From Picket Lines to Pull Requests

Labor Organizing Lessons for Building
Free and Open Source Software Communities

A Strategic Framework for the FreeMoCap Foundation

January 29, 2026

Contents

Abstract

The most successful labor campaigns and the most sustainable open source projects share a counterintuitive truth: **community-building precedes and outweighs any tactical or technical consideration**. Jane McAlevey's organizing mantra—"community over contract"—mirrors Apache's founding principle of "community over code." This parallel reveals why FOSS projects that treat community development as seriously as code development tend to survive founder transitions, corporate pressure, and the burnout epidemic that claims **60% of maintainers** who have quit or considered quitting.

But the parallel extends further than most FOSS projects recognize. Labor movements understand that their power comes not primarily from leaders or professional organizers, but from the **broad base of ordinary workers** who may never hold union office but whose participation makes collective action possible. Similarly, FOSS communities depend not just on maintainers and code contributors, but on the vast base of **users**—artists, researchers, hobbyists, educators—who use the software, populate forums, create tutorials, evangelize to peers, and constitute the community without which the project has no purpose.

For a project like FreeMoCap transitioning from scrappy origins to foundation governance, labor organizing offers a tested playbook for building durable, democratic institutions that can grow without losing their soul. This report synthesizes decades of labor movement strategy, FOSS governance case studies, cooperative economics, and sustainability research into actionable frameworks—with particular attention to mobilizing the full spectrum of community members, from passive users to core maintainers.

1 Introduction: The Community Beyond the Codebase

When we think about open source communities, we often default to thinking about developers: the people who write code, review pull requests, fix bugs, and maintain infrastructure. This developer-centric view misses the majority of most FOSS communities—and misses the source of their actual power.

Consider Blender, the 3D creation suite that has become the gold standard for open source creative software. Blender has a relatively small core development team, but its community numbers in the **millions**. The vast majority of Blender users will never write a line of C++ code. They are artists, animators, architects, game designers, educators, and hobbyists who use Blender to create. They populate forums answering each other's questions. They create tutorials on YouTube that introduce newcomers to the software. They evangelize Blender to colleagues skeptical of open source tools. They contribute to the Blender Development Fund not because they understand the codebase, but because they value what the software enables them to do.

Without these users, Blender would be a technically impressive but irrelevant project. **The community is not an appendage to the software; the software exists to serve the community.**

This insight connects directly to labor organizing wisdom. Unions don't derive their power primarily from professional organizers or union staff. They derive power from **ordinary workers**—people who may never hold union office, who aren't labor law experts, who simply want fair treatment and are willing to stand together with coworkers to achieve it. The organizer's job is to identify, develop, and mobilize these people—not to replace them.

For FreeMoCap, this reframing is essential. The foundation's constituency includes:

- **Researchers** using motion capture for biomechanics, psychology, sports science, clinical assessment
- **Artists and animators** seeking affordable motion capture for creative projects
- **Educators** teaching movement analysis, computer vision, or 3D animation
- **Hobbyists and makers** exploring what's possible with accessible technology
- **Contributors** who improve documentation, answer forum questions, or report bugs
- **Developers** who contribute code, review PRs, or maintain modules

Each group has different needs, different relationships to the project, and different potential contributions. A strategy that focuses only on recruiting developers ignores the vast majority of the community and the vast majority of the project's potential impact.

This report applies labor organizing principles to the full spectrum of FOSS community building—from converting skeptical potential users, to activating passive users into community members, to developing community members into evangelists, to supporting the maintainers who keep everything running.

2 Deep Organizing: Reaching Beyond the Already-Converted

2.1 The Distinction That Changes Everything

Labor organizer Jane McAlevey distinguished between three tiers of engagement that translate directly to FOSS contexts:[1]

- **Advocacy** involves writing checks while professionals handle problems—the FOSS equivalent of users who donate but never participate in community life.
- **Mobilizing** gets existing supporters off the couch—rallying current community members for releases, hackathons, or funding drives.
- **Deep organizing** does something fundamentally different: it persuades people who don't yet agree to change their opinions and join the cause.

McAlevey's insight cuts against instincts that keep FOSS communities insular: "For organizers, we wake up every morning asking how to engage the people who don't agree with us—or who think they don't agree with us. These folks are definitely not part of our social media feeds, and they're not coming to our activist meetings."[2]

Applied to FreeMoCap, deep organizing means systematically reaching:

- Researchers using expensive proprietary motion capture systems who assume open source can't match quality
- Artists skeptical that "free software" could meet professional standards
- Educators who've never considered motion capture accessible for their contexts
- Potential contributors—not just developers, but technical writers, designers, community managers—who don't see themselves as "open source people"

The goal is not broadcasting to people already in your orbit, but **building relationships with people outside it** and persuading them through those relationships.

2.2 Structure Tests: Measuring Real Capacity

The technique that operationalizes deep organizing is the **structure test**: any collective action measuring what proportion of your constituency participates in activities of increasing commitment and risk. McAlevey defines success as "supermajority" participation—80%+ for most actions.

