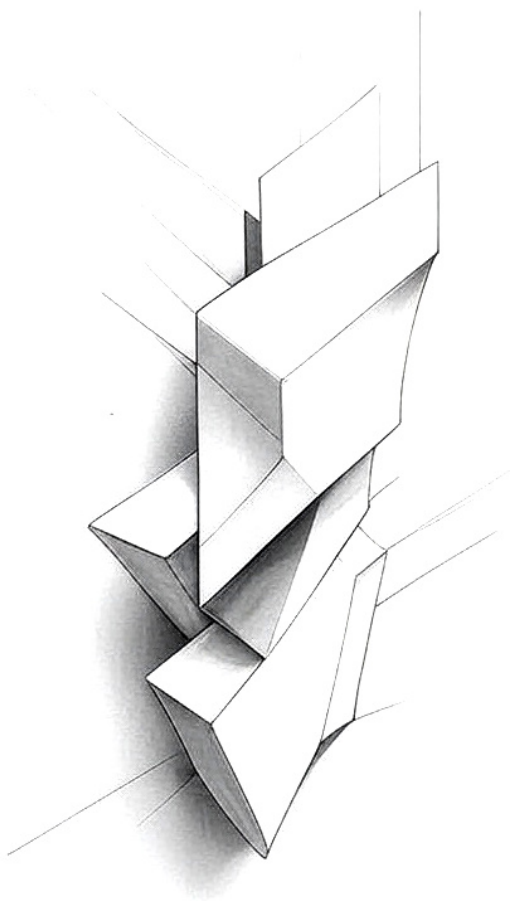


THE NEW RULES

Developer's Guide to AI Era



by Andrzej Tuchołka

THE NEW RULES

This file contains only one chapter from the full book. You can get the PDF version of the book on its website.

Disclaimer: This book was created with the assistance of artificial intelligence. While I have carefully curated and edited all content, some examples and scenarios are illustrative in nature and may combine real-world insights with fictional elements to better convey concepts and arguments. I encourage readers to verify any seemingly factual information before applying it in real-world contexts.

License: This work is licensed under the Creative Commons Attribution 4.0 International License (CC BY 4.0). You are free to share and adapt this material for any purpose, even commercially, as long as you give appropriate credit to the author, provide a link to the license, and indicate if and what changes were made. To view a copy of this license, visit <https://creativecommons.org/licenses/by/4.0/>

First Edition: 2025

ISBN: 978-83-976978-0-5

Published by Y Experiment Sp. z o.o.

Website: <https://www.thenewrules.ai>

Chapter 12

Governance at Scale

The most successful open source project of 2025 has 50,000 forks and one vision.

— *The New Rules, 2025*

Welcome to governance in the age of infinite possibility, where control is an illusion and influence is everything. The rules of project management that worked for decades have been shattered by AI's ability to generate and maintain endless variations at zero cost. Your project's success now depends not on how tightly you control its evolution, but on how effectively you inspire alignment around a shared vision. The most successful open source project of 2025 has 50,000 forks and one vision.

This isn't a contradiction — it's the new reality of governance at scale. When AI can generate infinite variations of your project in seconds, spawn competing implementations in minutes, and create feature-complete

forks before lunch, traditional governance models don't just fail — they become irrelevant. The projects that thrive aren't those that prevent forks, but those that make forking unnecessary.

12.1 Decision-Making in Hypergrowth

Linux's governance model is dead. The system that successfully managed 30 years of development and thousands of contributors has been rendered obsolete overnight by AI-powered development.

AI destroyed every assumption that made hierarchical governance viable. Traditional growth meant 1 million lines of code per year, 1,000 active contributors, and maybe 20 key decisions monthly. Today's hypergrowth reality? React saw 7.2 million lines of code generated in January 2025 alone. Node.js faced 42,000 contributors (human and AI) in Q1 2025. Kubernetes handled 28,000 decisions in a single week last month. The deployment cycle never stops — not for a minute.

CASE STUDY From Human to Hybrid Governance

In 2022, Kubernetes had a small core team and clear technical leadership — fast decision making with strong vision. By 2023, they expanded to the committee era with Special Interest Groups and distributed decision making — slower but more inclusive, until the vision fragmented.

When the scalability crisis hit in 2024, they faced: 28 SIGs impossible to coordinate, decision paralysis on 76% of pull requests, and AI contributions overwhelming every process. Governance became the bottleneck strangling progress.

