

# FREEDM SYSTEMS CENTER

## The ANDES Journey: Ten Years of Open-Source Innovation in Power Systems

Hantao Cui  
Associate Professor

Electrical and Computer Engineering  
North Carolina State University

2025 INFORMS Annual Meeting, Atlanta, Georgia

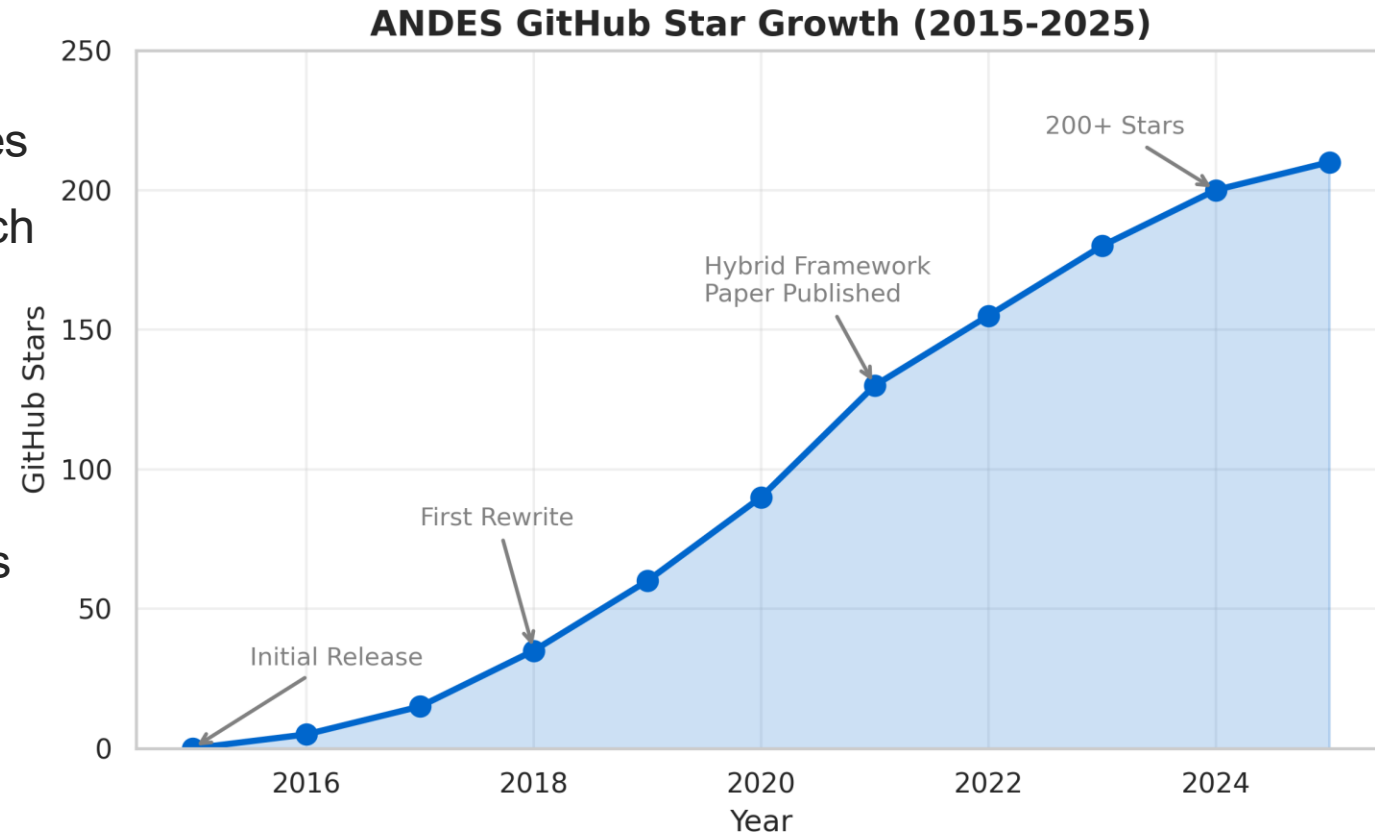


- **What is ANDES?**

- Open-source power system dynamics simulator (Python)
- Validates grid stability under disturbances
- Complements optimization tools (dispatch -> stability)

