Open Source Is Not Just Code

Designing Communities That Actually Scale

Sinduri Guntupalli

SPEAKER

About Me

Sinduri Guntupalli

Sr Developer Programs Engineer,
OSPO team at Dynatrace

- 5.5 years in Drupal: developer, then Product Manager
- Volunteer organizer across Drupal Austria, Drupal Switzerland, and beyond
- Open Source Program Office team at Dynatrace
- Building OffOn.dev, a vendor-neutral community to sustain open source contributors and grow the maintainers of tomorrow

WHY THIS MATTERS

Open Source Runs Everything

Linux
Most servers, the cloud, and every Android phone.

Git
Version control behind nearly all software.

curl
Moves data inside cars, TVs, phones, and servers.

OpenSSL
Secures a huge share of web traffic.

Python
Powers data work and most of modern AI.

Kubernetes
Runs cloud infrastructure at scale.

Most of the open source projects are maintained by small teams, often unpaid volunteers.

PART 1

1

The Burden

Most projects don't fail on code. They fail under invisible weight.

CONTEXT

The Comfortable Myth

- The myth: success comes from code quality
- Just ship better code, and the rest follows
- **Mostly wrong**

REALITY

What Actually Kills Projects

- Unclear contribution paths
- Expectations never written down
- A few absorb all the pressure
- **Fragile, then it stalls**

REFRAME

A Socio-Technical System

- A **socio-technical system**
- Code and community are one system
- **Community design matters as much as code**

PROBLEM

The Maintainer Trap

- Work piles onto a few people
- The hero maintainer burns out
- **A single point of failure is a design flaw**

A Good Model: Rust

Topic-focused teams (compiler, language, libraries, tooling, infrastructure) each own their area independently. A leadership council coordinates across teams. Ownership is spread by design, so no single person carries the whole project.

PART 2

2

The Fellowship

The work is too much for one person. It has to be shared.

FRAMEWORK

Six Pillars That Decide Whether a Project Lasts

Governance
Who decides, and how. Make decision-making explicit before you need it.

Contributor Experience
How easy and rewarding it is to help. Lower the cost of showing up.

Recognition
Whether people's work is seen and valued. Most community labor is invisible.

Local Community
Where belonging actually forms. Global projects still need local homes.

Communication
Whether people can explain what it solves, and why. This is how a project gets known.

Funding and Sponsorship
Whether the work is paid for. Time is never really free.

PILLAR 1

Governance

- Decide how you will decide, **early**
- Put shared functions under a neutral body
- Spread maintainer roles across people
- Give conflict a clear path
- Pick a model: BDFL, council, or foundation. Big projects layer several
- Single-company control is the opposite: one owner, not a neutral body

A Good Model: Apache Software Foundation

The PMC elects committers on merit. Individuals represent themselves, not employers. Decisions happen in the open: if it wasn't recorded, it didn't happen.

PILLAR 1 · GOVERNANCE · CODE OF CONDUCT

A Code of Conduct

- Explicit rules, not vague good vibes
- Name what is **not acceptable**: harassment, discrimination, personal attacks, sustained disruption
- Rules only matter if they are enforced

A Good Model: Contributor Covenant

A widely adopted code of conduct. Clear expected behavior, clear unacceptable behavior, and a path to report and enforce.

PILLAR 2

Contributor Experience

- Lower the cost of a first contribution
- Give newcomers a clear place to start
- **Visible** path to trusted contributor
- If it is confusing, they quietly leave