A FOSS structure test progression might look like:

1. Can 80% of mailing list members complete a simple survey about their needs?
2. Can 80% of active forum participants attend a community call?

[1] Jane McAlevey, *No Shortcuts: Organizing for Power in the New Gilded Age* (Oxford University Press, 2016).

[2] Current Affairs, "Jane McAlevey on How To Organize for Power," April 2019.

3. Can 80% of community call attendees commit to a specific small action (share the project, write a short testimonial)?
4. Can 80% of regular contributors participate in governance decisions?
5. Can the community collectively push back against corporate exploitation or unfair criticism?

Each test builds capacity while revealing organizational weak points before they become crises. If you can't get 80% participation in low-stakes actions, you'll never get it for high-stakes ones.

2.3 Public Narrative: Story of Self, Us, and Now

Marshall Ganz, former United Farm Workers organizer now at Harvard, contributed another essential framework: **public narrative**.[3] His "Story of Self, Story of Us, Story of Now" structure builds collective identity through shared meaning:

- **Story of Self**: Why am I called to this work? What personal experiences and values drive my involvement?
- **Story of Us**: What brings our community together? What values and experiences do we share?
- **Story of Now**: Why must we act collectively, and why now? What's the urgent challenge requiring response?

For FreeMoCap, cultivating public narrative means:

- Collecting and sharing origin stories from users across domains—the researcher who couldn't afford commercial systems, the artist who wanted creative freedom, the educator who saw students light up
- Building a canonical narrative about why democratized motion capture matters—connecting individual stories to shared purpose
- Framing specific releases, campaigns, or governance decisions as "Story of Now" moments requiring community response

3 The Ladder of Engagement: From Passive User to Core Contributor

3.1 Designing Pathways, Not Hoping for Miracles

Labor movements developed the **ladder of engagement** concept: people should progress through stages of increasingly deep commitment, with clear rungs and intentional transitions between levels.

[3] Marshall Ganz, "Public Narrative, Collective Action, and Power," in *Accountability Through Public Opinion*, ed. Sina Odugbemi and Taeku Lee (World Bank, 2011).

Most FOSS projects have a broken ladder. They recruit users and hope some magically become maintainers, without designing the intermediate steps. The result: a vast passive user base, a tiny burned-out maintainer core, and almost nothing in between.

3.2 The Blender Model: Users as Community Foundation

Blender Foundation provides the clearest model for building community around users who will never write code.[4] Their approach includes:

- **Blender Cloud**: Subscription service providing tutorials, assets, and production files—creating ongoing relationship with users
- **Blender Open Movies**: Production projects (like *Sprite Fright* and *Charge*) that employ artists, showcase capabilities, and generate community excitement
- **Blender Conference**: Annual gathering mixing developers and artists, technical talks and creative showcases
- **Blender Artists Community**: Forums where users help users, reducing maintainer burden while building peer relationships
- **Development Fund**: Transparent funding model where individuals and corporations support development—creating stakeholder investment without governance capture

The genius is recognizing that **artists evangelizing to other artists** is more powerful than developers trying to recruit artists. Users who love the software become its most effective advocates precisely because they speak the language of potential users.

3.3 A Comprehensive Engagement Ladder for FreeMoCap

A structured engagement pathway for FreeMoCap might include:

The critical insight from labor organizing: **organizations must intentionally design pathways between levels and track where people are on the ladder**. Moving someone from passive user to engaged community member requires different tactics than moving a contributor to maintainer.

3.4 Making Artists into Scientists, Scientists into Artists

FreeMoCap occupies a unique position at the intersection of domains that rarely overlap: motion capture technology traditionally siloed between expensive research labs and film/game production studios. This creates an opportunity that most FOSS projects lack: the chance to **cross-pollinate communities**.

- **Artists → Scientists**: Creative users exploring motion capture develop intuitions about biomechanics, data quality, and measurement that formal training often lacks. An animator who's captured thousands of movements understands embodied knowledge that complements academic expertise. Creating pathways for artists to contribute to research applications—or to pursue research themselves—expands both communities.

[4] Blender Foundation organizational structure and community programs.

Table 1: Ladder of Engagement for FOSS Communities

Level	**Description**	**Transition Mechanism**
Passive User	Downloads software, uses it	Welcome sequence, documentation
Engaged User	Joins mailing list, follows updates	Forum invitation, community calls
Community Member	Participates in forums, helps others	Recognition, mentorship invitation
Contributor	Creates tutorials, reports bugs, improves docs	Contributor guide, paired projects
Evangelist	Actively recruits others, presents at conferences	Speaking opportunities, case study features
Module Owner	Maintains specific area (code or non-code)	Gradual responsibility increase
Governance Participant	Votes, serves on committees, shapes direction	Clear pathways to leadership

- **Scientists → Artists**: Researchers with deep technical knowledge often lack creative outlets or exposure to how their tools might enable artistic expression. Exposing researchers to creative applications can spark new research directions while making the research community more accessible to outsiders.
- **Users → Community Members**: The transition from "person who uses the software" to "person who identifies as part of the community" is the critical first step. It happens through relationship—someone answers your forum question, you attend a community call, you see your use case reflected in project communications. Designing for this transition means creating touchpoints where passive users encounter community.
- **Community Members → Evangelists**: People who feel ownership over a project naturally want to share it. The foundation's job is making evangelism easy: providing shareable assets, case study templates, talking points for common objections, and recognition for those who spread the word.

