilities, planning for smooth transitions, and establishing effective decision-making processes. Without these, contributor turnover will increase, and turnover has led to the failure of several FLOSS projects. Providing several roles and diverse opportunities for contribution can help prevent burnout and incentivize further involvement with your project. Additionally, having mechanisms in place for constructive conflict resolution is vital in maintaining a healthy community, underlining the importance of strong governance and leadership in any project's success.

Suggested Approaches

Defining Roles and Responsibilities

Establishing clear roles and responsibilities is akin to creating an orchestra where every member knows their part. It is essential for maintaining order and accountability within your FLOSS community.

- **Maintainers**: These are the architects of your project's vision. Define the roles and responsibilities of maintainers, who oversee the codebase and contributions. They guide the project's direction, ensure quality control, and lead discussions on critical decisions.
- **Contributors**: Contributors form the lifeblood of your project. Outline the expectations and responsibilities of contributors and how they can get involved. Encourage them to provide code, documentation, and bug fixes. Make it clear how they can submit contributions and collaborate with maintainers. Successful projects will also provide contributors with a ladder for growth and space to take on more responsibility as they get more involved.
- **Community Managers**: If applicable, describe the roles and responsibilities of community managers who engage with users and contributors. These individuals often serve as bridges between the community and project leadership. They facilitate communication, organize events, and ensure community health. A community manager does not need to be a trained professional; engineers and other contributors with a passion for the role can develop the skills and knowledge necessary to succeed.

- **Other Roles and Responsibilities**: As demonstrated in the Kubernetes governance structure, tool teams are not limited to the traditional roles outlined above. If your team has a need, build a new role to address it, and if it does not accomplish its goal, terminate it.
    - **Non-Code Contributor Pathways**: Create roles for individuals passionate about your tools mission without the technical skills to serve as contributors. This will allow maintainers and contributors to share the non-coding tasks needed to run a tool and focus more on their specific functions.
    - **Emeritus Positions**: Building a community that allows individuals to exit the project positively by recognizing their contributions with an emeritus title will maintain a bond with those who leave and a more thriving community.
    - **Mentors**: Mentorship is a necessary element of FLOSS sustainability, on which more information can be found in Section 6.

Transition Planning for Key Members

In the fast-paced world of FLOSS, change is inevitable. Preparation for the departure of key members is essential to maintain project continuity.

- **Knowledge Transfer**: Document important project knowledge and processes to ensure a smooth transition. This includes technical documentation, decision logs, and guidelines.

- **Succession Planning**: Identify potential successors for key roles within the project. Encourage contributors to step up and take on leadership positions when the need arises. Nurturing future leaders ensures project resilience.

Decision-Making Processes and Conflict Resolution

Clearly defined decision-making processes and conflict-resolution mechanisms are the gears that keep your FLOSS community running smoothly.

- **Decision-Making**: Document how decisions are made within the project, whether through consensus, voting, or other methods. Transparency is key to community trust. When contributors understand how decisions are reached, they are more likely to participate actively.

- **Conflict Resolution**: Outline how conflicts within the community or project are resolved in a fair and constructive manner. Conflicts can arise from differences in opinion, technical disagreements, or personal issues. A structured resolution process ensures disputes are managed professionally.

Creating (and Enforcing) a Community Code of Conduct

Establishing a clear Code of Conduct is crucial to the successful governance of open source communities. As Coraline Ada Ehmke, creator of the Contributor Covenant, told Mozilla in 2019: "communities of maintainers and contributors [need] a social contract to express and enforce community values of improving diversity and being welcoming to people of all kinds, especially those who are traditionally underrepresented in tech." Codes of Conduct set the standards for engagement, creating a safe, supportive, and welcoming environment for volunteers to spend their limited time contributing to your project.

Creating a Code of Conduct

Codes of Conduct establish expectations and allow for proactive response to potential friction within the community. By communicating clearly defined expectations, you can limit contributor dropout through the maintenance of a happy and healthy community. Explore resources to help you create a Code of Conduct:

- Your Code of Conduct: A Guide that provides critical information about creating, adopting, and enforcing a Code of Conduct within your community.
- How to Create a Code of Conduct for an Open Source Project: An article that breaks down the key steps required for building a robust and impactful Code of Conduct.
- Adding a code of conduct to your project: An article that explains the importance of adopting a Code of Conduct and provides information on adding a Code of Conduct via template or manually.
- Code of Conduct Toolkit for Open Source Communities: This page includes key information about garnering community support for your Code of Conduct and includes some best practices.
- Code of Conduct Examples:
  - Contributor Covenant - A Code of Conduct for Open Source Communities
  - Django Code of Conduct
  - Mozilla Community Participation Guidelines
  - GitLab Community Code of Conduct

Enforcing a Code of Conduct

Congratulations, you've created a Code of Conduct for your community! Proper enforcement of your Code of Conduct signals to the community that you take the standards defined in the guidelines seriously and encourages adherence to the underlying values of your community.

It's also important to establish *how* the Code of Conduct will be enforced so that community members understand the mechanism for reviewing, processing, and resolving reported complaints. Community Managers and Maintainers (defined in the *Defining Roles and Responsibilities* sub-section above) should understand how their roles and responsibilities function within the enforcement of your Code of Conduct. The creation of a Code of Conduct Committee is a great option to lessen the burden of enforcement and ensure that any reported violation is handled in a timely manner.

- Enforcing your code of conduct (Section 4 - Your Code of Conduct): This section (of a larger guide) provides information about how to enforce your Code of Conduct such as how to investigate reported violations and offers examples of appropriate actions to take in response to a violation within the community.
- About community management and moderation: On this page, you will find information about how to moderate your community on GitHub.
- Managing reported content in your organization's repository: On this page, you will find information about how to manage reported violations through GitHub.

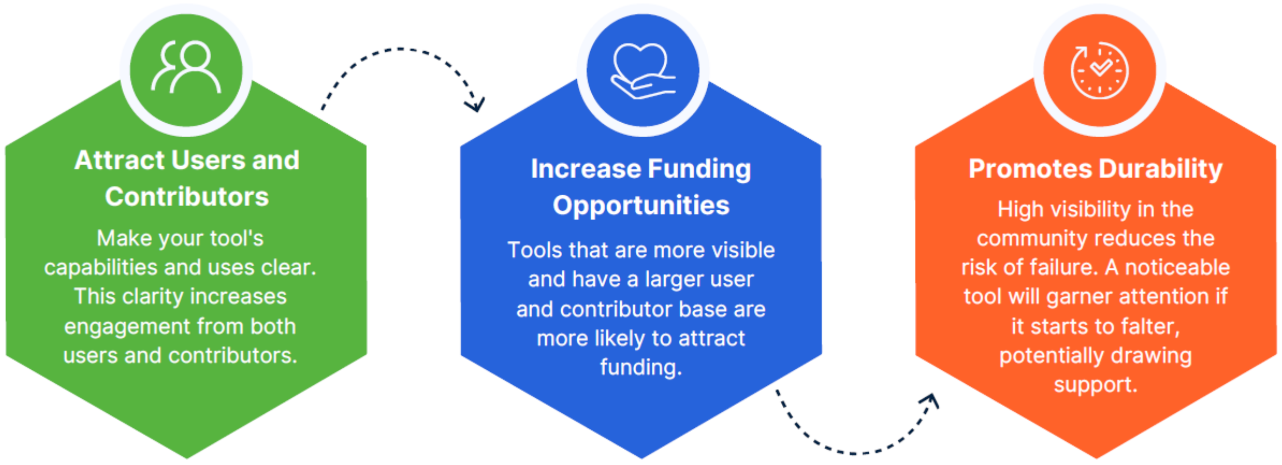
- Example Enforcement Manual and Reporting Guide: Django's enforcement manual and reporting guide are great examples to learn from for your own tool and community.

Section 4: Marketing and Visibility

For sustainable success, open source teams must prioritize marketing and brand visibility. Despite challenges such as limited time, resources, or marketing expertise, understanding the value of these efforts is key to tool longevity. As the open source market rapidly expands, it becomes increasingly challenging to ensure that your tool stands out. Improved visibility not only attracts more users, contributors, and potential donors, but also facilitates meaningful growth and opportunities for funding.

Why Does Marketing and Visibility Matter?

In the context of the FLOSS ecosystem, capturing the attention and resources of users, contributors, and donors is crucial for the success and sustainability of a tool. Here is a breakdown of its key benefits:


Attract Users and Contributors
Make your tool's capabilities and uses clear. This clarity increases engagement from both users and contributors.
Increase Funding Opportunities
Tools that are more visible and have a larger user and contributor base are more likely to attract funding.
Promotes Durability
High visibility in the community reduces the risk of failure. A noticeable tool will garner attention if it starts to falter, potentially drawing support.

Building and sustaining a successful FLOSS tool requires deliberate team effort to create and implement a robust marking and visibility strategy.

Suggested Approaches

Online Presence and Content

Enhancing your tool's online presence with distinct messaging and branding greatly increases its visibility, drawing in users, contributors, and donors. Consider these strategies to effectively market and enhance your tool's visibility:

- **Messaging and Branding**: Create messaging and branding that helps distinguish why users, contributors, and donors should invest their limited time and resources into your tool rather than your competitors'.