A Good Model: Kubernetes

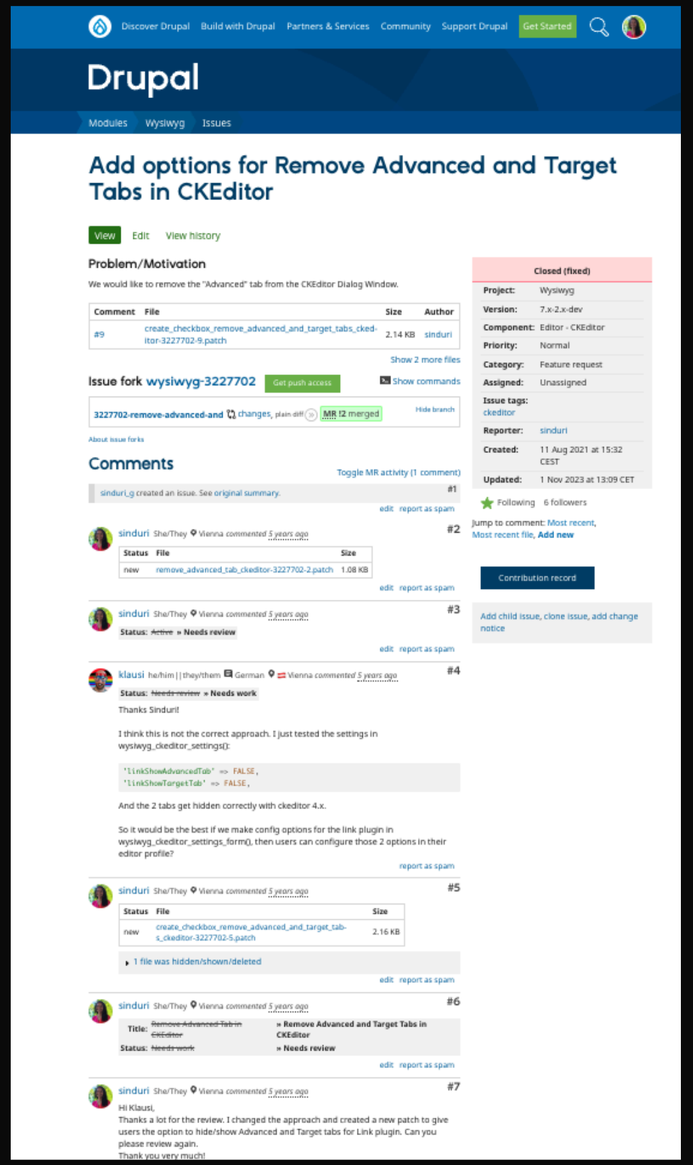
Published contributor ladder: from member to approver, with clear expectations at each step. Good first issues, mentoring cohorts, and a graceful way to step back.

PILLAR 2 · FROM EXPERIENCE

My First Contribution

- 2021, during Covid, new to the community
- **45 min** just to open the issue, scared of being judged
- Nobody judged. Community was welcoming

drupal.org/node/3227702



PILLAR 3

Recognition

- Most labor is **invisible**: review, triage, docs, mentoring
- Invisible work gets undervalued
- Record who did the work, sponsors included

A Good Model: Drupal Contribution Credit System

A public record of who did what, across code and the invisible work. Contribution becomes visible instead of assumed.

PILLAR 3

Why Recognition Changes Behavior

- Tie recognition to **real work**
- Companies get visible standing for what they fund
- It becomes routine, not a favor

When a company treats contribution as part of the actual work, with real time and budget behind it, the alignment becomes real and it lasts. Good incentive design beats good intentions every time.

PILLAR 4

Local Community

WHY IT MATTERS

- Global projects need **local belonging**
- Pair global events with local ones
- "Come for the code, stay for the community"

AT A LOCAL EVENT

- **Mentorship workshops** for first-timers
- **Contribution rooms** to work together
- **Student workshops**: Drupal in a Day

A Good Model: Drupal

DrupalCon globally, local volunteer-run camps, small and accessible by design. That's where new people get pulled in and where most new organizers learn the ropes.

PILLAR 5

Communication

- Open source projects need people who can **explain what problem it solves and why**
- Great code that nobody understands stays unused
- Especially important **early**, to make a project known
- Docs, tutorials, blogs, videos, talks, case studies

PILLAR 6

Funding and Sponsorship

- Time is not free. Someone pays, in money or unpaid hours
- Sustainable funding keeps maintainers from **burning out**
- Fund the boring, critical work, not just features
- Not a prerequisite. It matters once others **depend on you**

A Good Model: Django

The Django Software Foundation pays Fellows to do the unglamorous maintenance: triage, reviews, releases, and security. Routes to fund this: Open Collective, GitHub Sponsors, Patreon, Tidelift, and the Sovereign Tech Fund.