4 Governance: From Founder-Led to Community-Accountable

4.1 Learning from Governance Crises

The Rust governance crisis of 2021–2022 offers the starkest recent lesson on institutional design. When the entire moderation team resigned citing "structural unaccountability" of the Core Team, they revealed that **no mechanism existed for oversight**—the governance document had what developers might call “undefined behavior.”[5]

The resolution required extensive community consultation, ultimately dissolving the Core Team in favor of a representative Leadership Council with clear lines of accountability. The lesson applies directly to foundations transitioning from founder-led to institutional governance: **every team must be accountable to someone, including the top**.

Python’s post-BDFL transition succeeded partly because the PEP (Python Enhancement Proposal) process already existed for structured decision-making. When Guido van Rossum stepped down after the contentious walrus operator debate in 2018, the community used that same process to debate seven governance proposals and select a Steering Council model. The transition worked because mechanisms for community deliberation predated the crisis.

Node.js demonstrates how fork threats can drive governance reform. Dissatisfaction with Joyent’s control led four of Node’s top five contributors to create io.js with open governance in 2014. Within a year, the Node.js Foundation formed, adopted the open governance model, and merged the projects. The key insight: **corporate stewardship must allow genuine community input**, or the community will exercise its ultimate check—forking the codebase.

4.2 Governance Models Compared

Different governance models suit different project stages and communities:

BDFL (Benevolent Dictator for Life) provides clear decision-making and consistent vision but creates succession crises and bottlenecks as projects scale. Python, Linux, and Ruby used variants of this model in their early years. It works when the BDFL has broad trust and bandwidth; it fails when either runs out.

Meritocracy, exemplified by Apache’s model, grants influence through demonstrated contribution.[6] The Apache Way emphasizes “community over code,” requires all discussions on public mailing lists, and explicitly bars sponsors from governance influence. Critiques note that how “merit” gets recognized can mask bias and favor those with more time to contribute—often those with more privilege.

Electoral democracy, as practiced by Debian for decades, enables broad participation through annual elections, constitutional frameworks, and community votes on major decisions. It’s slower but more resilient to capture. Debian’s model includes provisions for expelling developers, recalling leaders, and amending the constitution—mechanisms that provide accountability even when rarely used.

Foundation-backed governance through organizations like the Linux Foundation provides legal, financial, and marketing support.[7] The trade-off involves corporate membership models where paying members get board seats, creating potential conflicts between community and sponsor interests.

[5] Rust moderation team resignation statement, November 2021.

[6] Apache Software Foundation, “The Apache Way,” `https://www.apache.org/theapacheway/`.

[7] Linux Foundation governance documentation.

4.3 Including Users in Governance

Most FOSS governance discussions focus on developers, but projects serving substantial user communities face a question: **how do users have voice in decisions affecting them?**

Blender's approach involves transparency without direct user governance: development priorities are discussed publicly, the Development Fund creates a sense of stakeholder investment, and the Blender Conference mixes developers and users. But formal governance remains with the Blender Foundation board and development team.

Alternative models might include:

- **User advisory councils** with defined consultation rights on UX decisions
- **Multi-stakeholder boards** including user representatives alongside developers and funders
- **Community assemblies** for major strategic decisions affecting project direction
- **Survey-driven prioritization** giving users structured input on roadmaps

The labor parallel is instructive: unions represent all workers in a bargaining unit, not just activists or leaders. Similarly, FOSS governance should consider the interests of the full community, not just those with commit access.

4.4 Foundation Structure Options

For projects considering formalization, the choice between fiscal sponsorship and independent foundation depends on scale and goals:

Fiscal sponsorship through organizations like Software Freedom Conservancy, NumFOCUS, or Open Collective provides nonprofit status, handles administrative and legal burdens, and lets projects focus on community and code.[8] SFC charges 10% of donations; projects maintain governance autonomy while SFC handles "everything else." This model suits most small-to-medium projects and can continue indefinitely—NumFOCUS hosts NumPy, pandas, and Jupyter without those projects needing independent foundations.

Apache Software Foundation represents the vendor-neutral membership model: all participants are individuals (not company representatives), sponsors explicitly have no governance influence, and projects enter through an Incubator teaching "The Apache Way."[9] This model protects independence but requires projects to adopt Apache's governance culture.

Linux Foundation exemplifies corporate membership models with tiered sponsorship ($500,000+ Platinum members get board seats). This generates substantial funding but creates governance tensions between community and corporate interests.

