When Values Eat Their Young: How Ideal-Driven Groups Drift into Their Own Shadow

14 min read • 3,057 words

Themes: Consciousness Mental Health Programming Recursive Spiritual Contemplative

---

Picture this: A Slack channel for an open-source project that prominently displays "Be excellent to each other" in its community guidelines. Six months later, a maintainer posts a thoughtful critique of a proposed code of conduct change. Within hours, they're called "toxic," told they're "literally causing harm," and face demands for their removal. The pile-on continues for days.

Nobody seems to notice the irony.

This isn't some made-up scenario. This is real shit that happens. I've watched it unfold in projects I love, in companies I've worked for, in communities I've believed in.

> I've been in the Python community since 2008. I've seen us go from "be nice" to... something else. But this isn't just about tech. Watch any church split, any activist group implode, any company betray its founding principles. The pattern is universal.

1

The same algorithmic thinking that helps us debug code can help us understand why communities systematically fail their stated values. These aren't moral failures—they're predictable system dynamics that we can recognize, understand, and potentially prevent.

> This is programming as systems thinking: recognizing that human communities, like complex codebases, have emergent behaviors that can't be reduced to individual intentions. We debug by understanding the entire system, not just blaming individual actors.

The Paradox

The more a community cares about its values, the more vulnerable it becomes to betraying them. Not because the values are wrong. Not because the people are bad. But because human social dynamics create predictable failure modes that turn principles into their opposite.

This mirrors how algorithmic systems systematically consume human virtue—well-intentioned optimization targets inevitably corrupt the very thing they're meant to improve. The recursive loop applies to communities just as it applies to code: the systems we create to embody our values end up shaping new values, often contradicting our original intentions.

> "Every great cause begins as a movement, becomes a business, and eventually degenerates into a racket." — Eric Hoffer

Your community—whether it's an open source project, a startup, a church, a yoga studio, or an activist group—will face these pressures. The question is whether you'll build guardrails before you need them.

The Machinery of Inversion

Virtue Signaling: When Performance Replaces Practice

Virtue signaling gets a bad rap, often weaponized to dismiss legitimate moral concerns. But the phenomenon is real. In any community with strong values, there's social currency in being seen as embodying those values.

The problem? Performance gets rewarded more than practice.

Think about it like code coverage metrics. The goal is good: write tested, reliable code. But once coverage percentage becomes the metric that determines promotion, you get developers writing meaningless tests that check nothing but boost the numbers.

> I've literally seen tests that assert `true === true` just to hit coverage targets. Goodhart's Law in action: "When a measure becomes a target, it ceases to be a good measure."

The measure became the target, and the target became meaningless.

This is identical to how [engagement optimization destroys the very human connections it claims to foster](). When belonging becomes performative, when inclusion becomes a competitive sport, when kindness becomes a metric—the original values die in the measurement.

In value-driven communities, this looks like competitive displays of ideological purity. Increasingly elaborate pronoun signatures. Lengthy self-flagellating acknowledgments that crowd out actual work.

> When your project README is 90% virtue signaling and 10% documentation, you've lost the plot. Users need to know how to use your software, not your politics.

The person who speaks most loudly about inclusion might be the same one privately blacklisting colleagues for minor infractions. Performance becomes the measure of virtue, and like all metrics that become targets, it corrupts the thing it was meant to measure.

Purity Spirals: The Revolution Eating Its Children

A purity spiral begins innocently: someone raises the bar for what counts as living up to the community's values. Others, not wanting to appear less committed, raise it further. Soon, you're in a bidding war where yesterday's progressive position is today's problematic stance.

Consider what happened in Extinction Rebellion splinter groups between 2019-2021: It started with "reduce your carbon footprint." Then "go vegan." Then "don't fly." Then "don't have children." Eventually, members were being shamed for taking jobs that required commutes. Multiple XR chapters

hemorrhaged members, keeping only the most privileged who could afford such restrictions—ironically undermining their stated goal of building a mass movement.