- **Growth over 10 years:**

- 200+ GitHub stars, 150+ citations
- Used by 30+ institutions in 10+ countries
- Active community across academia and industry



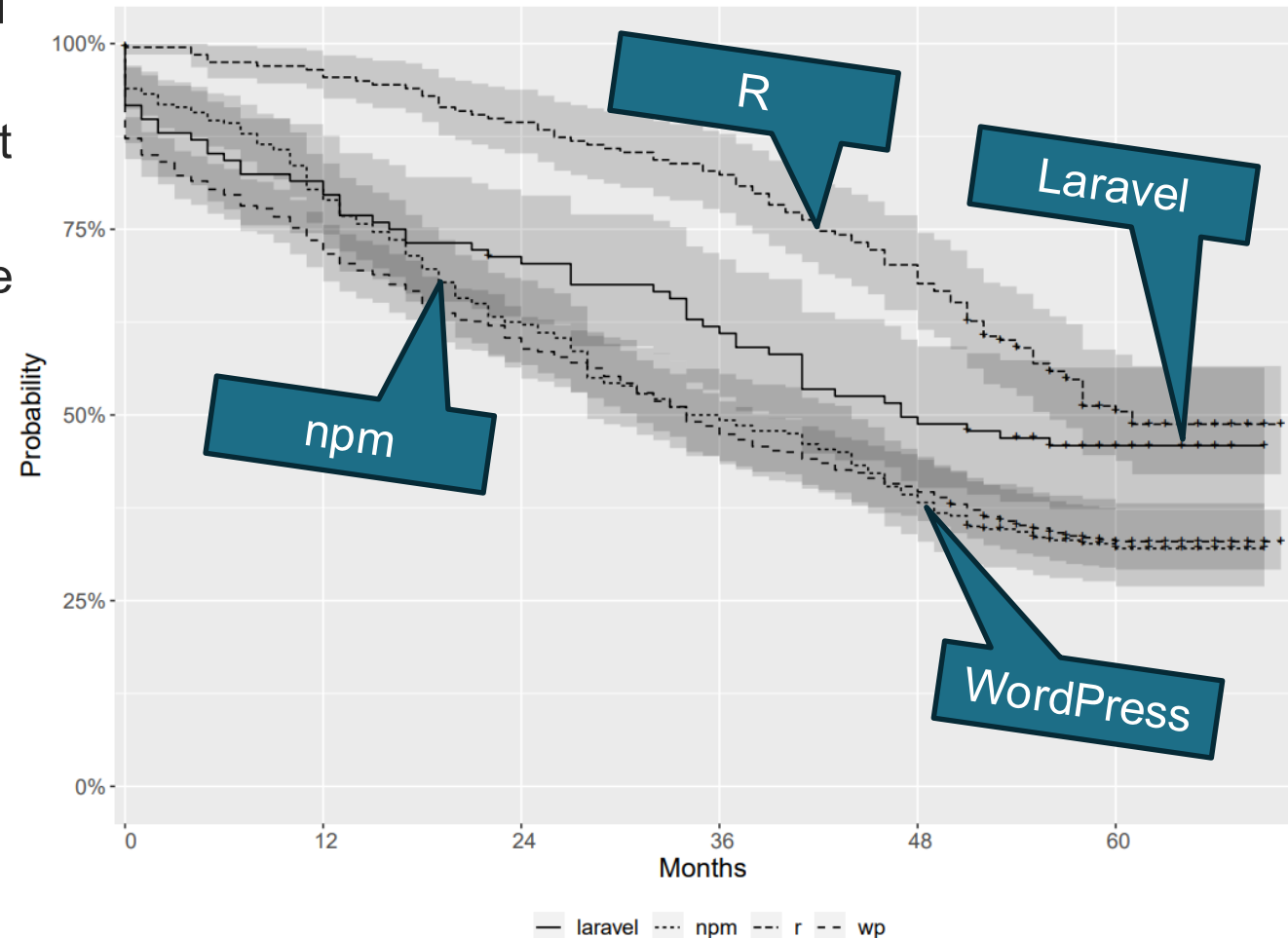
- **The Sustainability Crisis**

- 60% of specialized OSS are abandoned within 5 years
- Maintenance costs exceed development 3-5x
- Funding for innovation, not maintenance