HOW TO TELL

Signals of a Healthy Community

- **Bus factor:** how many can leave before it stalls
- **Time to first response** on issues and pull requests
- **Retention:** do first-time contributors come back
- Is the **maintainer count** growing, flat, or shrinking

These tell you if the pillars are working, or just look good on paper.

PART 3

3

The Road Ahead

What to prioritize, what AI changes, and what you can actually do.

PRIORITIES

What to Do Now vs Later

OPTIMIZE EARLY

- Clear contribution paths
- Explicit, written expectations
- Distributing authority early

Cheap now. Expensive to fix later.

SAFE TO DELAY

- Heavy formal governance
- Complex tooling and automation
- Over-structuring small communities

Heavy process on a tiny project can strangle it before it grows.

NUANCE

Good Advice Needs the Right Partner

"Be welcoming to everyone"
Works when paired with triage, so maintainers can keep up.

"Move fast"
Works when paired with transparency, so contributors can follow the changes and not get blindsided.

"Document everything"
Works when paired with clear ownership, so the docs stay current.

Context decides whether a practice helps. The practices aren't the problem; the pairing is.

THE AI QUESTION

Help and Hazard

WHERE AI HELPS

- Lowers the barrier to a first contribution
- Speeds up docs, translation, and issue triage
- Could ease the "privilege of free time" open source depends on

WHERE AI STRAINS

- Low-effort contributions can overwhelm reviewers
- The cost shifts from author to maintainer
- Access is unequal. AI can become a new privilege.

AI doesn't replace community design. It raises the stakes. Share the cost and skill. Build it into contributor experience.

THE AI QUESTION · IN PRACTICE

When AI Becomes Slop

- Low-effort AI output lands as **real work** for maintainers
- Bogus AI security reports flood small volunteer teams
- **Seconds to generate, hours to debunk**
- The cost lands on the reviewer, not the author

Example: curl

curl was flooded with AI-generated security reports. By 2025 about 1 in 5 submissions was slop, and the genuine rate fell below 5%. The team shut down its paid bug bounty to stop the noise. The Python Software Foundation and Open Collective hit the same wall.

MENTAL MODEL

Thrive vs Decay

THRIVING PROJECTS

- Share decisions
- Make it easy to help
- Recognize the work
- Build local belonging
- Communicate the why
- Fund the work

DECAYING PROJECTS

- Concentrate decisions
- Leave contributors guessing
- Let invisible work go unseen
- Stay purely global
- Stay hard to understand
- Rely on unpaid time

The drift is slow and quiet. That's exactly why it's easy to miss.

YOUR TURN

What You Can Realistically Influence

You don't need to be a maintainer, or even a contributor. If you care about open source, any of these help.

- Document one unclear process
- Review a first contribution
- Credit invisible work out loud
- Help one local event
- Fund a project you rely on
- Share a project you use on social media

These are small structural acts. They compound.