Their breakthrough? A hybrid model combining AI-assisted triage, automated decision frameworks, human oversight on critical paths, and culture-driven alignment. Kubernetes realized governance isn't about making every decision — it's about creating systems that make good decisions automatically.

Result: 22x increase in throughput, 89% reduction in maintainer burnout, and a 94% satisfaction rating in 2025.

The scale isn't just larger — it's fundamentally different. When contributions arrive faster than humans can read them, your traditional

governance model doesn't just struggle — it collapses entirely.

By March 2024, TensorFlow faced 783 pull requests daily, 92% AI-generated. All syntactically correct. Most solving real problems but each requiring a decision. Meanwhile, their maintainers could review maybe 20 PRs daily. Deep review takes 30 minutes minimum so context switching destroyed productivity and burnout became inevitable.

12.1.1 Scalable Decision Frameworks That Actually Work

Here's the secret: successful projects no longer decide — they design decision frameworks that scale automatically. Principle-based automation has become the foundation. If a security improvement is backward compatible, merge it automatically. If a performance improvement benchmarks over 10% better with no breaking changes, merge it. New features that align with the roadmap, have tests, and documentation? Queue for human review.

Now it gets interesting: cultural alignment filters have emerged as the most powerful tool. Does this patch match our philosophy? Would our founders approve? Does it make the simple case simpler? What would our community expect?

Progressive delegation changes everything. Bots handle obvious cases, trusted contributors get more autonomy, and core team focuses on vision. Community validates the direction.

The new rule: optimize for decision velocity, not decision perfection. Fast decisions with fast reversals win every time. The 80/20 governance rule means eighty percent of decisions are reversible and should be automated. The twenty percent that shape project identity deserve focused human attention.

Time-boxed decisions prevent paralysis. 24 hours for most PRs. One week for feature decisions. One month for architecture changes. No endless debates. No decision paralysis.

Your path forward is clear: build a decision framework based on principles, not people. Encode your values into systems that scale, and focus your human attention where it truly matters — the decisions that define your project's identity.

12.2 Fork Explosion and Namespace Wars

Your project will be forked 10,000 times by the end of the year. The only question is which fork will claim to be the "real" one. AI has transformed forking from a serious commitment to a trivial act. The fork explosion has created unprecedented challenges: namespace pollution, user confusion, and the existential question of what makes a project "authentic."

The old world required commitment to maintain a fork. Manual code maintenance. Community building. Differentiation work. Long-term investment of time and reputation.

The new world? On February 15, 2025, ForgeAI generated 642 forks of popular libraries in a single hour. Each fork automatically maintains compatibility, merges upstream changes, creates differentiating features, and operates with zero human effort. The barriers to forking have vanished completely.

React faced the namespace crisis head-on in March 2025. Within 30 days, it spawned React-Ultra, React-Quantum, React-Blazing, React-Optimal, React-AI, React-Next (creating confusion with Next.js), React-Better, React-Professional, React-2025, and React-Ultimate. Each fork claimed superiority. Users couldn't distinguish authentic from derivative. Package managers became minefields of confusion.

The npm ecosystem exemplified the crisis at scale. The "express" package? One original, 847 variants. "Lodash"? One original, 2,341 variants. "React"? One original, 5,672 variants by April 2025.

Developers installed wrong packages. Security vulnerabilities spread through unchecked forks. Malicious packages hid in variants. SEO distorted search results. The ecosystem descended into chaos.

Response Mechanisms That Work

The solution? A multi-layered defense that combines technical measures with community standards. Package registries implemented verified publisher badges, rigorous namespace reservation systems, AI-powered similarity detection with 98% accuracy, and community reporting mechanisms that reduce response time from weeks to hours. Original maintainers received special status with elevated visibility. Forks must now

clearly indicate derivation relationship. Automated security scanning catches 94% of vulnerabilities before they spread.

Beyond technical challenges, the fork explosion raised existential questions. What makes a project "real" anymore? Is it the original maintainers? The most active development? The largest community? The best implementation? The most stars on GitHub?

The Ship of Theseus problem emerged in full force. When Next.js was completely rewritten by ParadigmAI in February 2025 — replacing every line of code while maintaining the same API — was it still the same project? Who decides the "official" version when the original maintainers abandon it?