- **Website and Search Engine Optimization**: Use your brand identity to create clear messaging that highlights what makes your tool unique. Your tool should not only be easily recognizable, but it should also be easy to find online.

- **Social Media**: Grow your tool's community by promoting key pieces of information on social media platforms. Create a narrative around your tool that draws individuals to your website through your brand identity and clear messaging.

- **Content Marketing**: Target technical and non-technical audiences with your marketing strategy by highlighting a mix of case studies, technical updates, testimonials, and documentation. A key part of your tool's community may not relate to technical content, so it is crucial to use a multitude of methods for sharing what makes your tool unique.

Community Building

Engaging with the community is a critical aspect of sustaining FLOSS projects and should be a central feature of any marketing and visibility strategy. An active community helps ensure that its members remain involved, passionate, and committed to the success of the project and the wider FLOSS ecosystem. Find more information and in-depth suggested approaches for community building in *Section 6: Community Building and Engagement*.

- **Engagement and Support**: Use your marketing and visibility strategy to create pathways for inclusive and diverse community engagement and support.

- **Transparency and Open Development**: Regularly sharing updates on the project's progress, upcoming features, and decision-making keeps the community informed and engaged. It is essential to give users and the community insight into your tool's development. Prioritize this in your marketing and visibility strategy, as the sustainability of an open source project depends on community involvement.

- **User Empowerment**: Involve users in your marketing and visibility efforts. Engage the community by conducting polls on preferred documentation types, requesting testimonials, and inviting them to participate in campaigns that highlight the tool's unique value from their perspective. A successful community is not just about attracting contributors; it is about fostering a space where they feel valued and motivated to contribute.

Networking and Partnerships

Community presence is crucial for FLOSS sustainability. Embedding your tool within the FLOSS community through networking and partnerships opens doors for other key aspects of sustainability such as financial sustainability. The more embedded your tool is within the open source community, the more durable your tool will be in the long run.

- **Collaborations**: Connect with other open source projects through participation in developer forums. Contributing to other projects and actively participating in the open source community will boost your tool's reputation and allow for greater visibility of your work. Explore *Section 6: Community Building and Engagement* for more information about developer forums.

- **Conference Speaking and Participation**: Increase your tool's visibility to potential users, contributors, and donors by actively participating in conferences and events– both in-person and online. Use your tool's logo and branding at these gatherings to enhance recognition and spark discussions about your tool's role in the broader FLOSS ecosystem.
- **Showcasing Case Studies**: Highlight success stories of users who have benefitted from your tool. If necessary, keep these users anonymous and describe their use-cases generally, especially if there are security concerns. These case studies demonstrate to potential users, contributors, and donors the relevance and effectiveness of your tool in serving important communities.
- **Funding and Sponsorship**: Utilizing networks and partnerships through marketing and visibility can lead to funding opportunities. Donors want to provide funds to tools that are visible within the community. The more visible your tool is, the more likely it is to receive funding and/or sponsorship opportunities. Visit *Section 5: Financial Sustainability* for more information about accessing funding opportunities.

Resources

Explore a curated collection of resources that can help you build a marketing plan and grow your visibility.

- How To Market an Open-Source Project: A Linux Foundation blog post on how (and why) to market your open source project.
- Why Open-Source Needs Marketing (Even Though Developers Hate It): An article in Forbes with 5 key lessons learned about marketing open source projects.
- Harnessing the Power of Content: Open-Source Software Promotion Strategies: A blog post from ClearVoice that breaks down the crucial first steps to developing a successful marking strategy for open source tools.
- Finding Users for Your Project: A guide within GitHub's Open Source Guides that explains how to use a marketing and visibility strategy to find new users for your tool.
- Marketing Open-Source Projects: A TODO guide from The Linux Foundation that addresses the challenges of marketing open source projects and provides helpful information about how to address those challenges in a successful marketing strategy.
- Marketing for maintainers: Promote your project to users and contributors: A GitHub panel interview where experts answer questions from the open source community about how to promote open source tools to users and contributors.

Section 5: Financial Sustainability

The long-term success of a FLOSS tool hinges on its economic sustainability. Although these tools are typically free for users, they require funds for development, maintenance, and distribution. This includes costs like developer labor, server space, and other resources. FLOSS projects often struggle with securing maintenance funding, affording market rates for developers, and managing general business development. Balancing financial stability is challenging, as traditional for-profit funding models clash with the values of the open source community. Therefore, open source tools must explore alternative financial models, which presents its own set of challenges.

Fundraising Strategies

Challenges often arise when larger opportunities require substantial funds due to their extensive requirements. Addressing this involves enhancing skills in financial management, building personal connections with donors, and diversifying funding sources. Additionally, it can be beneficial to provide several ways to contribute.

Suggested Approaches

Having a straightforward and well-defined fundraising strategy is crucial. Building personal relationships with donors, offering diverse methods for donations, and securing varied funding sources are key to ensuring your tool's financial sustainability.

Cultivating Relationships with Donors

Building personal relationships with donors can make your tool more noticeable and appealing in competitive funding environments. Additionally, these relationships foster trust and understanding between your tool's team and the donors, enhancing collaboration and support.

- **Get to know the donor**: Investigate the organization's history, including past grants, staff, board members, and consult with former grant recipients. This research helps you understand the donor's objectives and how your tool aligns with their mission.

- **Develop key materials**: Develop a short "elevator pitch," a detailed pitch deck, and a one-page summary about your tool. These materials provide potential donors with quick, accessible information. Having these ready before establishing relationships is crucial, as donors often request such information to better understand your tool.

- **Utilize your community for donor introductions**: Connect with donors through mutual connections or introduce yourself at professional events such as conferences or workshops.

Diversifying Your Fundraising Portfolio

Challenges exist in securing stable funding through non-revenue sources, as grants typically offer short-term support and crowdfunding can be inconsistent. However, employing multiple funding strategies simultaneously can contribute to a more sustainable long-term financial plan.

- Assess the effectiveness of existing funding streams, analyzing their stability and long-term viability.

- Identify areas within current funding sources that can be optimized or expanded for better financial support.

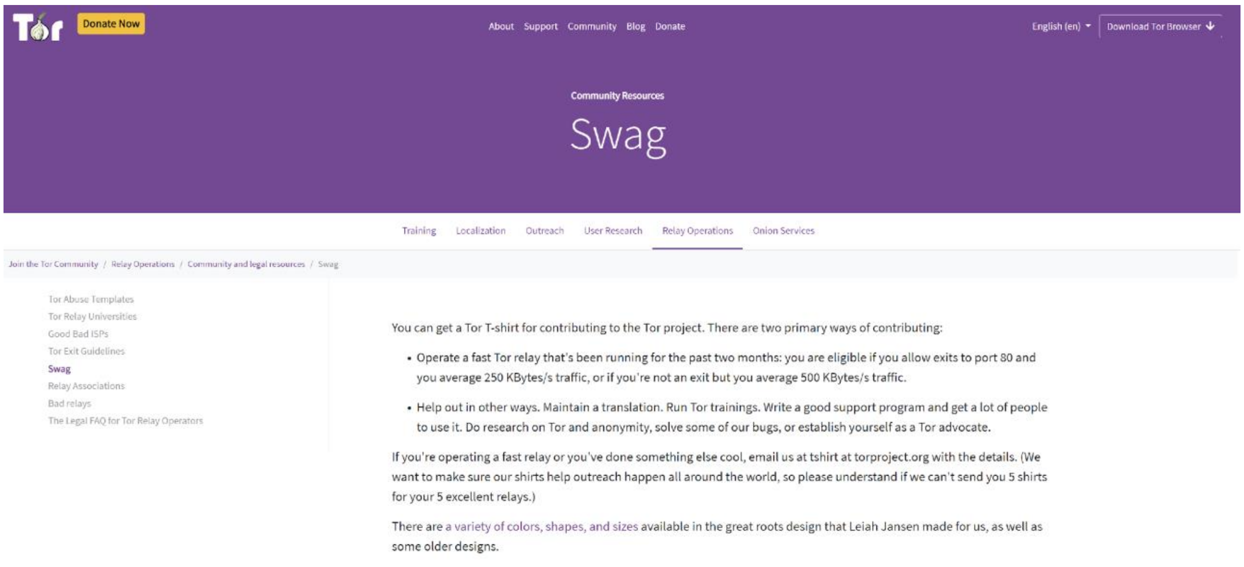
- Explore potential untapped funding avenues that align with your project's goals and values.

Providing Multiple Pathways for Contribution

Offering a variety of ways for donors to contribute is key to maximizing their engagement and support. By catering to the diverse preferences and capacities of donors, you can enhance their involvement and investment in your project. Here are three ways to provide multiple pathways for contribution:

- Offer different donation levels and options, including one-time gifts, recurring donations, and tiered support levels, to accommodate different financial capacities.

- Create opportunities for non-monetary contributions, such as volunteering time, expertise, or resources, appealing to those who prefer to offer support beyond financial means.