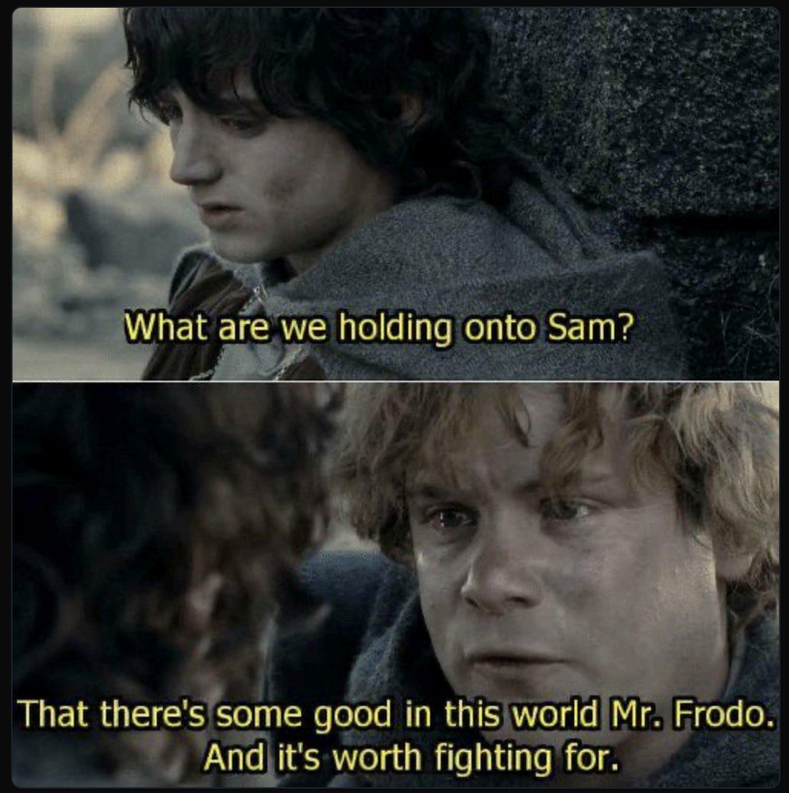
CLOSE

Open Source Is Not Just Code

- The system around the code is what scales
- **Design it on purpose**

REALITY CHECK

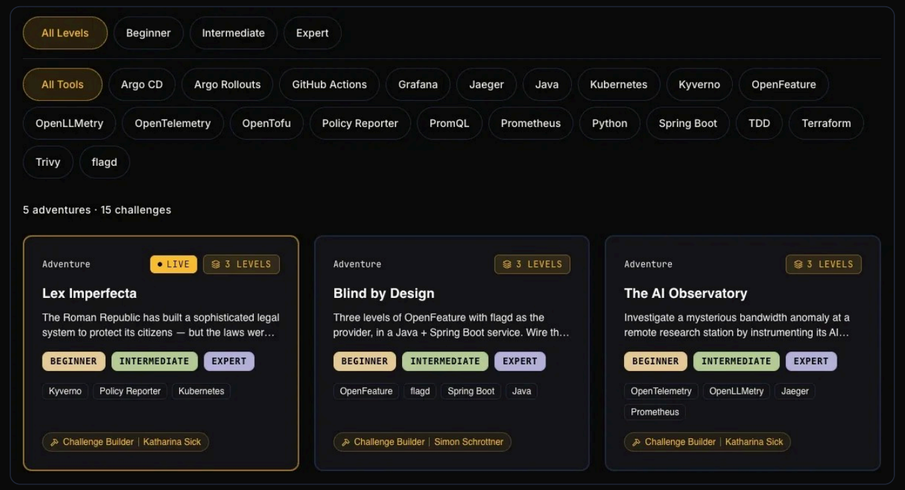
- No quick fix
- You control your corner, not the whole
- Changing defaults is slow, and that is normal
- **Start with one pillar**



COMMUNITY

Join OffOn.dev

- A welcoming space for learners, contributors, maintainers, writers, designers, translators, and organizers
- Built to sustain open source contributors and grow the maintainers of tomorrow
- **Play Challenges:** hands-on challenges around real open source tools, in GitHub Codespaces.
- **Open Source Talks:** for the Austrian open source community to present projects and connect with enthusiasts.



OffOn.dev

SOURCES

Sources & Further Reading

PRIMARY READING

- David Hirsch, Open Source Communities
- Dries Buytaert, The Privilege of AI in Open Source
- Chris Short, OSPO Notes: Open Source Governance

COMMUNITY EXAMPLES

- Rust, Governance and teams
- Apache, The Apache Way
- Kubernetes, Contributor ladder
- Django, Fellowship program

DRUPAL

- Contribution credit
- Mentoring and first-time contributor workshops
- Contribution days at events
- Drupal in a Day and Open University

MEDIA CREDITS

- The Lord of the Rings trilogy (2001–2003), New Line Cinema. GIFs via GIPHY: smW5FBep69d3q, RlZO2mKipB3u8, 8Xu2IkvLsjyLe, g67pETponv03e.

Thank you

Q&A

Leave Feedback

Slides & Resources