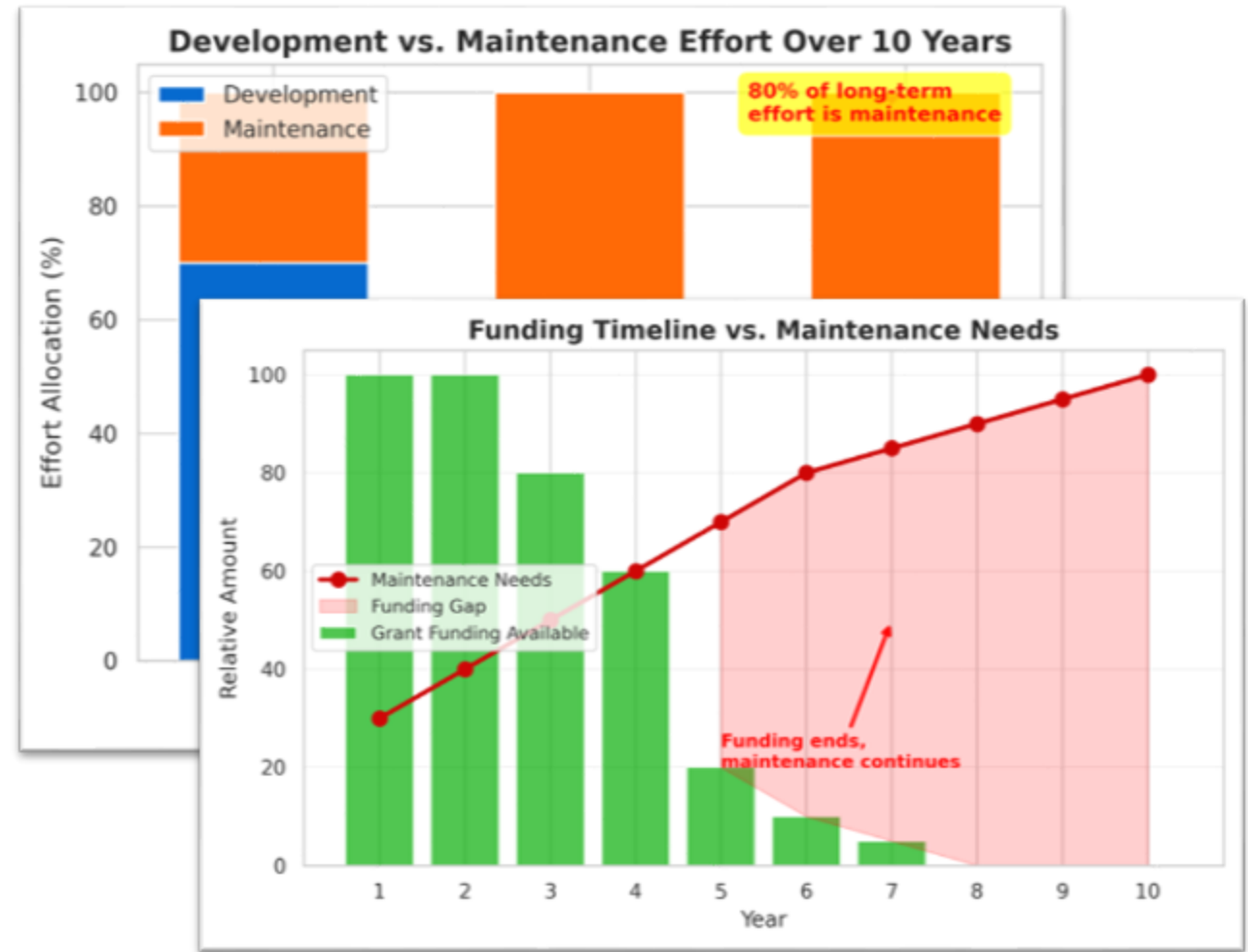
- **Unique Challenges in Technical Domains**

- Small user base vs. web development
- Deep expertise: power + software + math
- Academic incentives favor novelty over maintenance

- **This is a challenge across all specialized domains**



- **Effort allocation over 10 years:**
  - Years 1-3: 70% dev, 30% maintenance (NSF funded)
  - Years 4-6: 50/50 split (funding dries up)
  - Years 7-10: 20% dev, 80% maintenance (unfunded)
- **What maintenance means:**
  - Bug fixes, security patches, dependency updates
  - User support, documentation, PR reviews
- **The OR problem:**
  - How to sustain long-term infrastructure?
  - 80% of effort, 0% of dedicated funding