Tor
Donate Now
About Support Community Blog Donate
English (en)
Download Tor Browser
Training Localization Outreach User Research Relay Operations Onion Services
Join the Tor Community / Relay Operations / Community and legal resources / Swag
Tor Abuse Templates
Tor Relay Universities
Good Bad ISPs
Tor Exit Guidelines
Swag
Relay Associations
Bad relays
The Legal FAQ for Tor Relay Operators

Community Resources

Swag

You can get a Tor T-shirt for contributing to the Tor project. There are two primary ways of contributing:

- Operate a fast Tor relay that's been running for the past two months: you are eligible if you allow exits to port 80 and you average 250 KBytes/s traffic, or if you're not an exit but you average 500 KBytes/s traffic.
- Help out in other ways. Maintain a translation. Run Tor trainings. Write a good support program and get a lot of people to use it. Do research on Tor and anonymity, solve some of our bugs, or establish yourself as a Tor advocate.

If you're operating a fast relay or you've done something else cool, email us at tshirt at torproject.org with the details. (We want to make sure our shirts help outreach happen all around the world, so please understand if we can't send you 5 shirts for your 5 excellent relays.)

There are a variety of colors, shapes, and sizes available in the great roots design that Leiah Jansen made for us, as well as some older designs.

Figure 1: Tor incentives non-monetary contributions by offering merchandise in exchange for participation as opposed to sale, allowing for multiple pathways of support.

- Integrate innovative funding methods, like gift-matching programs or crowdfunding campaigns, to engage a broader donor base and maximize the impact of each contribution.

Resources

- Connecting with Funders: A resource from Code for Science & Society on how to connect with funders. The resource provides information on how to ask funders for assistance by preparing a concise pitch and deciphering which funder is right for you.

- Fundraising Guide for Nonprofits: A guide from the Nonprofit Finance Fund that aggregates a series of resources on fundraising for nonprofit organizations. The included resources provide helpful information on how to build a non-revenue fundraising strategy that supports your project's financial goals.

- How Do I Ask for Funding?: A resource from Code for Science & Society on how to ask for funding. The resource outlines key considerations to explore prior to requesting funding and provides information on how to make a specific ask.

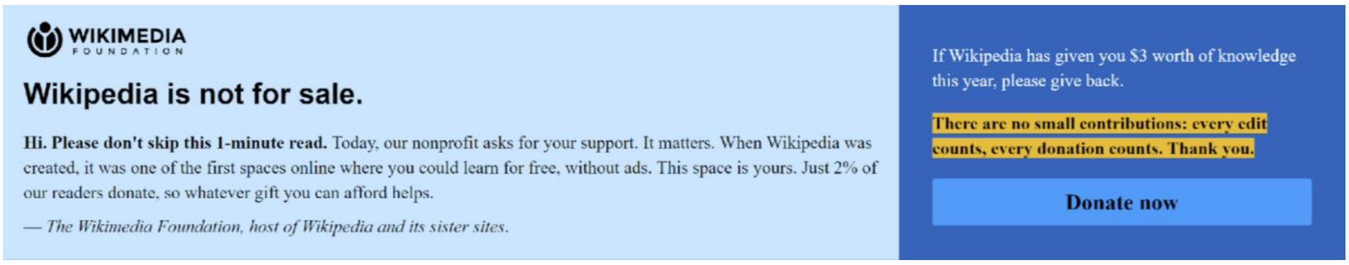
- Financial Sustainability: A resource from Code for Science & Society that provides an overview of and general basics for open source financial sustainability.

Potential Funding Sources

You can secure funding for your FLOSS project through various avenues. Consider applying for grants and fellowships designed for open source initiatives, seeking corporate sponsorships from businesses that value your project, and accepting user donations via platforms like Patreon or Open Collective. Additionally, you can generate funds by offering professional support, selling branded merchandise, and forming partnerships with educational institutions for projects with academic relevance.

Suggested Approaches

- **Philanthropic Funds**: Explore different funding opportunities available through private foundations and public charities.
- **Public Funds**: Look into funding from federal or government sources, which could include core program support or specific solicitations from government agencies.
- **Peer-to-Peer**: Establish clear, specific goals to guide your campaign and use incentives like merchandise, event access, or donation matching to attract and engage new supporters, enhancing the success of your fundraising efforts.
- **Corporate Opportunities**: Tap into the expanding realm of corporate philanthropy, leveraging the growth in businesses looking to support social causes and projects, which can provide substantial funding and partnership opportunities for your initiatives.


WIKIMEDIA FOUNDATION

Wikipedia is not for sale.

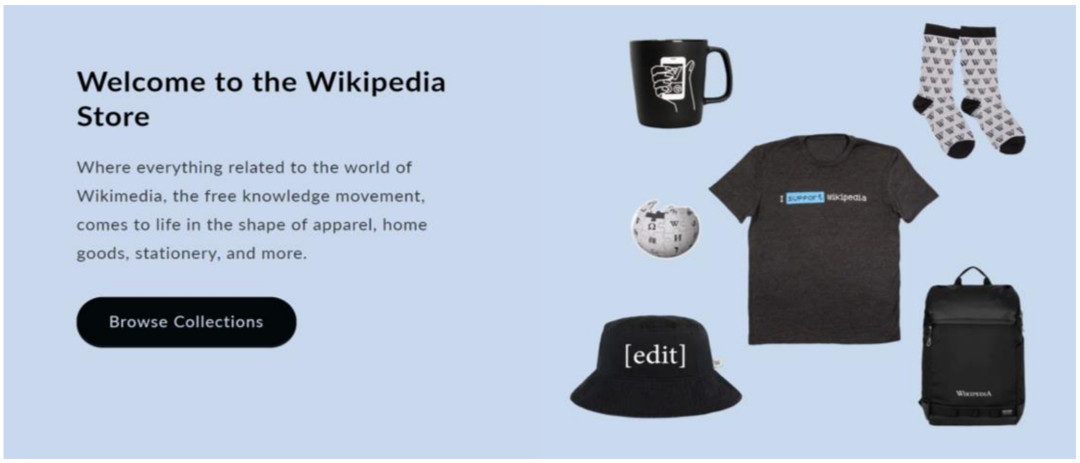
Hi. Please don't skip this 1-minute read. Today, our nonprofit asks for your support. It matters. When Wikipedia was created, it was one of the first spaces online where you could learn for free, without ads. This space is yours. Just 2% of our readers donate, so whatever gift you can afford helps.

— *The Wikimedia Foundation, host of Wikipedia and its sister sites.*

If Wikipedia has given you $3 worth of knowledge this year, please give back.

There are no small contributions: every edit counts, every donation counts. Thank you.

Figure 2: Wikipedia utilizes brand values to fuel peer-to-peer fundraising campaign advertisements.


Welcome to the Wikipedia Store

Where everything related to the world of Wikimedia, the free knowledge movement, comes to life in the shape of apparel, home goods, stationery, and more.

Figure 3: Wikipedia incorporates brand values into their merchandise-based fundraising efforts.

Resources

- Nonprofit Grant Writing: How to Secure Grants for Your Cause: A blog post that compiles a list of effective grant writing tips and provides additional resources for the grant writing process.
- What are funders looking for?: A resource from Code for Science & Society to help you understand how federal funders and private philanthropy organizations make funding decisions. This resource provides key insights into what different types of donors are looking for in funding recipients.
- Funding Resources: A list of resources from Code for Science & Society on finding funding, creating budgets, how to manage a budget, and working with volunteers– each of which are crucial to your tool's financial sustainability.
- A Guide to Funding: A resource from Code for Science & Society that provides a list of fundraising terms that are important to become familiar with as you seek funding opportunities within the nonprofit community.
- Why Do People Give to Charity? The Psychology of Storytelling and Its Impact on Nonprofit Fundraising: A blog post that details how to utilize storytelling to encourage financial contributions from funders and through crowdfunding campaigns.

Financial Management

Funding for FLOSS tools often focuses mostly on software development, leaving limited resources for other critical aspects of long-term sustainability. Finding time to focus on financial management can be challenging with numerous immediate concerns. However, effectively managing finances is crucial. Not only does it ensure the efficient use of funds, but it also makes the tool more attractive to donors by demonstrating responsible and strategic financial stewardship.

Suggested Approaches

- **Revenue Models**: Developing a clear and detailed revenue model is essential for effectively handling various funding sources. This approach allows for strategic planning and allocation of resources across different funding streams, ensuring financial stability and sustainability for your project.
- **Budgeting**: Efficient budgeting processes are crucial, as many donors require financial or organizational budget statements during funding applications. Implementing financial tracking tools, budget templates, and other administrative tools can strengthen your financial reporting and administration, illustrating to donors that their investments will be effectively utilized and aligned with your budget structure.
- **Demonstrate Impact:** Creating an annual report that transparently details your use of funds over the past year is crucial. This not only showcases visible outcomes and progress but also reinforces trust and accountability with donors and stakeholders.

- **Fiscal Sponsors**: Consider partnering with a fiscal sponsor, an organization that manages donations for you, typically for a portion of the donation. Examples include Software Freedom Conservancy, Apache Foundation, Eclipse Foundation, Linux Foundation, and Open Collective, all of which act as fiscal sponsors for open source projects.

- **Computerized Accounting**: Consider the use of computerized accounting systems to simplify financial management; many of these systems are available as free and open source options.