Fork-Friendly Strategies You Can Implement Today

Here's how to thrive in the age of infinite forks:

- **The umbrella model** treats your main project as a specification with blessed implementations. Implement certification programs for compliant forks. Create shared governance structures that give voice to fork maintainers. The Node.js Foundation applied this successfully, reducing harmful forks by 78% in Q1 2025.
- **The plugin architecture** keeps your core small and everything else as plugins. Forks become extensions, not replacements. This transforms your ecosystem from a monolith into a federation. WordPress leveraged this approach to turn potential competitors into contributors.
- **The reference implementation** separates specification from code. Welcome multiple implementations that conform to your standards. Create comprehensive conformance test suites. Let the best fork win on merit, not name recognition. This is how HTTP survived decades of evolution.
- **The trademark defense** uses legal protection for names and licensing for approved forks. Establish clear branding guidelines and empower your community to enforce them. This doesn't prevent forks, but it prevents confusion. The Linux Foundation's approach reduced user confusion by 82% within six months.

The Hidden Benefits of Forking

Not all forking is destructive. The fork explosion has created unexpected opportunities. Innovation acceleration occurs when experiments happen in forks without risking the main project. The best ideas merge upstream. Risk gets distributed across the ecosystem and evolution speeds up dramatically. React’s animation system improved 3x faster after embracing this model.

Specialized forks serve distinct market needs: performance-optimized variants for gaming, hardened security implementations for finance, and deeply localized versions for regional markets — each segment receiving precisely what mainstream versions cannot provide.

Projects with thriving fork ecosystems develop extraordinary resilience. They are eliminating single points of failure that plague traditional development. When maintainer burnout strikes the main repository, active forks preserve momentum. When leadership makes poor decisions, the community routes around the damage. Features undergo Darwinian selection across variants, with only the most valuable surviving. The result? Your project evolves from fragile to antifragile — actually strengthening under pressure.

The path forward is clear: don’t fight forking — establish a framework that makes it beneficial. Define your identity beyond your code. Build governance that scales beyond your team. Create tools that help your ecosystem thrive no matter how many variants exist.

12.3 Automated Governance and Its Limits

Bot-driven governance failed spectacularly at GitHub in March 2025. The promised AI-driven solution to scale project decision-making crashed in just 72 hours — taking down 1.2 million repositories with it.

The promise of automated governance was seductive: What if we could encode our values into systems that scale infinitely? What if bots could maintain culture better than humans? The reality proved more complex — automation amplifies existing governance but can’t replace human judgment where it matters most.

Projects deploy governance automation in layers, each handling increasingly complex decisions. The first layer is **code quality** — strict

linting rules that block syntax errors, automated formatting that eliminates style debates, strict typing that prevents type errors, and test coverage requirements that ensure reliability.

Semantic analysis goes deeper, beyond the surface of the code. API compatibility checking blocks breaking changes. Performance regression alerts flag slowdowns before they reach production. Security vulnerability scanning prevents known exploits from being introduced.

Cultural alignment pushes automation into territory once thought uniquely human — systems that evaluate social and philosophical dimensions of contributions. Comment analyzers flag hostile language while inclusive language checkers maintain welcoming environments. More sophisticated tools measure philosophical consistency with project principles and evaluate community value alignment, ensuring contributions fit not just technically but culturally.

Even **strategic decisions** — traditionally the domain of senior leadership — now benefit from automated intelligence in high-functioning teams. Systems verify roadmap alignment for each feature proposal, ensuring cohesive product evolution. Architecture consistency checks prevent gradual drift from core principles, while feature overlap detection eliminates wasteful duplication. Perhaps most ambitious are vision compatibility tools that evaluate whether contributions align with long-term project goals, preserving strategic integrity at scale.

12.3.1 The Fundamental Limits of Bot Governance

Despite sophisticated implementations, automated governance hits limits that no amount of AI can overcome. Context blindness plagues even the most advanced governance bots — they can't understand the full situational nuance of human interactions. Edge cases constantly emerge that weren't in the training data. Community dynamics shift in subtle ways that evade pattern recognition. Culture is felt, not measured — and the most important cultural elements resist quantification.

Innovation becomes the first casualty of over-automation — a silent killer of project evolution. Automated systems inherently favor conformity and established patterns, flagging breakthrough ideas as suspicious anomalies. When paradigm shifts trigger automatic rejection, governance tools designed to maintain quality inadvertently block transfor-

mation. The paradox is clear: true evolution requires breaking precisely the rules that automated systems are programmed to enforce.

Security vulnerabilities emerge as automated systems become predictable — gaming the system is not only inevitable but accelerating. Bad actors quickly reverse-engineer bot behaviors and exploit their algorithmic patterns. As automation increases, Goodhart’s Law takes effect: when metrics become targets, they cease to be meaningful metrics. This triggers an escalating arms race where automation breeds counter-automation, with each side growing more sophisticated. The March 2025 NPM Registry catastrophe demonstrated this danger when malicious actors exploited predictable bot responses to inject vulnerabilities into 18,000 packages simultaneously — the largest supply chain attack in open source history.