*Typical pattern of efforts and funding gaps in open-source software*

- **Code contributor data (2015-2025):**

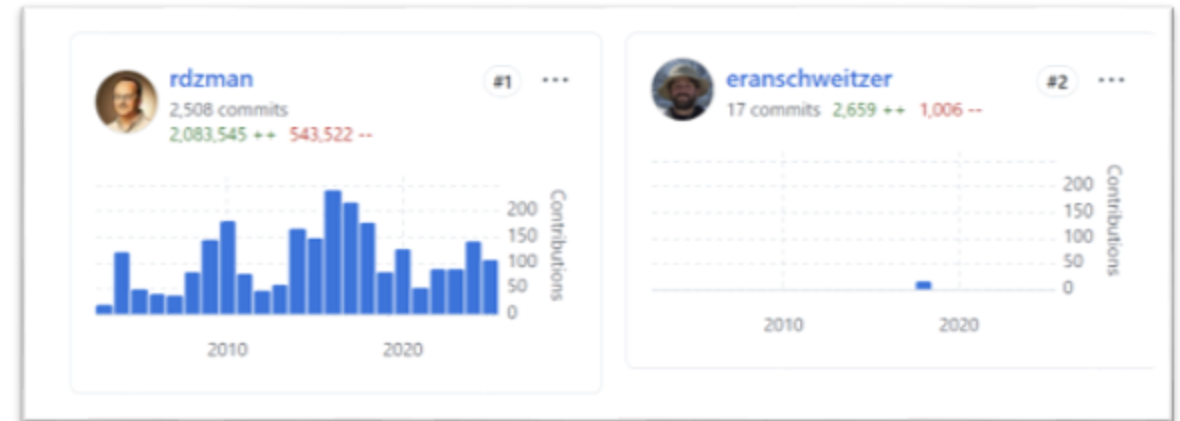
- # of contributors is in the 10s
- The vast majority of code written by a single contributor, the key developer
- 85% were grad students (churn after graduation)

- **What does not work:**

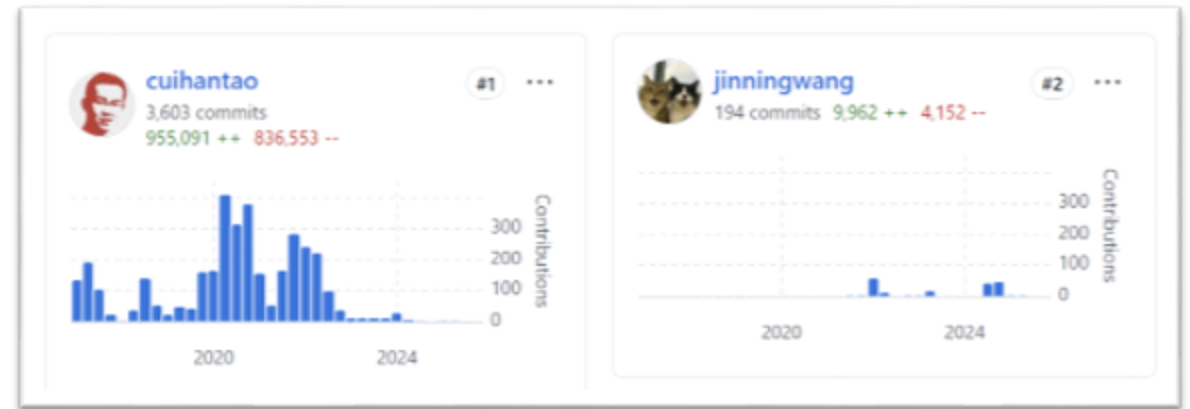
- Relying on volunteer enthusiasm (burns out)
- Academic credit alone (insufficient)

- **What does work:**

- Compensation: grants, part-time positions
- Skin in the game: using ANDES for research
- Clear onboarding, recognition