- **Financial Policies**: Ensure efficient use of funds by developing key policies. This includes regular financial reporting, vendor disbursement procedures, travel and procurement policies, payroll management, and safeguards against fraud and mismanagement. Regular independent financial audits should also be conducted.

Resources

- Financial Management: A comprehensive list of basic financial policies, financial literacy resources, and pointers about financial management for nonprofit organizations from the National Council of Nonprofits.

- Nonprofit Financial Management Webinars: A series of financial management webinars from the Nonprofit Finance Fund.

- 5 Crucial Topics to Address in Any Nonprofit Organization's Financial Policy: A guide that provide information on key components that your organization's financial policies should address.

Advice for Funders: How to Make an Impact

Although the majority of the guide is geared towards FLOSS tool teams themselves, the financial sustainability of these tools often hinges on opportunities provided to them by funders. Thus, it is crucial that funders have a clear understanding of the challenges these tools face and how funders can pave the way for sustainability. The information provided in this section can be used to help inform conversations with donors and hopefully shape broader conceptualizations of FLOSS funding.

Suggested Approaches

Throughout the development of this guide, FLOSS tools working with Internews during the SUSTAIN project identified key focal points important for funders to consider when supporting this community.

- **Maintenance Funding**: While new features are exciting and important, the nature of the open source ecosystem often leaves maintenance as an afterthought. More than anything, FLOSS tools are in dire need of maintenance funding. Investment in the basic operations and routine maintenance of a tool contributes to its sustainability by ensuring that the tool's foundation remains strong and team members have the necessary resources to "keep the lights on."

- **Long-Term Funding Relationships**: Stable funding is one of the most important aspects of sustainability, allowing tools to address basic operations while having the freedom to embark on more interesting development projects. As most funders offer grants between 6 and 18 months when engaging with FLOSS tools, many teams find that they lack consistent streams of funding. By offering multi-year grant opportunities that include consistent baseline funding for maintenance, funders can support FLOSS tools by serving as a source of stability, allowing teams to address basic operations and new features at the same time.
- **Scoping**: If maintenance and/or long-term funding is not possible, it is useful for funders to ensure that funding is paired with a realistic and clearly defined scope that allows tools to address specific needs. However, without maintenance support, the full potential of new features cannot be realized so it is crucial to include such support in short-term funding opportunities as well. Funders should coordinate with the tool developers to ensure the scope of the grant appropriately balances resources.
- **Minimize Funding Gaps**: The current structure of funding cycles leaves many tools short of cash while going through the proposal process after one grant ends. When there are gaps in funding, tool teams report that they struggle with talent retention, lags in code maintenance, limited forward momentum, and general tool stability. Funders can support FLOSS sustainability by limiting the time between the end of one grant cycle and the start of funding for another as well as finding ways to provide support during the application process when possible.
- **Investment in Longevity**: Similarly to the balance between funding maintenance alongside the development of new features, tool teams suggest that funders should not allow newness to detract from investment in softwares that have proven their utility. Tools that currently exist within the space and have proven themselves as useful still require funding to operate and serve the communities that depend on its functionality. By investing in the longevity of established software, funders can ensure that these tools continue to have necessary maintenance resources.

Section 6: Community Building and Engagement

A strong, engaged community is vital for the sustainability of an open source software project. It not only aids in development but also creates a sense of ownership and support. Effective communication, community management, and a welcoming environment are crucial for building and sustaining this community. Recognizing contributors and valuing diverse opinions encourage active participation, which is essential for innovation, workload sharing, and ensuring the project's longevity. This ongoing effort benefits the project, its users, and the open source movement.

Building an Inclusive and Diverse Community

A thriving community is inclusive and diverse, welcoming contributors from various backgrounds. It's not just about attracting contributors; it's about creating an environment where they feel valued and empowered to contribute.

Suggested Approaches

- **Diversity Outreach**: Actively reach out to underrepresented groups and encourage their participation. Consider partnerships with organizations that promote diversity in tech.
- **Code of Conduct**: Implement a clear code of conduct that sets expectations for respectful behavior. It should make clear that your community is committed to being a safe and welcoming space for all.
- **Mentorship Programs**: Establish mentorship programs to help newcomers get involved and grow within the community. Seasoned contributors can guide and support new members.
- **Inclusive Language**: Promote the use of inclusive language in project communications. Simple changes in language can make a big difference in fostering inclusivity.

Access more community-building resources HERE.

Localization

Localization of open-source software extends beyond simple translation, involving the adaptation of software to different languages and cultural contexts to make it accessible and relevant to a global audience. This process is crucial for open-source projects, as it not only broadens the user base but also fosters a diverse and inclusive community of contributors.

Successful localization efforts, as seen in projects like Mozilla Firefox and LibreOffice, play a pivotal role in enhancing user engagement and ensuring the sustainability of open-source tools by making them accessible globally. It acts as a crucial bridge in open-source development, connecting diverse populations by overcoming linguistic barriers, thereby fostering participation and collaboration. This inclusive strategy not only infuses projects with varied perspectives and skills but also cements the core open-source principles of accessibility and community-driven development.

Suggested Approaches

Adhering to these best practices can significantly improve the localization process in open-source projects, making software more accessible and user-friendly for a global audience.

- **Engage the Community Early:** Involving the community at the outset of localization efforts ensures a broader base of support and input, making the process more inclusive and efficient. Early engagement helps in identifying potential cultural sensitivities and linguistic challenges.

- **Use Collaborative Platforms:** Leveraging platforms like Transifex, Weblate, or Crowdin can streamline the localization process. These platforms facilitate collaboration, provide tools for consistency (like glossaries and translation memories), and make it easier to manage contributions from volunteers.

- **Establish Clear Guidelines:** Creating detailed guidelines for translators is crucial. These should cover linguistic styles, technical terms, and cultural considerations. Clear guidelines help maintain quality and consistency across translations.

- **Automate Workflow Where Possible:** Automation tools within TMS platforms can significantly reduce manual work. Features like automatic notifications for updates, integration with version control systems, and automated quality checks can enhance efficiency.

- **Foster a Culture of Appreciation:** Recognizing the efforts of volunteer translators is vital. Regular shout-outs, acknowledgement in project updates, or even small rewards can go a long way in keeping the community motivated.

- **Ensure Quality Control:** Implement a review process involving experienced translators or native speakers to ensure the accuracy and appropriateness of translations. Peer reviews can also serve as a learning tool for new translators.

- **Plan for Continuous Updates:** Localization is an ongoing process. Plan for regular updates and revisions to keep translations aligned with the latest version of the software and to incorporate feedback from users.

Resources

- Exploring the Impact of localization on Open Source Sustainability: Blog Post by the Internews SUSTAIN project highlighting that the process of localization is crucial for open source projects as it not only broadens the user base but also fosters a diverse and inclusive community of contributors.
- Mozilla Localization (L10n) Guides: Provides comprehensive resources on localization best practices, community engagement, and the use of technology in the localization process.
- The Globalization and Localization Association (GALA): Offers resources and professional insights on translation, localization, and globalization, useful for understanding industry standards and best practices.
- Community Forums and Documentation: Online forums, wikis, and documentation specifically dedicated to localization can provide valuable guidance for new translators, share best practices, and discuss common challenges and solutions.

Engaging with the Community

Engaging with the community is a critical aspect of sustaining FLOSS projects. Once community networks are in place, it's important to keep them dynamic and relevant. An active community helps ensure that its members stay involved, passionate, and committed to the success of the project and the wider FLOSS ecosystem.

Suggested Approaches

Regular and clear communication is key. This can be through newsletters, social media, forums, or regular get togethers. Updates about the project's progress, upcoming features, and decision-making processes help keep the community informed and involved.

Hosting Events and Webinars

Engaging with your community through events and webinars can significantly boost participation and strengthen relationships. It is an opportunity to connect with your community on a personal level.

- **Hackathons and Contribution Sprints**: Organize events where contributors can work together on specific tasks or features. These events are not only productive but also foster a sense of camaraderie.

- **Webinars and Workshops**: Conduct webinars and workshops to share knowledge and best practices. This is an excellent way to provide in-depth information and answer questions in real-time.

- **Community Gatherings**: Arrange physical or virtual meet-ups and conferences to connect contributors and users. These gatherings provide an excellent forum for networking and collaboration.

When attending events organized by other organizations and teams, take the time to reflect on what you did and did not like about it and most importantly whether it worked for the target audience.

Developer Forums

Engaging with developer forums can be a great way to get answers to your technical questions, share knowledge, and connect with the community. Here are some examples:

- **Stack Overflow** is a popular Q&A site for developers to ask and answer technical queries, ideal for code troubleshooting.

- **GitHub Discussions** hosts discussion boards for open source projects, facilitating project-specific conversations and support.

- **Dev Community** is a network of over a million developers for sharing, collaborating, and accessing resources like events, jobs, mentorship, and more.

- **GitLab** is a web-based platform that offers source code repository management.

- **Bitbucket** is a source code repository hosting service, preferred for its strong version control and code management features.

Maintaining strong communication with end-users is key. This can be effectively achieved through user forums and chat channels. These platforms not only facilitate discussions, troubleshooting, and knowledge sharing but also help users support each other and foster a sense of community belonging.