For FreeMoCap, the fiscal sponsorship path through NumFOCUS (given the scientific computing focus) or Software Freedom Conservancy likely makes sense at current scale. Key questions for any formalization: How are decisions made? Who can vote? How do new

[8] Opensource.com, "How a fiscal sponsor can help your open source project grow," January 2019.

[9] Apache Software Foundation governance documentation.

contributors gain trust and responsibility? How is the foundation accountable to its full community—users as well as developers?

5 Commons Governance: Principles for Shared Resources

5.1 Ostrom's Design Principles

Elinor Ostrom's Nobel Prize-winning research on commons governance provides eight design principles that communities successfully managing shared resources tend to exhibit.[10] The SustainOSS community has translated these directly to open source contexts:[11]

1. **Clear boundaries**: "Our projects are clear and explicit about who is involved and in what ways." This means defined membership tiers with explicit rights and responsibilities at each level—including user tiers, not just contributor tiers.

2. **Proportionality between rules and local conditions**: Matching governance complexity to community scale. What works for a five-person project differs fundamentally from a five-hundred-person project.

3. **Collective-choice arrangements**: "Our projects offer structures and processes to support fair decisions." Governance legitimacy depends on those affected by rules being able to participate in modifying them.

4. **Monitoring by accountable parties**: "Participants have a shared understanding of purpose and progress." Monitors must be accountable to the community, not just to leadership.

5. **Graduated sanctions**: Using "a range of methods to encourage re-alignment—from gentle admonishment up to removal from the community." Proportional responses preserve community while addressing violations.

6. **Accessible conflict resolution**: Creating arenas for resolving disputes before crises erupt. This must be "accessible so all participants feel like they can participate."

7. **Recognition of rights to organize**: External authorities (platforms, governments, corporate sponsors) must not undermine community self-governance.

8. **Nested governance** for larger projects: "Multiple parties and various scales"—modules, working groups, and sub-projects with appropriate autonomy within larger structures.

5.2 Digital Commons Differ from Natural Commons

A critical insight for digital commons: unlike natural resources where the challenge is preventing over-extraction, **open source must stimulate both use and contribution**. The

[10] Elinor Ostrom, *Governing the Commons* (Cambridge University Press, 1990).

[11] SustainOSS, "The Principles of Governing Open Source Commons," https://sustainoss.pubpub.org/pub/jqngsp5u/.

code itself is non-rivalrous—my using FreeMoCap doesn't prevent your using it. The scarce resource isn't the code; it's **maintainer attention and time**.

This inverts traditional commons logic. Nadia Eghbal's research found that **low-quality contributions can be worse than no contributions** because they consume scarce review bandwidth.[12] A flood of poorly-documented feature requests, duplicate bug reports, or drive-by PRs that don't follow project standards creates work rather than reducing it.

The implication: community management must balance encouraging participation with protecting maintainer capacity. This might mean:

- Templates and bots that guide contributions toward useful formats
- Tiered support where community members help users before issues reach maintainers
- Clear documentation of what kinds of contributions are welcome
- Saying no gracefully to contributions that don't fit project direction

6 Cooperative Models and Platform Cooperativism

6.1 Worker Cooperatives as Organizational Model

Platform cooperativism, developed by Trebor Scholz and Nathan Schneider, offers a framework for technology owned and governed by stakeholders rather than shareholders.[13] The CoTech network of UK technology worker cooperatives demonstrates practical application: member organizations are fully worker-owned, decisions happen by consensus on Loomio, and annual gatherings build the trust that sustains remote collaboration.

Mondragon, the world's largest worker cooperative federation with **70,000+ workers across 266 businesses**, provides governance lessons applicable to FOSS foundations:[14]

- One-worker-one-vote regardless of capital ownership
- Maximum **6x pay ratios** between highest and lowest earners
- Worker reallocation across cooperatives during downturns rather than layoffs
- Federation structure providing risk pooling, knowledge sharing, and economies of scale while maintaining local autonomy
- A cooperative university for ongoing education and leadership development

[12] Nadia Eghbal, *Working in Public: The Making and Maintenance of Open Source Software* (Stripe Press, 2020).

[13] Platform Cooperativism Consortium, https://platform.coop/.

[14] Oxford Saïd Business School, "Mondragon Case Study," 2018.

6.2 Cooperative Principles for FOSS

Nathan Schneider's recent book *Governable Spaces* warns that online platforms suffer from "implicit feudalism"—a bias for building communities as fiefdoms with all-powerful admins and benevolent dictators.[15] He argues this "trains us to give up on our communities' democratic potential" and calls for designing tools that "bring people up the ladder of engagement, from volunteering to leadership."