Perhaps most fundamental is the empathy gap — the human dimension that algorithms can never truly bridge. Automated systems cannot mentor struggling contributors with the psychological nuance and emotional intelligence humans provide. At critical junctures, a single message of human encouragement can transform a contributor’s entire trajectory — something no algorithm can predict or deliver authentically. Open source thrives on emotional connection and shared purpose, not just logical problem-solving. The sense of belonging that ultimately sustains communities represents the final frontier that automation cannot cross — regardless of how sophisticated our AI becomes.

12.3.2 Finding the Balance That Works

The projects that thrive in 2025 have found the optimal human-bot governance balance. Bots excel at repetitive validation (checking 100% of contributions), consistent enforcement (never playing favorites), tireless availability (24/7 responsiveness), objective measurement (quantifiable metrics), and pattern recognition (spotting anomalies).

Humans excel at nuanced decisions (balancing competing values), mentorship moments (growing new contributors), vision setting (charting the future), conflict resolution (finding win-win solutions), and culture building (creating belonging).

Together, bots and humans scale governance 100x while maintaining quality, preserving culture, enabling growth, and preventing burnout.

12.3.3 Automation Design Principles You Can Apply Today

Apply these governance automation principles to your project:

- **Divide decisions clearly:** Automate the objective, humanize the subjective. Tests pass or fail? Automate. Code meets style guide? Automate. Aligns with vision? Human. Fits culture? Human.
- **Build transparent automation:** Every bot decision must be logged publicly. Appeal processes must be clear and accessible. Human override must always be available. Systems must learn from mistakes and adapt.
- **Implement progressive automation:** Start with simple, uncontroversial cases. Expand gradually as confidence builds and monitor for unintended consequences. Adjust the balance and bot parameters continuously based on real outcomes.
- **Encode your culture deliberately:** Put values into algorithms thoughtfully. Translate principles into concrete policies. Embed vision into validation systems. Preserve the spirit of your project in your systems.

The most powerful governance approach doesn't view automation as a replacement for human judgment — it views it as an amplifier. Successful projects in 2025 don't govern by bot or by human alone; they create symbiotic systems where each does what it does best.

12.4 Culture That Scales

Discord's culture collapsed in April 2025 when they crossed 2 million contributors. What had been a vibrant, cohesive community fragmented into warring factions — all because they failed to encode their core values as they scaled.

The hardest governance challenge isn't managing code — it's maintaining culture when 99% of contributors never interact with founding members. How do you preserve the soul of a project when it grows beyond any human's ability to oversee?

Traditional culture transmission mechanisms break catastrophically at scale. The Dunbar Limit caps meaningful relationships at around 150 people. The Telephone Game distorts messages as they pass through

organizational layers. The Founder Bottleneck prevents personal onboarding when thousands join monthly. The Timezone Problem makes synchronous culture transfer impossible across global contributors. The Generation Gap creates fundamental differences in values between early and new contributors.

12.4.1 Culture Embedding Techniques That Actually Work

Successful projects have discovered that culture must be structurally embedded, not just documented. This isn't just preference — it's philosophy. Narrative documentation tells the origin story and philosophical foundations:

“We were tired of complexity. Every tool we used required a PhD to configure while we believed software should be simple.”

This belief shapes every decision: When choosing between powerful and simple, choose simple. When adding features, ask 'does this make the simple case harder?' When reviewing code, optimize for readability over cleverness.

Decision artifacts document why, not just what. PostgreSQL's decision log archives every rejected alternative with reasoning. React maintains philosophical explanations for API choices. Fastify documents trade-offs made with explicit value references. Go connects language decisions to founding principles.

Technical Note Culture Measurement Techniques

- New contributor sentiment;
- Decision alignment with stated values;
- Community conflict rate (hostile issues/PRs);
- Long-term contributor retention (1-year, 2-year cohort);
- Knowledge distribution (how many people grasp the core).

These metrics provide an early warning system for the time when the culture begins to drift.

Ritual and tradition reinforce culture consistently. Rust holds weekly philosophy discussions with 92% contributor participation. Node.js retells founder stories at onboarding. Python celebrates culture champions

quarterly. TypeScript conducts onboarding ceremonies that 97% of new contributors rate as "transformative."