> This pattern was documented in Andreas Malm's "How to Blow Up a Pipeline" (2021) where he critiques the purity spirals within climate movements. The French Revolution parallel remains apt: the Jacobins who sent others to the guillotine eventually found themselves condemned by even purer revolutionaries (see Simon Schama's "Citizens").

The Iron Law of Institutions

Political journalist Jonathan Schwarz articulated this in his Iron Law of Institutions (2007):

> "The people who control institutions care first and foremost about their power within the institution rather than the power of the institution itself."

Watch what happens when a community leader's position depends on there being problems to solve. Suddenly, problems become very hard to solve. The Code of Conduct committee that needs violations to justify their existence. The working group that creates more working groups. The board member more interested in their board seat than the community itself.

You know what's fun? Watching these same leaders give keynotes about empathy while actively ignoring maintainer burnout in their own projects.

This isn't conscious villainy. It's structural incentive. When your position in a community depends on fighting dragons, you'll always find dragons to fight—or create them.

> This is why I'm skeptical when any group claims they need "permanent" positions to address "systemic" issues. If the issue is truly systemic, why would your job exist to solve it? The incentives are backwards from the start.

The same way social media algorithms create outrage because outrage drives engagement, institutional positions create problems because problems justify positions. The system optimizes for its own perpetuation, not for solving the problems it claims to address.

Goal Displacement: When the Means Become the Ends

Every organization starts with a mission. Over time, the processes created to achieve that mission become the mission itself: the meeting about the meeting, the committee to oversee the committee, the elaborate consensus process that ensures nothing ever gets decided.

One collective I observed spent six months perfecting their decision-making process. During those six months, they accomplished exactly zero of their actual goals.

> Process matters, but when your process for deciding how to help people takes longer than actually helping them would have taken, you've lost the plot. Ship something. Help someone. Then iterate.

Cognitive Dissonance: The Stories We Tell Ourselves

When behavior and beliefs conflict, something has to give. Usually, it's the beliefs that bend. A community that prides itself on kindness but regularly engages in cruel pile-ons doesn't admit hypocrisy. Instead, it develops elaborate justifications: "This isn't cruel, it's accountability." "We're not being exclusive, we're maintaining safety." "It's not censorship, it's consequences."

The human brain is exceptionally good at resolving dissonance through narrative. The problem is, these narratives become the water we swim in—invisible, unquestionable, and ultimately destructive to the very values they claim to protect.

> Like poorly designed software architectures, narratives can become legacy systems that resist refactoring. Our mental models are not neutral—they're code we've been unconsciously writing our entire lives.

Case Studies in Contradiction

The Inclusive Space That Wasn't

The Tech Workers Coalition, founded in 2014 with values of "radical inclusion" and worker solidarity, experienced multiple schisms over ideological purity. By 2020, several chapters had split over whether members could work at certain companies, whether to allow managers, and what constituted "real" tech work.

> Similar patterns documented in Logic Magazine's "Tech Worker Organizing" issue (2020) and in Wendy Liu's "Abolish Silicon Valley" (2020) where she discusses the contradictions within tech activism movements.

A member asked, in good faith, about the practical implications of abolishing prisons for violent crime victims. They were immediately labeled as "perpetuating carceral violence," subjected to a days-long struggle session, and ultimately banned. The community's response to someone seeking education on their values was to exclude them for not already embodying those values perfectly.

The bitter irony: the banned member was a formerly incarcerated person trying to understand how abolition frameworks addressed their own complex experiences with violence.

The Kind Open-Source Project

JavaScript framework communities have faced these patterns repeatedly. Communities that pride themselves on being welcoming—with comprehensive Codes of Conduct, community teams, and regular appreciation posts—still see maintainers harassed over technical decisions.

The pattern is predictable: disagreement escalates, GitHub issues become battlegrounds, and when exhausted maintainers take mental health breaks, they're accused of "weaponizing mental health to avoid accountability."

Python, a language and community that prides itself on progressive values of inclusion and kindness, faced similar tensions during the "walrus operator" debates of 2018-2019. Despite the community's "be nice to each other" ethos and

extensive diversity initiatives, the discussion became so toxic that Guido van Rossum, Python's creator, stepped down as BDFL (Benevolent Dictator For Life). His resignation email (July 12, 2018) was devastating:

> "I don't ever want to have to fight so hard for a PEP and find that so many people despise my decisions... I'm basically giving myself a permanent vacation from being BDFL, and you all will be on your own... I'm tired, and need a very long break."