For FOSS foundations, cooperative principles suggest:

- **Multi-stakeholder governance** including users, funders, and dependent projects alongside developers
- **Explicit documentation of ownership and control**—who can make what decisions?
- **Open-sourcing governance** itself, not just code—making organizational structures transparent and forkable
- **Connecting to broader solidarity economy movements** for mutual support and shared learning

7 Preventing Burnout: Structural Solutions, Not Just Self-Care

7.1 The Sustainability Crisis

The sustainability crisis in open source is acute. Tidelift surveys find **60% of maintainers are unpaid**, and **60% have quit or considered quitting**.[16] Solo maintenance is endemic—61% of unpaid maintainers work alone. The 2024 xz-utils backdoor demonstrated how burnout creates security vulnerabilities: the original maintainer, overwhelmed after years of solo work, accepted help from what turned out to be malicious actors who inserted a backdoor after gaining commit access.

Research on activist burnout offers complementary insights. Paul Gorski's studies identified a **"culture of martyrdom"** in movements as a major burnout driver—the same selflessness that motivates involvement becomes self-destructive when it prevents acknowledging limits.[17] The research found that activists who neglect burnout signals tend to leave activism at least temporarily, meaning **burnout helps injustice persist by removing defenders**.

[15] Nathan Schneider, *Governable Spaces: Democratic Design for Online Life* (University of California Press, 2024).

[16] Tidelift, "The 2023 State of the Open Source Maintainer Report."

[17] Chen & Gorski, "Burnout in Social Justice and Human Rights Activists," *Journal of Human Rights Practice*, 2015.

7.2 Nadia Eghbal's Reframe

Nadia Eghbal's research reframes the problem: **maintainer attention, not code, is the scarce resource**. Her proposed "one-way mirror" model would let users see code and discussions but require opt-in from maintainers before opening PRs or issues.[18]

Her "Christmas lights" analogy captures sustainable motivation: "People put up decorations because they want to, not because they must." The implication: design for intrinsic motivation, not obligation. Create conditions where maintenance feels rewarding rather than burdensome.

7.3 Collective Care Practices

Movement contexts have developed collective care practices that FOSS communities could adapt:[19]

- **Buddy systems and small group check-ins**: Pairing community members for mutual support
- **Rotating draining tasks**: So no one is "always the one" handling conflict, support requests, or tedious maintenance
- **Well-being check-ins during meetings**: Normalizing attention to how people are doing, not just what they're producing
- **Shared meals and bonding**: "Feeding the soul of the movement" through positive experiences, not just work
- **Explicit permission to rest**: Countering cultures where stepping back feels like betrayal

7.4 Structural Interventions

Individual self-care cannot substitute for structural change. Key interventions include:[20]

- **Distributed leadership**: Preventing dependence on single points of failure. No one person should be indispensable.
- **Documentation of philosophy, not just code**: Making onboarding possible by explaining *why*, not just *how*.
- **Creating tiers of contribution**: Not all engagement should require maintainer attention. Users helping users, contributors reviewing contributors, reduces load on core maintainers.
- **Aggressive automation**: Tests, bots, templates, and increasingly AI assistance can reduce manual burden.

[18] Eghbal, *Working in Public.*

[19] Museum of Protest, "The Psychology of Activism and Movement Longevity."

[20] GitHub, "Maintaining Balance for Open Source Maintainers," https://opensource.guide/maintaining-balance-for-open-source-maintainers/.

- **Paying maintainers**: Transforms sustainability more than any other intervention. Tidelift found paid maintainers are 55% more likely to implement critical security practices.[21]

7.5 Boundary Setting

Learning to say no is stewardship, not rejection. Mike McQuaid, longtime Homebrew maintainer, articulates the principle: "To meaningfully trust others, you cannot be someone who says yes to every request. In doing so, you maintain no boundaries, professionally or personally, and will not be a reliable coworker."[22]

Documentation should include what you're interested in doing and not doing, with expectations set before problems escalate. A project that clearly states "we don't accept feature requests for X" creates less conflict than one that lets requests accumulate unanswered.

8 Inoculation: Preparing for Predictable Challenges

8.1 The Labor Tactic

Labor organizers developed **inoculation tactics** to prepare workers for management pushback before campaigns begin. By discussing in advance how employers fight organizing—captive-audience meetings, anti-union consultants, divide-and-conquer tactics, sudden improvements designed to undercut campaigns—organizers prevent workers from being swayed when these predictable responses arrive.

The psychological mechanism: when people encounter something they've been warned about, it feels confirmatory rather than surprising. "They told me the company would do exactly this" builds commitment rather than creating doubt.

8.2 Inoculating FOSS Communities

For FOSS communities facing potential challenges, inoculation means discussing scenarios openly before they happen:

- **Corporate co-optation**: Corporations offering resources with strings attached, hiring away key contributors, creating competing projects, influencing governance through funding

- **Community conflict**: Inevitable disagreements, personality clashes, or governance disputes that can fracture communities

- **Burnout cycles**: Periods when maintainers need to step back, and how the community will handle transitions

- **External criticism**: Skeptics who dismiss open source capabilities, media coverage that misrepresents the project, or competitive attacks

[21] Socket, "The Unpaid Backbone of Open Source," 2024.

[22] Quoted in GitHub's maintainer guides.