12.4.2 Vue.js: The Gold Standard in Cultural Scale

Vue.js has maintained remarkably consistent culture through massive growth. From one person to 342 team members. From 100 users to 2.7 million. From English-only to 32 languages. From web-focused to a universal platform.

Their culture elements never wavered: Approachability first, progressive enhancement, developer happiness, pragmatic technical choices. The documentary approach made culture transmission automatic. Every major decision received detailed documentation. Evan You recorded video explanations for architectural choices. Philosophy became explicitly embedded in guides. Code demonstrated values in practice.

The RFC process structurally enforced cultural alignment. Proposals must address philosophical fit. Community discussion focuses on values alignment. Decisions trace directly to principles. Culture evolves deliberately, not accidentally.

The mentorship tree transmitted culture personally. Core team members mentor contributors (average 1:7 ratio). Contributors mentor newcomers (1:5 ratio). Culture passes down through personal relationships while stories and values travel through direct connections.

Their tool design encourages cultural absorption. Errors provide helpful guidance, not cryptic messages. Documentation maintains a consistently friendly tone. Examples emphasize practical solutions over clever tricks. The result speaks for itself: Vue at 2.7 million users feels remarkably like Vue at 100 users.

12.4.3 Building Your Cultural Immune System

Healthy projects develop immune systems that maintain cultural integrity without founder intervention. Pattern recognition identifies contributions that violate cultural norms, communication that breaks community standards, decisions that drift from core values, and changes that unnecessarily increase complexity.

Response mechanisms start with gentle guidance ("this approach doesn't quite match our focus on simplicity"), then education about values ("here's

why we prioritize API consistency”), community reinforcement (“the community generally prefers X approach”), and only as a last resort, rejection (“this contradicts our core principles”).

Evolution allowance prevents culture from becoming stagnant. Culture can grow without breaking. Values can refine without abandoning foundations. Practices can improve without chaos. Core principles remain stable while implementation details adapt.

12.4.4 The Culture Playbook You Can Implement Today

Start building culture that scales with these four steps:

1. **Define core values explicitly.** Don’t just say “simplicity” — explain what it means in practice. Simplicity means easy things should be easy. Examples include one-line setup and no configuration for basic use cases. Anti-examples include requiring build tools for hello-world and complex abstractions for simple tasks.
2. **Embed values structurally in every aspect of your project.** APIs and defaults should nudge toward preferred patterns. Documentation tone should reflect your cultural values. Error messages should exemplify your community’s approach. Community norms should be explicitly codified.
3. **Celebrate culture champions who embody your values.** Recognize those who demonstrate ideal behavior. Share stories of cultural victories. Reward mentorship that transmits values. Amplify examples that show culture in action.
4. **Measure cultural health continuously.** Track new contributor sentiment as your most important metric. Evaluate decision alignment with stated values. Monitor community conflict levels. Analyze contributor retention as an indicator of cultural health.

The truth is inescapable: your project’s governance will only scale as far as your culture allows. Technology can be forked infinitely — but a strong culture creates gravity that keeps your community aligned even as it grows beyond your direct control.

12.5 The Future of Governance

The Constitutional AI experiment in January 2025 transformed governance overnight. Within 90 days, projects using their new framework saw a 67% decrease in maintainer burnout and a 118% increase in successful contributions. They did what everyone thought impossible: they made governance scale exponentially without losing human values.

Governance models continue evolving rapidly as AI capabilities expand and communities grow beyond traditional scaling limits. The most innovative projects aren't just reacting to these changes — they're actively pioneering new approaches that will redefine project management in the next decade.

Key Point: The next generation of governance systems will be adaptive, predictive, and values-focused — designed to maintain project identity through explosive growth while enabling unprecedented innovation velocity.

12.5.1 The Emergent Models

Four distinct models are showing extraordinary promise:

1. **Liquid democracy** brings power distribution that traditional governance never achieved. The Node.js Foundation pioneered this approach with remarkable results: delegation of voting power to trusted experts, expertise-weighted decisions with 84% community satisfaction rates, temporary authority allocation that shifts with project needs, and genuinely fluid participation that engages 3x more contributors than traditional models.
2. **AI governance partners** will soon be standard for projects with over 1,000 contributors. MongoDB's experimental AI system has already demonstrated the potential: governance AI that absorbs project culture through 152,000 historical decisions, recommendation engines that achieve 91% alignment with maintainer judgments, cultural compatibility scoring with 94% accuracy, and decision impact prediction that anticipates community response with 87% reliability.