The kindness was performative. When tested by real conflict, the community defaulted to the same toxic behaviors they claimed to stand against, but wrapped in the language of justice and accountability.

> Multiple maintainers documented similar experiences. See Nolan Lawson's "What it feels like to be an open-source maintainer" (2017) and André Staltz's "Software below the poverty line" (2019). The same people who put mental health in their bios will destroy someone having a public breakdown.

This reveals the gap between stated values and lived reality—the recursive loop where communities shape the consciousness of their members, who then shape the communities, often in ways that contradict the original intentions.

> Think of this like a feedback loop in signal processing: each iteration slightly changes the signal. Our collective consciousness is constantly being recompiled, with each community interaction serving as a commit to the shared repository.

Personal Experience

I experienced this firsthand. I worked for a search API company that proudly championed neurodiversity—it was all over their hiring materials, their company values. They knew about my schizoaffective disorder when they hired me. They said they valued different types of minds.

Then I had an actual episode. Made an embarrassing, disturbing comment on GitHub during a manic phase triggered by a new medication—clearly unwell, clearly delusional. Within 24 hours, I was fired. No discussion of accommodation. No recognition that this was literally the neurodivergence they claimed to value. Just gone.

Neurodiversity was a recruiting tool, not a real commitment. As I've explored in [the cost of transparency](), companies love the performance of inclusion until they encounter the reality of what inclusion actually requires. The values were real in the marketing materials but absent when tested by actual human need.

Beyond Tech

This isn't just a tech problem. I attended a church for years that preached about community, love, and spiritual growth over worldly success. Then they let the pastor go—officially because he "wasn't focused on growth enough." He'd been focusing on deepening relationships, helping struggling members, building real community. But the board wanted bigger numbers, more tithing, a permanent building.

The irony? A significant portion of the congregation left in response to his firing. The growth-obsessed board destroyed the actual growth they had. They prioritized metrics over mission and lost both.

Why This Matters Anyway

Before you conclude that all value-driven communities are doomed to hypocrisy, let's acknowledge some crucial truths:

These communities exist for good reasons. The tech industry really does have inclusion problems. Open-source really can be hostile to newcomers. Traditional institutions really do perpetuate harm. The values these communities espouse aren't just nice-to-haves—they're attempting to address real, documented problems that cause real human suffering.

Many communities succeed. For every horror story, there are quiet success stories: communities that actually embody their values, resolve conflicts constructively, and create genuine change. These don't make headlines because "community functions as intended" isn't news.

> Some communities get it right. They recognize that mental health isn't a weapon or an excuse — it's a reality. They understand that "be kind" means being kind even when someone's struggling, not just when they're productive.

The alternative is worse. Communities without stated values don't avoid these problems—they just make them invisible. At least a hypocritical community can be called out on its hypocrisy. A community that never claimed to care about inclusion can exclude with impunity.

Structural forces are real. Social media algorithms reward outrage. Venture capital demands growth over health. Academic incentives favor publication over practice. These aren't excuses, but they are context that explains why even well-intentioned communities struggle against systemic forces designed to corrupt human virtue.

The Anti-Drift Checklist (For Humans)

Here's your practical playbook for keeping your community from eating its young:

Encourage disagreement. Make it someone's job to argue the other side—each meeting could have a designated devil's advocate role. Ask "what would change our mind?" before major decisions. Reward people who surface problems, not just those who maintain harmony. Healthy conflict prevents the kind of groupthink that leads to purity spirals.

Define kindness in behavior. "Kindness" means responding to crisis with support, not pile-ons. Track who gets second chances and who doesn't—this reveals your real values. Pay attention to who's leaving, not just who's joining. Actions speak louder than mission statements.

Value "good enough." Celebrate people who admit they were wrong. Ban comparative suffering competitions. Set maximum consequences—if everything is a firing offense, nothing is. Perfectionism is the enemy of progress and the friend of paralysis.