- **Success challenges**: Scaling problems, governance growing pains, and maintaining culture through rapid growth

When community members recognize these patterns because they've been discussed in advance, they remain committed rather than confused. This builds trust through the message: "We predicted this would happen, and we have a plan."

8.3 The AEIOU Model

The **AEIOU model** from IWW organizing provides a memorable sequence applicable to FOSS adoption campaigns:

- **Agitate**: Surface discontent with the status quo. What problems do proprietary tools create? What frustrations do potential users experience?
- **Educate**: Explain how collective action (adopting open alternatives, contributing to shared infrastructure) addresses those problems.
- **Inoculate**: Prepare advocates for opposition. What will skeptics say? How will vendors respond? What FUD will you encounter?
- **Organize**: Delegate tasks, build structure, identify internal champions, create committees.
- **Unite/Act**: Take collective action—coordinated adoption, public advocacy, joint development.

9 Movement Building: Coalitions and Accessible Framing

9.1 Lessons from Right to Repair

The Right to Repair movement offers the most instructive recent example of technical concerns achieving mainstream political success.[23] Starting from trade associations protecting independent repair shops, the movement built coalitions including iFixit, EFF, U.S. PIRG, Consumer Reports, and disability advocates.

Their **groundswell strategy**—"legislators listen to constituents"—combined with direct ballot initiatives achieved remarkable results. Massachusetts passed Right to Repair with **87.7% support** in referendum.[24] As of 2024, all fifty states have some form of Right to Repair legislation.

Key success factors translate to FOSS advocacy:

- **Bipartisan appeal**: Supported across party lines by framing around consumer choice and property rights
- **Concrete relatable framing**: "Will this make my screen repair cheaper?" rather than abstract rights arguments

[23] The Repair Association, "Our Mission and History," https://www.repair.org/history.

[24] iFixit, "Right to Repair Progress," https://www.ifixit.com/right-to-repair-progress.

- **Coalition breadth**: Environmental groups, disability advocates, farmers, independent businesses, consumers
- **State and local focus**: Building victories at accessible levels before national campaigns
- **Connecting technical issues to broader values**: Environment, consumer rights, economic fairness

9.2 Framing FOSS for Non-Technical Audiences

Public attitudes already favor digital autonomy messages. Pew Research found **81% of Americans** feel they have little or no control over corporate data collection, **77%** have little or no trust in social media leaders on privacy, and **72%** support more government regulation of company data use—with bipartisan support across party lines.

The challenge is connecting FOSS to these existing concerns. Effective frames for non-technical audiences include:

- **Rights-based framing**: "Right to repair, improve, and control your own tools"
- **Security and control framing**: "Take Back CTRL," "own your data," "know what your software does"
- **Environmental framing**: E-waste reduction, software longevity, circular economy
- **Economic framing**: Reduced costs, avoiding vendor lock-in, local expertise development
- **Democratic/sovereignty framing**: Digital self-determination, independence from Big Tech
- **Educational framing**: Learning how things work, developing real skills, demystifying technology

The Right to Repair insight applies: lead with problems people experience, not ideology. "Frustrated that your expensive software stopped working after an update you didn't want?" resonates more broadly than "software freedom is a fundamental right."

9.3 The Tech Literacy Crisis

A broader context for FOSS advocacy: younger generations are increasingly **tech consumers rather than tech literate**. Raised on locked-down smartphones and corporate platforms designed to hide complexity, many young people have less understanding of how technology works than previous generations who grew up with more accessible (if less polished) systems.

This creates both challenge and opportunity. The challenge: potential users may not understand why open source matters or have skills to evaluate it. The opportunity: FOSS communities can position themselves as spaces for **learning and empowerment**—places where people develop genuine technical understanding rather than passive consumption.

FreeMoCap specifically offers a pathway from user to technically literate participant that few software projects can match. Motion capture involves hardware, software, mathematics, and domain expertise in ways that invite deep engagement. An animator learning FreeMoCap might discover interests in computer vision; a researcher might develop programming skills; an educator might understand their subject more deeply through building their own tools.

10 Comprehensive Campaigns: Combining Multiple Tactics

10.1 Bronfenbrenner's Research Applied

Kate Bronfenbrenner's research at Cornell found that unions employing **comprehensive campaigns**—combining multiple tactics simultaneously—achieved dramatically higher success rates.[25] Her statistical finding: campaigns utilizing five or more organizing tactics had **74% first contract rates compared to 58%** for less comprehensive efforts.

The tactics that mattered included: representative organizing committees, house calls, solidarity days, community support, rank-and-file participation, and advance preparation for employer opposition.

10.2 Translating to FOSS Adoption

A comprehensive campaign for institutional FreeMoCap adoption might combine:

- **Internal champion committees**: Identify respected researchers, artists, or educators who will advocate from within institutions
- **Peer-to-peer advocacy**: Current users speaking to potential users in their own language and context
- **Issue-focused messaging**: Addressing specific pain points (cost, vendor lock-in, customization needs, reproducibility) rather than leading with ideology
- **Escalating demonstrations**: From individual pilots to departmental trials to institutional proposals
- **External alliances**: Partnerships with academic institutions, professional associations, creative communities
- **Implementation support**: Having resources ready before adoption decisions, not scrambling afterward
- **Success documentation**: Case studies, testimonials, and measurable outcomes that advocates can point to

The key insight: any single tactic can fail, but **comprehensive approaches create redundancy** and multiple pathways to success.