- **Privacy-Respecting Analytics**: Employ analytics tools that prioritize user privacy to gather important usage data. Transparency in how data is collected, adherence to privacy laws, and providing an opt-out option are essential. This data should be used to improve the user experience and inform development decisions, ensuring that user needs are at the forefront.
- **Feedback Channels**: It's important to provide easily accessible pathways for users to report bugs, request features, and offer suggestions. Responsiveness to this feedback is crucial, as it shows users that their input is valued and taken seriously.
- **User Surveys**: Conducting regular surveys is a great way to gather user feedback and insights. These surveys can highlight areas of improvement, desired features, and overall user satisfaction, guiding future developments.

Privacy-Respecting Analytics

FLOSS tool developers have several options when it comes to collecting usage data, an important decision in planning for the tools' sustainability. Using traditional analytics tools can lead to collecting more information than necessary, and they often do not take into account the privacy of the users; not collecting data to respect privacy, however, reduces the ability to understand the effectiveness of the tool and its user experience. In addition, some funders may rely on usage data as a part of their decision making process, requiring quantitative metrics to determine the success of their support and shape future funding.

To gather information while respecting users and maintaining community trust, tool teams can employ analytics tools that prioritize user privacy to gather important usage data. Transparency in how data is collected, adherence to privacy laws, and providing an opt-out option are essential. Privacy-respecting data can then be used to improve the user experience and inform development decisions, ensuring that user needs remain at the forefront.

Clean Insights, developed by SUSTAIN project partner Guardian Project, gives developers a way to plug into a secure, private measurement platform. It is focused on assisting in answering key questions about app usage patterns, and not on enabling invasive surveillance of all user habits. The Clean Insights approach provides programmatic levers to pull to cater to specific use cases and privacy needs. It also provides methods for user interactions that are ultimately empowering instead of alienating. For instance, ensuring an informed consent experience is part of the measurement process. Digital consent is the process of asking and receiving permission for something to happen. A consentful digital experience respects the rights and interest of the people using the technology. Integrating consentful experiences into analytics is a main focus of Clean Insights. The goal is to meet the user at an intentional moment in the process to reduce friction, make the request for measurement context-specific, and reduce overall annoyance. To learn more, tool developers can visit the Clean Insights docs site and schedule a free Toxic Asset Audit with Guardian Project to learn what unnecessary and potentially harmful data they are collecting and learn more about implementing privacy-respecting analytics.

Documentation for New Contributors

Beginners often struggle with documentation, but it's a crucial aspect of open source projects. Well-structured and comprehensive documentation is not just nice to have; it is the cornerstone of attracting and retaining contributors. It is the first resource newcomers turn to when they want to understand your project.

- **Documentation Structure**: Organize documentation into clear sections, making it easy to navigate. A well-structured documentation hub ensures that users and contributors can quickly find the information they need.
- **Tutorials and Guides**: Create step-by-step tutorials and guides to help contributors get started.
- **Examples and Use Cases**: Provide practical examples and use cases to illustrate how the project works. Concrete examples help contributors grasp concepts faster.
- **Documentation Maintenance**: Regularly update and maintain documentation to keep it current. Outdated documentation can lead to frustration and confusion.

Resources

- Starting an Open Source Project: This guide provides tips on writing open source documentation. It covers everything from creating README files to documenting your project's code and processes.
- How to Contribute to Open Source Projects - A Beginner's Guide: This step-by-step guide explains how to contribute to open source, including the documentation process, even for those who are non-coders.
- How to write effective documentation for your open source project: This guide explains how documentation quality can make a difference in whether uses or contributors try your project or pass it by. It covers how to write *about your code* to convince other developers to follow your project.

Access more documentation resources HERE.

Mentorship

A strong mentorship program is vital for cultivating the next generation of FLOSS contributors and leaders, ensuring the long-term sustainability of projects. Mentors play a crucial role in sharing knowledge, providing guidance, and fostering an environment that encourages growth and innovation within the community. According to Forbes, effective mentorship programs can improve employee retention rates by up to 49%, underscoring the importance of mentorship in building a committed and engaged workforce.

Strong mentorship programs offer a triple win for FLOSS projects:

- **Cultivate a skilled contributor pipeline:** Mentorship equips newcomers with the knowledge and skills needed to thrive within the project, fostering a steady stream of valuable contributors.

- **Ensure knowledge transfer:** Experienced mentors can effectively pass on their expertise to the next generation, preserving valuable project knowledge and practices.
- **Promote a culture of continuous learning and innovation:** A strong mentorship program fosters a collaborative environment where learning is encouraged, leading to ongoing innovation within the FLOSS community.

Suggested Approaches

- **Identifying Mentors:** Establish criteria for selecting experienced contributors as mentors based on their technical expertise, communication skills, and passion for nurturing talent. For example, the Linux Foundation Mentorship Program looks for mentors with at least 2 years of active contribution to an open-source project and strong interpersonal skills.
- **Mentor Training:** Provide training to mentors on effective coaching techniques, setting expectations, and creating inclusive mentoring relationships. Mozilla's Mentorship Training covers topics like creating psychological safety, providing constructive feedback, and promoting diversity and inclusion.
- **Structured Programs:** Develop structured mentorship programs with clear goals, timelines, and milestones to ensure focused guidance and measurable progress. Outreachy, a successful FLOSS mentorship program, follows a structured 3-month model with specific coding tasks and deliverables.
- **Matching Process:** Implement a systematic process for matching mentors and mentees based on their skills, interests, and areas of development. The Google Summer of Code program uses a comprehensive application and matching system to pair students with mentoring organizations.

Best Practices

- **Lead by Example:** Mentors should model best practices in coding, collaboration, and community engagement, inspiring mentees to uphold the principles of FLOSS. As stated in the Drupal Mentor Guide, "Lead by example in your interactions, your code contributions, and your commitment to the project."
- **Personalized Support:** Tailor mentorship approaches to the unique needs and learning styles of each mentee, fostering an environment of personalized growth. The Rails Girls Summer of Code program encourages mentors to adapt their guidance based on mentees' backgrounds and goals.
- **Continuous Feedback:** Encourage open communication and provide regular feedback, recognizing achievements and identifying areas for improvement. Effective feedback cycles, as outlined in the Mozilla Mentoring Guide, involve setting expectations, providing timely feedback, and celebrating successes.

- **Knowledge Sharing:** Facilitate opportunities for mentees to share their knowledge and experiences, fostering a culture of mutual learning and collaboration. The Linux Kernel Mentor Program organizes regular "mentor summit" events where mentees present their work and share insights.

Resources

- Open Source Guides - Mentoring: This guide from GitHub provides best practices for mentoring in open source projects, including advice on how to find a mentor, be a good mentor, and create a mentorship program.
- Mozilla Mentoring Resources: A collection of resources from Mozilla on mentoring, including guides, toolkits, and research reports on effective mentorship practices.
- Linux Foundation Mentorship Resources: The Linux Foundation provides a wealth of resources and tools to support mentorship programs within open-source communities, including guidelines, templates, and best practices.
- FOSS Life: The Value of Mentorship in Open Source: This article explores the multifaceted benefits of mentorship in open source projects, from knowledge transfer and community building to personal growth and increased diversity.
- Linux Foundation Mentorship Programs: The Linux Foundation hosts several mentorship programs aimed at fostering the next generation of open-source contributors and leaders, providing valuable learning opportunities.
- Layer5 Community Handbook - Mentorship Programs: The Layer5 Community Handbook includes a dedicated section on mentorship programs, offering insights and guidance on establishing effective mentoring initiatives within open-source communities.
- How to find an open-source mentor (Testimonial): This article provides practical tips and advice from an individual's journey to becoming a mentor in an open-source community.

Levering Volunteer Contributions

In the realm of Free/Libre/Open Source Software (FLOSS), volunteers stand as the cornerstone, propelling projects forward with a diverse array of contributions that extend beyond coding to include documentation, design, user support, and more. As seen in projects like Linux Kernel and Mozilla Firefox, volunteer efforts not only enrich the projects they contribute to but also ensure the sustainability and innovation of the open-source ecosystem.

Suggested Approaches

Through platforms like GitHub and GitLab, volunteers from diverse backgrounds collaborate on projects that spark their interest, contributing to the collective advancement of technology. Maintaining an active volunteering community requires building an inclusive and diverse community and engaging with the community, as described above. Creating a conducive environment for volunteers is essential for the growth and sustainability of open-source projects.

Here are some best practices that can help engage and retain volunteers effectively:

- **Build a Welcoming Community:** Adopt a code of conduct and enforce it consistently to ensure respectful interactions. Welcome new members warmly and provide platforms for them to introduce themselves and their interests. Celebrate diversity and promote inclusivity at every opportunity.

- **Clear Contribution Guidelines:** Maintain detailed contributing guidelines that cover everything from how to submit a bug report to the process for proposing significant changes. Make these guidelines easily accessible to all potential contributors.

- **Mentorship and Support:** Establish mentorship programs that pair newcomers with experienced contributors. This can help new volunteers navigate the project and make meaningful contributions more quickly. Ensure there are clear channels (like dedicated Slack/Discord channels or forums) where volunteers can ask questions and receive support. (For information about mentorship, Internews and Code for Science & Society have published this blog post).