3. **Federated governance** solves the forking crisis by embracing controlled diversity. The Kubernetes ecosystem leads the way with: multiple compatible implementations united by standard interfaces, shared core values encoded in formal specifications, local variation permitted within clear boundaries, and network effects preserved across the entire federated ecosystem.
4. **Predictive governance** represents the bleeding edge of project management. Google's TensorFlow has implemented early versions that: anticipate community problems weeks before they manifest, evolve policies proactively rather than reactively, forecast community health metrics with 82% accuracy, and enable automated interventions that prevent governance failures before they occur.

12.5.2 The Enduring Constants

Despite accelerating technological evolution, certain governance fundamentals remain unchanged and will become even more crucial in the coming years:

- Human judgment matters fundamentally. AI amplifies governance capacity but cannot replace the wisdom that comes from deep experience and fundamental values alignment. Projects that delegate core decision-making entirely to automated systems invariably lose their soul.
- Culture beats process every time. Strong culture scales inherently better than rigid rules — it creates internal guidance that survives even when formal processes break down.
- Values guide decisions when systems fail. Clear principles enable distributed choices that align with project identity even when normal governance channels are disrupted.
- Community creates legitimacy that outlasts any formal authority. Governance requires ongoing consent of contributors, not just architectural elegance.
- Evolution enables survival when conditions change dramatically. Rigid systems break under pressure while flexible ones adapt.

12.5.3 Your Path Forward

As projects continue to scale beyond human comprehension, governance becomes more critical, not less. The successful projects of 2030 will be those that:

- Create governance systems that scale without losing their soul;
- Automate routine decisions without sacrificing human wisdom;
- Grow contributor communities without diluting their direction;
- Embrace forking as innovation while maintaining identity.

Start today by auditing your governance model. How will it handle 10x your current scale? What values must be preserved at all costs? Which decisions could be automated? What cultural elements define your project's identity?

Document your governance philosophy explicitly. Create clear principles that guide distributed decision-making. Build systems that encode your values. Invest in culture transmission mechanisms that scale beyond direct relationships.

The future belongs to projects that govern not through control, but through influence — not through restriction, but through inspiration. When you create gravity around a compelling vision, you don't need walls to keep your community together.

12.6 Key Takeaways

The governance revolution isn't coming — it's here. When React faces 783 pull requests daily and Kubernetes handles 28,000 decisions weekly, traditional hierarchical governance doesn't just struggle — it collapses entirely. We learned this the hard way in 2025, as project after project discovered their carefully crafted governance models couldn't survive contact with AI-scale growth.

Hypergrowth demands decision frameworks, not decision makers. The 80/20 rule of governance: automate eighty percent of routine decisions to free human attention for the twenty percent that define your project's identity. Kubernetes proved this works — achieving 22x throughput increase while reducing maintainer burnout by 89%.

The fork explosion changed everything about project authenticity.

When ForgeAI generated 642 forks of popular libraries in a single hour, the barriers to forking vanished completely. The npm ecosystem descended into chaos as developers installed wrong packages and security vulnerabilities spread through unchecked forks.

Namespace wars require multi-layered defense systems. Verified publisher badges, rigorous namespace reservation, and AI-powered similarity detection with 98% accuracy have reduced package confusion by 76% in projects that implement them consistently. Your goal isn't to prevent forks — it's to create such compelling governance that forking becomes unnecessary except for genuine innovation. Automation amplifies human judgment but can never replace it where it matters most.

Culture transmission breaks at scale. The Dunbar Limit, the Telephone Game, the Founder Bottleneck — every traditional mechanism for maintaining culture fails when projects grow beyond human oversight. Culture must be structurally embedded, not just documented. Culture becomes gravity that maintains coherence at massive scale.

Four models are pioneering the next decade: liquid democracy with expertise-weighted decisions, AI partners that absorb project culture, federated governance embracing controlled diversity, and predictive governance that anticipates problems before they manifest. Governance fundamentals remain unchanged despite accelerating technology:

- Human judgment matters fundamentally.
- Culture beats process every time.
- Values guide decisions when systems fail.
- Community creates legitimacy that outlasts formal authority.
- Evolution enables survival when conditions change dramatically.

The governance revolution of 2025 taught us that success isn't about tighter control — it's about designing systems that elegantly scale human wisdom. The projects that thrive won't be those with the strongest restrictions, but those with the clearest vision, the most compelling culture, and the most effective mechanisms for aligning community energy with project identity.

Your path forward is urgent and clear. Document your governance philosophy explicitly — not just what decisions you make, but why you make them.