[25] Bronfenbrenner & Hickey, "Blueprint for Change," 2003.

11 The Free Rider Problem

11.1 No Perfect Solution, Only Cultural Mitigation

Both unions and FOSS communities face people who benefit without contributing. Unions historically used closed shops (now illegal in the US) or agency fees (struck down by *Janus v. AFSCME* in 2018 for public sector unions). Post-*Janus* union responses focus on demonstrating value through wins, creating member-only benefits, building workplace culture where non-membership is socially costly, and intensive one-on-one conversations to convert non-members.

FOSS applications include:

- **Social recognition**: Public acknowledgment of contributors at all levels—not just code, but documentation, community support, evangelism
- **Tiered access**: Early releases, governance participation, or exclusive community spaces for active contributors
- **Member-only benefits**: Premium support, mentorship, networking opportunities
- **Cultural expectation**: Building community norms where contributing back is valued even if not required

The strategic insight: **make contribution the path of least resistance** rather than trying to punish non-contribution. If participating feels rewarding and is made easy, more people will participate.

11.2 Users Are Not Free Riders

A critical distinction: **users are not free riders**. They are the community the project exists to serve. Without users, there is no project—only code no one uses.

Users contribute by:

- Using the software (validating its purpose)
- Reporting bugs and usability issues
- Answering questions in forums
- Creating tutorials and documentation
- Evangelizing to potential users
- Providing feedback on priorities and direction
- Funding development through donations
- Building the ecosystem of knowledge around the software

The goal is not extracting code contributions from users who will never code, but **creating pathways for the contributions users can make**—and recognizing those contributions as valuable.

12 Transition Strategy: From Insurgent to Institution

12.1 Historical Lessons

The 1935 split between AFL craft unionism and CIO industrial organizing, followed by their 1955 merger, illustrates tensions relevant to FOSS professionalization.[26] The CIO brought "youth and fervor," diverse membership, and willingness to accept radical organizers. The merger created institutional stability but arguably lost insurgent energy, shifting toward "business unionism."

FOSS projects face analogous tensions: needing paid staff and formal governance while fearing alienation of volunteer contributors who valued informal, mission-driven culture. The risk is becoming "service providers" rather than movements.

12.2 Maintaining Radical Roots

Strategies for maintaining founding spirit while building sustainable institutions:

- **Document founding values explicitly**: Write them into project charters, revisit them regularly, measure decisions against them
- **Design governance that protects original mission**: Constitutional provisions, supermajority requirements for fundamental changes
- **Continue recruiting "insurgent" voices**: Actively seek perspectives that challenge institutional comfort
- **Balance stability with disruption**: Create space for experimentation and challenge within stable structures
- **Maintain connection to user community**: Professionalization should not mean distancing from the people the project serves

Apache's 25-year track record demonstrates this is achievable—major projects like Kafka, Spark, and Hadoop have thrived despite competing corporate interests because governance explicitly separates funding from influence.

13 Conclusion: Community Over Code, Users Over Ideology

The deepest lesson from labor organizing for FOSS community building is also the simplest: **you cannot shortcut the relational work**. McAlevey's distinction between mobilizing (getting existing supporters to act) and organizing (expanding the base through relationship-building) exposes why many FOSS projects plateau or collapse. They mobilize existing enthusiasts without systematically converting skeptics or activating passive users.

For FreeMoCap Foundation, the research suggests these priorities:

[26] Social Welfare History Project, "AFL-CIO," Virginia Commonwealth University Libraries.

13.1 Immediate Actions

- Document current informal governance before formalizing
- Create explicit engagement pathways for users at all levels—not just developers
- Establish structure tests to assess community capacity
- Consider fiscal sponsorship through NumFOCUS or Software Freedom Conservancy rather than immediate independent incorporation
- Begin collecting user stories across domains for public narrative

13.2 Near-Term Development

- Build collective care practices into community rhythms: regular check-ins, rotating demanding tasks, celebrating wins
- Inoculate community members against predictable challenges including corporate co-optation attempts
- Develop comprehensive campaigns for institutional adoption combining internal champions, peer advocacy, and support infrastructure
- Create pathways for cross-pollination: artists into scientists, scientists into artists, users into contributors, contributors into evangelists

13.3 Longer-Term Strategy

- Design governance that explicitly includes user voice, not just developer voice
- Connect to adjacent movements (Right to Repair, digital sovereignty, research democratization, educational technology reform) for coalition strength
- Build the ladder of engagement with clear rungs from passive user to governance participant
- Develop funding models that don't compromise independence or create conflicts between community and sponsors

The labor movement's hardest-won wisdom is that sustainable power comes from organized relationships, not individual heroics. Every burned-out maintainer who quits, every user who drifts away feeling unconnected, every potential advocate who never learns about the project—these represent failures to build the collective structures that distribute load and sustain commitment.