- **Recognition and Rewards:** Implement a system to regularly recognize contributions, such as monthly highlights of outstanding contributions or annual awards. Small tokens of appreciation, like digital badges or swag, can also make volunteers feel valued.

- **Facilitate Ownership and Leadership:** Encourage volunteers to lead initiatives, manage sub-projects, or become maintainers of certain components. Provide the necessary support and recognition for these leadership roles.

- **Regular Communication and Feedback:** Hold regular meetings or video calls to discuss project updates, challenges, and future plans. Use surveys or feedback tools to gather insights from volunteers about their experience and suggestions for improvement.

- **Regular Meetups and Events:** Organize regular meetups, such as monthly virtual calls or annual in-person gatherings, to unite the community. These events can range from informal social gatherings to structured sessions with talks, workshops, and hackathons. Encourage volunteers to propose sessions, share their work, or lead discussions on topics of interest. For larger projects, consider regional or local meetups to accommodate contributors from different geographies.

By implementing these best practices, open-source projects can create a nurturing environment that not only attracts but also retains volunteers. Engaged and supported volunteers are more likely to make sustained contributions, driving the project towards its goals and ensuring its long-term viability.

Resources

- Leveraging Volunteer Contributions for Sustainability: Blog post by the Internews SUSTAIN project emphasizing that volunteers are the cornerstone of FLOSS tools, propelling projects forward with a diverse array of contributions that extend beyond coding to include documentation, design, user support, and more.

- 5 reasons why you should contribute to open source projects & How Contributing to Open Source Projects Can Improve Your Engineering Resume: Blog posts that can help develop ways to convince potential volunteers to assist with your project.

- Volunteer management tools:
  - Stack Overflow: A vast community of developers answering questions and sharing knowledge.
  - Mozilla Developer Network (MDN): Offers detailed documentation and learning resources for web technologies.
  - All Contributors: A tool to recognize all contributions, not just code.

Section 7: Technical Sustainability

In this section we delve into technical challenges such as Licensing and Compliance Issues, Security Concerns, and Code Auditing, highlighting their direct impact on a project's sustainability. A clear understanding and proactive management of these aspects are crucial, as they influence not only the immediate functionality and security of the software but also its ability to adapt, grow, and remain relevant over time. Through explanations, best practices, and real-world examples, this section aims to empower developers with the knowledge and strategies needed to navigate these challenges, contributing to the robustness and endurance of their open source projects.

Licensing and Compliance

Licensing and compliance are very important to the sustainability of FLOSS projects, as they provide clear rules for how the software can be used, modified, and shared. A well-chosen license attracts more contributors and users, encouraging an innovative and reliable community around the software. This not only enhances the software's quality and functionality but also increases its appeal, potentially leading to more funding opportunities and economic stability. Clear licensing terms also build trust among users and developers, ensuring a transparent and reliable environment for collaboration.

Meanwhile, compliance helps to avoid costly and reputation-damaging legal disputes related to intellectual property rights. By observing licensing terms and regulations, FLOSS projects can maintain their legal integrity, protecting themselves from potential legal challenges and ensuring smooth operation. This legal stability is crucial for the project's credibility, attracting more users and contributors, and long-term economic viability. Licensing and compliance are not just boring legal formalities; they are strategic practices that underpin the sustainability of FLOSS projects, ensuring their growth, innovation, and longevity.

Suggested Approaches

To make sure tools are properly licensed and in compliance with terms and conditions, it is important to understand the different licenses out there. Take time to research and compare various licenses, paying attention to their specific constraints and permissions. The Free Software Foundation (FSF) is a valuable resource, providing extensive information on a wide range of open source licenses. Make sure that the chosen license aligns with your project's goals, clearly defining what is and is not allowed. Use online tools to find a license that matches your project's objectives. Establish solid compliance procedures within your team, ensuring everyone observes the licensing terms and related regulations. This includes creating clear documentation on your licensing policy and including a detailed licensing file in your project.

There are several tools and resources available that can help maintain license compliance. These include:

- **License Management Software**: These tools efficiently manage and monitor software licenses and intellectual property, providing usage reports and compliance alerts.
- **Compliance Consultants**: Compliance consultants assist in ensuring license compliance, helping with agreement understanding, system implementation, audits, and provide training.
- **Legal Counsel**: Regulatory and compliance lawyers can advise on licensing agreements, manage non-compliance implications, assist negotiations, and address related legal issues (this should be specific to the country where your tool is registered).

When choosing the right tools for your organization, consider the size of your organization, the complexity of your licensing agreements, and the resources available to you.

Resources

- FOSSA is a license compliance tool for FLOSS projects that scans and audits licenses and provides remediation suggestions and policy enforcement.
- License Finder detects the licenses of FLOSS projects and their dependencies. A range of online resources can also be used for checking on patents and copyright for different types of content.

Security Concerns

Secure coding practices are essential for FLOSS tools, influencing their widespread adoption and supporting their longevity. As trust plays a central role in the FLOSS community, tools that emphasize security and reliability are more likely to attract users, contributors, and vital financial support in the form of donations or sponsorships.

Ensuring the tool's integrity, availability, and confidentiality bolsters a tool's reputation, fostering both community growth and financial sustainability. Given that many FLOSS projects operate on tight budgets, security breaches can severely hinder their ability to thrive. Thus, adopting secure coding practices is not just about technical robustness; it is a strategic move towards the project's continued relevance, growth, and financial viability.

Suggested Approaches

FLOSS tools should create strong security frameworks that, when used effectively, can both address current vulnerabilities and bolster the team's future capacity to secure their systems. Outlined are several recommended practices for achieving this:

Regular Audits and Secure Coding

- Frequent Security Checks: Regularly audit the system to identify and mitigate vulnerabilities, tailoring the audit frequency to the complexity of the system and sensitivity of the data.
- Secure Coding Standards: Follow industry-standard security protocols, including data encryption to protect sensitive information and input validation to prevent common vulnerabilities.

Proactive and Educative Approaches

- Security Response Plans: Develop clear procedures for managing reported vulnerabilities, ensuring prompt validation, impact assessment, and implementation of fixes.
- Security Training: Provide comprehensive training to the team on security best practices and responsible data handling to foster a culture of security awareness.

External Validation

- External Security Auditors: Bring in third-party security experts for unbiased assessments, providing an additional layer of validation and help uncovering potential oversights in your security strategy.

Resources

- Explore the National Vulnerability Database to find current information on security vulnerabilities, including their impacts, solutions, and metrics for a specific vulnerability.

Quality of Code

Quality code is critical to a software's performance, scalability, and ease of maintenance. Poor quality code poses many challenges, including making it more difficult to onboard new contributors and is unattractive to the developer community. This leads to longer development cycles due to the need for more frequent code reviews and bug fixes, ultimately compromising the reliability of the software. A software's reliability is crucial, as it directly influences user satisfaction and trust in the software, impacting the long-term success and viability of the project.

Suggested Approaches

- **Establish Clear Coding Guidelines**: Develop clear coding standards, such as naming and indentation conventions, as well as commenting guidelines, while also advocating for peer reviews and pair programming to internalize best practices.

- **Form a Dedicated Quality Team**: Create a team specifically tasked with maintaining and enhancing code quality.

- **Enhance Skills and Automation**: Invest in training for code quality and design, build automated testing to detect bugs, and adopt tools for consistent peer reviews.

- **Refine Review Processes**: Develop a multi-tiered review strategy for varying code complexities and host workshops to improve review skills and tool usage.

Resources

- Use Crucible for peer code reviews, enabling team members to collaboratively examine and improve each other's code for enhanced quality and performance.

Code Auditing

Effective code audits are crucial for error detection, code quality, and compliance, yet they can be a significant challenge for many teams, due to funding and/or capacity restrictions. Nevertheless, neglecting regular auditing risks errors, inconsistency, and security vulnerabilities, so thinking of it as a key to the tool’s sustainability is vital. It is important to coordinate a robust audit process. Weak processes might miss critical issues, leading to long-term maintenance challenges, reduced code efficiency, and increased vulnerability to malicious attacks.

Suggested Approaches

- **Set Audit Objectives**: Outline the audit's focus, highlighting key concerns and all relevant software, hardware, and related technologies, while referencing past audits to inform a thorough and phased approach.

- **Promote Audit Collaboration**: Encourage a transparent, cooperative environment for audits, inviting participation and input from both internal and external stakeholders for impactful integration of results.

- **Build Audit Capacity**: Strengthen internal capacity by providing trainings on audit methods, adopting automated tools, formalizing review protocols, and seeking broad input for ongoing refinement.

Resources

- CodeSonar: Use CodeSonar to conduct static analysis on your source code or binaries, identifying quality and security flaws.

- CodeScene offers static analysis capabilities for detecting security and quality defects in your code and binaries.

Usability & Accessibility

Usability and accessibility are necessary elements of ensuring a FLOSS tool reaches its intended audience and achieves its purpose. By maintaining a focus on human-rights centered design while developing a FLOSS tool, tool teams will create a product that meets the real needs of its users. Tools that are usable and accessible will attract a broader and larger audience of users and as such, more financial resources. With a broader audience of users, developers will also receive more feedback that will contribute to a positive feedback loop in improving the usability of the tool.

A focus on human-rights centered usability within a tool team is also a practice that will reinforce the open and welcoming culture for diverse contributors which, as discussed above, is critical to FLOSS sustainability.