Reality-check regularly. Compare your stated values to your last three decisions. Ask newcomers what they actually observe. Would your community's founders survive your current rules? External perspective reveals internal blindness.

Design for bad days. Build systems that assume people will struggle—including leadership. Create paths back from mistakes, not just punishment. Plan for mental health crises; they will happen. Human complexity requires systemic compassion, not wishful thinking about perfect behavior.

Common Failure Modes

Watch how quickly noble values transform into their opposites when institutional pressures take hold:

Inclusion often becomes exclusion of those who don't speak the "right" language or perform belonging correctly.

Kindness transforms into cruel enforcement of kindness norms—destroying people in the name of protecting people.

Safety gets weaponized to shut down disagreement rather than protect genuine vulnerability.

Accountability becomes "accountability for thee but not for me"—rules that apply to everyone except those who make them.

Consensus devolves into minority veto power that creates paralysis rather than genuine agreement.

Transparency becomes performative theater that hides real decisions behind elaborate public processes.

Growth happens at the expense of founding principles—scaling the organization while shrinking the mission.

Re-aligning Means with Ends

The tragedy of values eating their young isn't that people are hypocrites. It's that good people with genuine commitments to important values can create systems that betray those values through entirely predictable social dynamics.

The solution isn't to abandon values or become cynical. It's to recognize these patterns as universal human tendencies and design systems that account for them. Just as we write tests because we know code will have bugs, we need to build guardrails because we know communities will drift.

This connects to the broader theme running through my work: how the systems we build to serve consciousness end up shaping consciousness. When we understand this recursive loop, we can design more consciously—creating communities that actually embody their values rather than just performing them.

> "The curious task of economics is to demonstrate to men how little they really know about what they imagine they can design." — Friedrich Hayek

The same humility applies to designing communities. We can't perfect human nature, but we can create structures that fail gracefully, correct course, and keep the gap between stated values and lived reality as small as possible.

Look.

I still believe in open source. I believe in communities built on values like kindness, inclusion, and collaboration. But good intentions aren't enough. You need systems. You need guardrails. You need to actually give a shit about people when they're at their worst, not just when they're shipping code.

This is what programming as spiritual practice really means—recognizing that every system we create either serves human flourishing or exploits human vulnerability. Community building is no different from building software: the quality of our intention matters as much as the quality of our implementation.

> We are always writing code—whether in text editors or social interactions. The question is whether that code amplifies human capability or constrains human potential. Our most important algorithms are how we treat each other.

I'm only able to write about these patterns because Sarah creates the conditions where this kind of reflection becomes possible—designing guardrails for my own worst days, reality-checking my blind spots, and building the kind of sustainable partnership that enables contemplative work rather than consuming it.

> The recursive loop applies here too: conscious relationships enable conscious work, which shapes collective consciousness. Sarah's insights about building systems that support people through struggle directly inform how I think about community design. The personal is the professional when you're debugging human systems.

Build the guardrails before you need them. Because once you've driven out the people who came to you for help—the ones who believed in your stated values, who were vulnerable enough to show their struggles—you can't un-eat them.

And they won't come back.

I didn't.

This exploration of institutional drift and values corruption provides context for understanding how communities systematically fail vulnerable members. Further reading: Jonathan Haidt's *The Righteous Mind (2012) on moral psychology and why good people divide, Eric Hoffer's The True Believer (1951) on mass movements and fanaticism, James C. Scott's Seeing Like a State (1998) on how well-intentioned schemes fail, Jerry Z. Muller's The Tyranny of Metrics (2018) on when measurement goes wrong, Albert O. Hirschman's Exit, Voice, and Loyalty (1970) on organizational decline and response, Jonathan Schwarz's The Iron Law of Institutions (2007) on power dynamics within organizations, Nolan Lawson's What it feels like to be an open-source maintainer (2017), Andreas Malm's How to Blow Up a Pipeline (2021) on purity spirals in climate movements, Project Aristotle findings from Google (2014) on psychological safety research, and Frances Haugen's Congressional Testimony (2021) on Facebook's internal culture.*