The FreeMoCap Foundation has the opportunity to build those structures from the start—treating community development not as overhead distracting from "real work," but as the foundation that makes all the other work possible. The software exists to serve users. The foundation exists to serve the community. **Community over code. Users over ideology. Relationships over recruitment.**

Key Researchers and Resources

Labor Organizing

- **Jane McAlevey**: *No Shortcuts*, *A Collective Bargain*, *Raising Expectations (and Raising Hell)*
- **Kate Bronfenbrenner**: Cornell ILR School, research on comprehensive campaigns
- **Marshall Ganz**: Harvard Kennedy School, public narrative framework
- **Labor Notes**: Publication and training organization for rank-and-file activists
- **Kim Moody**: *On New Terrain*, analysis of contemporary labor

FOSS Governance and Sustainability

- **Nadia Eghbal**: *Working in Public*, *Roads and Bridges* report
- **Tidelift**: Annual maintainer surveys and sustainability research
- **SustainOSS**: Community and publications on open source sustainability
- **Apache Software Foundation**: Governance documentation, "The Apache Way"
- **GitHub Open Source Guides**: Practical guidance including maintainer balance

Commons and Cooperatives

- **Elinor Ostrom**: *Governing the Commons*, design principles
- **Yochai Benkler**: *The Wealth of Networks*, commons-based peer production
- **Nathan Schneider**: *Governable Spaces*, platform cooperativism
- **Trebor Scholz**: Platform Cooperativism Consortium
- **CoTech**: UK network of technology worker cooperatives

Movement Building and Digital Rights

- **Electronic Frontier Foundation**: Digital rights advocacy
- **Software Freedom Conservancy**: Fiscal sponsorship and FOSS advocacy
- **Right to Repair Coalition**: iFixit, Repair Association
- **European Open Source Academy**: Digital sovereignty initiatives

Case Studies

- **Blender Foundation**: User-centered FOSS community
- **Rust**: Governance transition from Core Team to Leadership Council
- **Python**: BDFL to Steering Council transition
- **Node.js/io.js**: Fork and reunification
- **Debian**: Long-running democratic governance

Bibliography

Apache Software Foundation. "Briefing: The Apache Way." https://www.apache.org/theapacheway/.

Apache Software Foundation. "A Primer on ASF Governance." https://www.apache.org/foundation/governance/.

Bronfenbrenner, Kate, and Robert Hickey. "Blueprint for Change: A National Assessment of Winning Union Organizing Strategies." In *Rebuilding Labor*, edited by Ruth Milkman and Kim Voss. Cornell University Press, 2003.

Chen, Cher Weixia, and Paul C. Gorski. "Burnout in Social Justice and Human Rights Activists: Symptoms, Causes and Implications." *Journal of Human Rights Practice* 7, no. 3 (2015): 366–390.

Current Affairs. "Jane McAlevey on How To Organize for Power." April 2019. https://www.currentaffairs.org/news/2019/04/jane-mcalevey-on-how-to-organize-for-power.

Eghbal, Nadia. *Working in Public: The Making and Maintenance of Open Source Software.* Stripe Press, 2020.

GitHub. "Maintaining Balance for Open Source Maintainers." https://opensource.guide/maintaining-balance-for-open-source-maintainers/.

iFixit. "Right to Repair Progress." https://www.ifixit.com/right-to-repair-progress.

McAlevey, Jane. *No Shortcuts: Organizing for Power in the New Gilded Age.* Oxford University Press, 2016.

Museum of Protest. "The Psychology of Activism and Movement Longevity." https://museumofprotest.org/guides/guide-the-psychology-of-activism-and-movement-longevity/.

Opensource.com. "How a fiscal sponsor can help your open source project grow." January 2019. https://opensource.com/article/19/1/fiscal-sponsors-open-source.

Ostrom, Elinor. *Governing the Commons: The Evolution of Institutions for Collective Action.* Cambridge University Press, 1990.

Oxford Saïd Business School. "Mondragon Case Study." 2018. https://www.sbs.ox.ac.uk/sites/default/files/2019-07/RBF%20EoM%202018%20Case%20Study%20-%20Mondragon.pdf.

Platform Cooperativism Consortium. https://platform.coop/.

Schneider, Nathan. *Governable Spaces: Democratic Design for Online Life.* University of California Press, 2024.

Socket. "The Unpaid Backbone of Open Source: Solo Maintainers Face Increasing Pressure." 2024. https://socket.dev/blog/the-unpaid-backbone-of-open-source.

SustainOSS. "The Principles of Governing Open Source Commons." https://sustainoss.pubpub.org/pub/jqngsp5u/.

The Repair Association. "Our Mission and History." https://www.repair.org/history.

Tidelift. "The 2023 State of the Open Source Maintainer Report." https://www.sonarsource.com/open-source-maintainer-survey-2023.pdf.