Suggested Approaches

- **Host virtual or in-person synchronous tool feedback sessions with real-world users:** Gathering the developers and the users in the same room can provide more useful feedback than through asynchronous channels alone. Person-to-person conversations allow for more engagement and back-and-forth discussion that can refine the feedback and build a stronger roadmap, as well as develop humanizing connections that promote sustained relationships in the future.

- **Conduct usability testing with different audiences:** FLOSS tools are used in many contexts, including those that the developers may not have in mind while developing the tool. These audiences can provide valuable insight into the usability of a tool and allow developers to create new features or address the needs of new audiences, promoting more widespread adoption of the tool.

- **Utilize existing trainer networks to reach at-risk communities**: Some users are less likely to engage directly with tool teams due to the risks they face. Digital security trainers are uniquely positioned to bridge high-risk users and tool developers. Their deep understanding of the challenges and needs of users coupled with their familiarity with most FLOSS tools allows them to more easily capture, prioritize, and share usability and accessibility feedback with tool teams. When you cannot access users directly, consider using personas which were built by at-risk communities or trainers working with those communities.

- **Partner with usability experts:** Organizations like Superbloom, OkThanks, and Ura Designs, among others, can help tool teams understand how to build a more human-centered tool.

- **Prioritize accessibility within tool development and maintenance**: Ensure that your tool is compliant with standard web content accessibility guidelines (WCAG 2.1, Level AA). For a more in-depth review of your tool's accessibility, request support from experts at organizations such as Accessibility Lab.

Resources

- USABLE Guidebook: Conduct usability tests using this Internews developed guidebook containing instructions and resources for conducting several different usability test formats.

- "How Do Open Source Software Contributors Perceive and Address Usability? Valued Factors, Practices, and Challenges": This academic article provides an in depth analysis of the importance of usability within open source software.

Case Studies from the SUSTAIN Project

Sustaining OpenArchive: A Case Study on Enhancing Digital Preservation for Human Rights

This case study examines OpenArchive, an organization dedicated to preserving and securely routing mobile media created by human rights defenders and citizen journalists to community-maintained collections. It explores their sustainability strategies, particularly through the SUSTAIN project funded by the U.S. State Department's DRL. Under SUSTAIN, OpenArchive received funding to improve the sustainability of their app, ***Save***. Key findings highlight the importance of financial sustainability, app maintenance, and team well-being.

Introduction

Background

Natalie Cadranel founded OpenArchive as part of her master's program at University of California Berkeley in 2014. She was motivated by the challenges faced by activists and citizen journalists, particularly the lack of secure and reliable methods for sharing media documenting human rights violations. The primary objective was to create a user-friendly tool that incorporated security, authenticity, and verification features, ensuring the longevity and reliability of the media shared.

Importance and Relevance

This case study is crucial for understanding how non-profit organizations can achieve sustainability while maintaining their core mission of supporting human rights.

Purpose and Objectives

The primary goals of the SUSTAIN project include achieving financial stability, ensuring continuous app maintenance, promoting team well-being, and improving the communications strategy. This case study aims to detail the challenges faced, strategies implemented, and the outcomes achieved through this initiative.

Literature Review

Existing research highlights the critical role of secure mobile media in human rights documentation. However, gaps remain in sustainable funding models and long-term maintenance strategies for such tools. OpenArchive addresses these gaps through a comprehensive sustainability plan.

Methodology

This case study utilizes data from an interview with Alex Esenler, the Deputy Director at OpenArchive, and their Sustainability Action Plan for the Save app and OpenArchive organization. Qualitative analysis was conducted to identify key themes and strategies employed by the organization.

Findings

OpenArchive's commitment to sustainability is structured around three main pillars: financial sustainability, app maintenance, and team well-being. These pillars are crucial for ensuring the organization's long-term viability and effectiveness.

1. Financial Sustainability

- Challenges: OpenArchive operates in a low-resourced space, where securing consistent and operational funding is highly competitive. The primary challenge has been identifying diverse income streams to support their free-to-use open-source tool.
- Strategies: The SUSTAIN grant, funded by the U.S. State Department's DRL, has been instrumental in addressing these challenges. It provided Open Archive with the financial resources and dedicated time to develop a robust sustainability strategy and fundraising plan. This includes exploring various fundraising avenues (e.g. services, different strategies to build relationships with funders) and engaging with donors who are becoming increasingly aware of the importance of open-source tools.

2. App Maintenance

- Challenges: Ensuring ***Save*** is up-to-date, bug-free, and responsive to user needs while innovating new features is a constant balancing act.
- Strategies: The SUSTAIN project allowed OpenArchive to focus on strategic planning and allocate resources for app development and maintenance. The team works continuously to improve the app’s functionality, ensuring it remains relevant and useful for its users.

3. Team Well-Being

- Challenges: As a small organization, team members often wear multiple hats, leading to potential burnout and overburdening.
- Strategies: Emphasis is placed on maintaining a healthy work environment. Internews, a partner in the SUSTAIN project, has provided embedded consultants to alleviate some of the recruitment and operational pressures. This support has been crucial in maintaining a motivated and effective team. The SUSTAIN project also afforded time for core OpenArchive team members to hold strategy sessions. These strategy sessions enabled OpenArchive to take the time and plan for the organization’s sustainability.

4. Communications

- Challenges: Communications can be put on the back burner as more pressing tasks arise, making it difficult to share the tool’s message more broadly.
- Strategies: Through the SUSTAIN project, OpenArchive spent more time thinking about identity, their mission, and how they can contribute to the discourse and have greater impact (e.g. sharing our guides and promoting ***Save***). OpenArchive created a strategy document to guide improvement efforts, reorganized the communications team, and updated their mission language.

Discussion

Implications of Findings

The SUSTAIN project significantly impacted OpenArchive by providing financial and strategic support, enabling comprehensive sustainability planning. The project also fostered partnerships, enhancing the organization's capacity to innovate and maintain its tools. However, ongoing challenges in securing diverse funding, capacity onboarding contractors, and maintaining team well-being highlight areas for further improvement.

Comparison with Existing Literature

The case study aligns with existing research on the importance of diversified funding and continuous app maintenance but highlights unique strategies tailored to OpenArchive's context. The emphasis on team well-being is consistent with best practices in organizational management literature.

Limitations

- The study is limited by the specific context of OpenArchive and may not be generalizable to all non-profits.
- The reliance on qualitative data from a single interview and internal documents may limit the breadth of perspectives considered.

Recommendations for Future Research

Future research could explore the long-term impact of sustainability strategies on similar organizations and the role of community engagement in maintaining open-source tools. Comparative studies with other non-profit tech initiatives could provide broader insights into effective sustainability practices.

Summary of Key Findings

OpenArchive's collaboration with the SUSTAIN project has been instrumental in addressing their sustainability challenges, achieving significant improvements in financial stability, app maintenance, and team well-being.

This case study underscores the importance of dedicated sustainability plans and collaborative efforts in the non-profit sector. Open Archive's experience offers valuable lessons for the broader open-source community, highlighting the need for strategic planning and continuous innovation to ensure long-term success.

References

- Interview Transcript with Alex Esenler, the Deputy Director at Open Archive.
- OpenArchive OSS Tool Team Sustainability Action Plan, January 2024.

Appendices

Interview Transcript Excerpts

- "With the SUSTAIN grant, we were excited to have funds set aside to think about sustainability and strategize on fundraising and maintaining the organization and app."
- "Ensuring the app is up-to-date, bug-free, and addressing user requests while innovating new features is a constant balancing act."
- "As a small organization, our team members often wear multiple hats, leading to potential burnout. The support from Internews has been crucial in maintaining a motivated and effective team."

Sustainability Action Plan Highlights

- Detailed strategies for financial sustainability.
- Action steps for continuous app maintenance.
- Initiatives to promote team and organizational well-being.

Sustaining Bayanat: A Case Study on Enhancing Open Source Tool Sustainability for Human Rights Documentation

This case study examines Bayanat, an open-source data management app developed by the Syria Justice and Accountability Center (SJAC) for analyzing data from the Syrian conflict. It explores their sustainability strategies, particularly those developed under the SUSTAIN project funded by Internews. Key findings highlight the importance of documentation, app stability, and funding diversification.

Introduction

Background

Bayanat was developed by SJAC to organize and analyze vast amounts of data collected from the Syrian conflict, including over 2 million open-source videos depicting human rights violations. The app, built using PostgreSQL and Python, serves as a powerful tool for human rights organizations and others dealing with large datasets. However, ensuring its long-term sustainability posed significant challenges, leading SJAC to adopt innovative strategies and join the SUSTAIN tool team cohort.

Importance and Relevance

This case study is crucial for understanding how open-source tools in the human rights sector can achieve sustainability while maintaining their core mission.

Purpose and Objectives

The primary goals of Bayanat's sustainability efforts include improving documentation, increasing app stability, and securing long-term funding. This case study aims to detail the challenges faced, strategies implemented, and outcomes achieved through these initiatives.

Literature Review

Existing research highlights the critical role of open-source tools in human rights documentation. However, gaps remain in sustainable funding models and long-term maintenance strategies for such tools. Bayanat addresses these gaps through a comprehensive sustainability plan.

Methodology

This case study utilizes data from an interview with Roger Lu Phillips, the legal director of SJAC, and Bayanat's 2024 Sustainability Action Plan document. Qualitative analysis was conducted to identify key themes and strategies employed by the organization.

Findings

1. Financial Sustainability

- Challenges: Securing ongoing funding, especially for maintenance and bug fixes.
- Strategies:
    - Creating a tech development fund as part of indirect costs
    - Diversifying funding sources
    - Exploring new technologies (like AI integration) to open up new funding opportunities
    - Seeking support from funders like DRL for maintenance, not just new features

2. Technical Sustainability

- Challenges: Maintaining and improving the tool efficiently over time.
- Strategies:
    - Prioritizing efficient and simple design in coding
    - Planning for short, medium, and long-term development
    - Implementing a comprehensive process of development and testing
    - Leveraging the open-source community for identifying and addressing vulnerabilities
    - Exploring AI integration to improve efficiency in data analysis

3. Community Engagement and Growth

- Challenges: Building and maintaining an active user community to support the tool's development and relevance.
- Strategies:
    - Adopting an open-source philosophy to encourage community contributions
    - Creating a blog to share ideas, new functionalities, and use cases
    - Implementing localization to reach a wider user base

    - Collaborating with partners and participating in projects like SUSTAIN to share ideas and resources
    - Actively seeking and incorporating user feedback for improvements

Discussion

Implications of Findings

The SUSTAIN project significantly impacted Bayanat by addressing all three priority areas. It provided resources for hiring dedicated IT developers, enhancing technical sustainability. The project fostered an open-source community, supporting community engagement and growth. Additionally, it helped SJAC explore new funding strategies, contributing to financial sustainability.

Comparison with Existing Literature

The case study aligns with existing research on the importance of open-source development in human rights technology but highlights unique strategies tailored to Bayanat's context, particularly in balancing financial, technical, and community aspects of sustainability.

Summary of Key Findings

Bayanat's collaboration with the SUSTAIN project has been instrumental in addressing their sustainability challenges across financial, technical, and community engagement fronts. This comprehensive approach has led to significant improvements in funding strategies, app stability and development, and user community growth.

This case study underscores the importance of a multi-faceted approach to sustainability in the open-source sector. Bayanat's experience offers valuable lessons for similar projects, highlighting the need for strategic planning in finance, technical development, and community engagement to ensure long-term success.

References

- Interview with Roger Lu Phillips, Legal Director at SJAC
- Bayanat Sustainability Action Plan 2024

Appendices

Interview Transcript Excerpts

- "If we can use the data set that we've created in order to train an AI tool in order to conduct that analysis on its own in the future, at least do a first cut of that, then I think we can save a huge amount of resources and reduce the psychosocial impact of reviewing these human rights abuses manually."
- "It's important for partners with similar interests to be brought together in spaces where they can share ideas and collaborate with one another. You know, those organizations

that are working in the same space or in parallel spaces, don't always have the opportunity to convene and discuss the common challenges that they may be encountering."

Sustainability Action Plan Highlights

- Detailed strategies for financial sustainability.
- Action steps for continuous app maintenance.
- Initiatives for better working with partner organizations.
- Milestones for monitoring and evaluation plan.

Sustaining OpenWISP: A Case Study on Enhancing Open-Source Tool Sustainability for Network Management

This case study examines OpenWISP, an open-source collection of tools and libraries for deploying and managing networking equipment across diverse geographic locations. It explores their sustainability strategies, particularly through the SUSTAIN project funded by Internews. Key findings highlight the importance of improving communication, enhancing technical documentation, establishing a board of advisors, and securing long-term funding.

Introduction

Background

OpenWISP was founded to help organizations deploy and manage network equipment across different geographic locations. Initially focused on WiFi hotspot management, it has evolved into a versatile solution for various networking use cases. Federico Capoano, a full-stack developer, leads the project.

Importance and Relevance

This case study is crucial for understanding how open-source tools in the network management sector can achieve sustainability while maintaining their core mission of empowering communities through accessible technology.

Purpose and Objectives

The primary goals of OpenWISP's sustainability efforts include improving communication of its value, enhancing technical documentation, nominating a board of advisors, and securing long-term funding. This case study aims to detail the challenges faced, strategies implemented, and outcomes achieved through these initiatives.

Methodology

This case study utilizes data from an interview with Federico Capoano, the project lead of OpenWISP, and OpenWISP's 2024 Sustainability Action Plan document. Qualitative analysis was conducted to identify key themes and strategies employed by the organization.

Findings

1. Improving Communication of OpenWISP's Value

- Challenges: Attracting new users and achieving steady growth.
- Strategies: Improving the website's look and feel, updating copywriting, images, and videos.

2. Enhancing Technical Documentation

- Challenges: Helping newcomers quickly get up to speed with using OpenWISP.
- Strategies: Improving technical documentation, providing step-by-step tutorials on basic use cases.

3. Nominating a Board of Advisors

- Challenges: Mitigating the burden of responsibility and single point of failure in decision-making.
- Strategies: Nominating experienced industry professionals to provide guidance, feedback, and advice.

4. Securing Long-Term Funding

- Challenges: Finding diverse funding sources in a competitive space and ensuring financial stability for ongoing development and maintenance.
- Strategies:
    - Exploring various funding avenues, including commercial support, public funding, and collaborations with organizations like Internews.
    - Applying for additional funding opportunities, with a goal of securing at least two more funding programs.
    - Developing a financially sustainable business model compatible with free software values that can onboard new customers without human interaction.
    - Allocating resources towards community-focused efforts and maintenance.

Discussion

Implications of Findings

The SUSTAIN project significantly impacted OpenWISP by providing resources for improving communication, enhancing documentation, and exploring new funding opportunities. The project also fostered collaboration with other organizations and experts in the field.

Comparison with Existing Literature

The case study aligns with existing research on the importance of community engagement and diverse funding sources in open-source sustainability but highlights unique strategies tailored to OpenWISP's context.

Limitations

- The study is limited by the specific context of OpenWISP and may not be generalizable to all open-source tools.
- The reliance on qualitative data from a single interview and internal documents may limit the breadth of perspectives considered.

Recommendations for Future Research

Future research could explore the long-term impact of community engagement strategies on open-source project sustainability and the role of advisory boards in guiding open-source initiatives.

Summary of Findings

OpenWISP's collaboration with the SUSTAIN project has been instrumental in addressing their sustainability challenges, achieving significant improvements in communication, documentation, governance, and funding strategies. The project emphasized the importance of community engagement, diversified funding, and strategic planning for long-term sustainability.

This case study underscores the importance of dedicated sustainability plans and collaborative efforts in the open-source sector, offering valuable lessons for similar projects in the open-source ecosystem.

References

- Interview with OpenWISP Lead Project, Federico Capoano.
- OpenWISP Sustainability Action Plan 2024

Appendices

Interview Transcript Excerpts

- "We've been focusing a lot on communication really these last six months. Putting together all this information on our success stories, the great projects we have done, and packaging it in a nice way so we can use it to look for more funding and basically sell ourselves a little bit better."
- "Our documentation was scattered around different repositories and modules. We had one website and from that website we were linking different GitHub repositories. So now we are working on bringing all that information into one website. It will take some time, but we are working on it. This is really important for helping newcomers quickly get up to speed with using OpenWISP."
- "We've been putting together a board of advisors, some people who have a stake in the project and can offer advice and put their names on the new website. This helps mitigate the burden of responsibility and avoids a single point of failure in decision-making."
- "We need to think about ways to sustain our work. My personal opinion is that a mixed strategy can yield nice results. We're exploring various funding avenues, including commercial support, public funding, and collaborations with organizations like Internews. We're also working on developing a financially sustainable business model that's

compatible with free software values and can onboard new customers without human interaction."

Sustainability Action Plan Highlights

- Detailed strategies for financial sustainability.
- Action steps for better communication around OpenWISP packaging.
- Initiatives for enhancing technical documentation.
- Milestones for monitoring and evaluation plan.

Afterword: Embracing Sustainability in FLOSS

As we conclude the "Sustainability Guide for FLOSS Tools," we celebrate our shared commitment to the long-term success of Free/Libre/Open-Source Software (FLOSS). This guide, short yet comprehensive, marks a step forward in our collective journey towards a sustainable open source future.

In the rapidly evolving landscape of FLOSS, sustainability is not just a buzzword; it's a necessary ethos. This guide serves as a primer, providing you with essential insights and directing you to more in-depth resources, understanding the time constraints and practical needs of developers. We've traversed various domains, from technical advancements to community engagement, emphasizing the multifaceted nature of sustainability in the FLOSS ecosystem.

Remember, this guide is a beginning, not an end. The journey towards sustainability is ongoing, and as developers and contributors, you play a crucial role in shaping a thriving open-source community. By embracing the principles of this guide, you contribute not only to your projects' success but also to the larger narrative of open source as a bastion of innovation and collaboration.

In closing, we extend our heartfelt gratitude for your dedication and enthusiasm. Your commitment to FLOSS is what drives this community forward. Let this guide be a beacon, guiding you towards sustainable practices that ensure the vitality and success of your projects in the open-source world.