*Top contributors of MATPOWER*



*Top contributors of ANDES*

- **Model structure:**

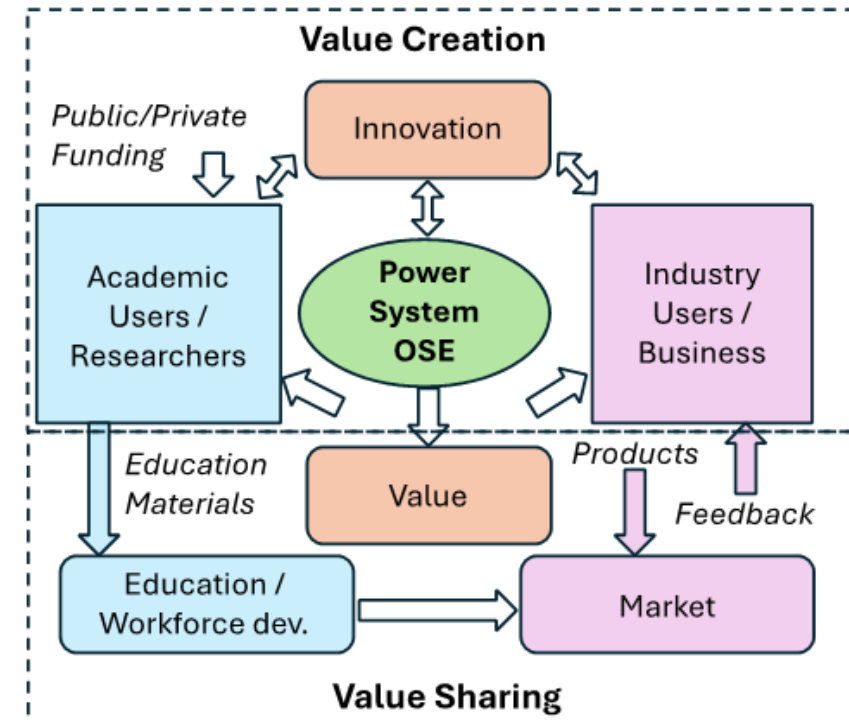
- Open core: APIs, standard models (GPL/MIT)
- Proprietary: Custom features, enterprise tools
- Dual license: Free for research, paid for commercial
- Services: Consulting, training, support

- **Value flows:**

- Academia: contributes algorithms, gets free access
- Industry: pays for reliability, gets guarantees
- Business: maintains core, docs, community

	Sustainability	Market Reach	Innovation Speed	Revenue Potential
Pure OSS (Volunteer)	Low	High	High	Low
Pure Commercial (Closed Source)	High	Low	Low	High
Hybrid (Open Core)	High	High	High	Medium

*Business model comparison for specialized OSS*

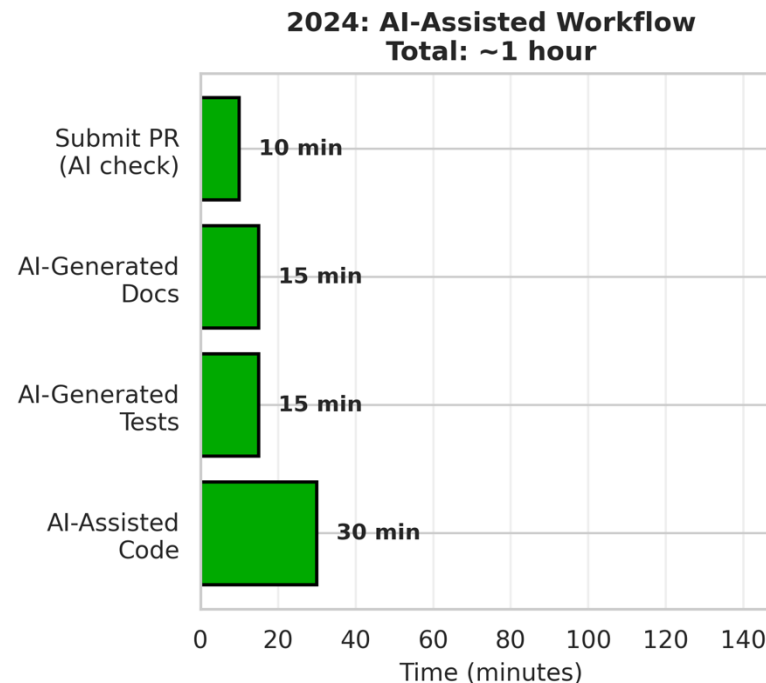
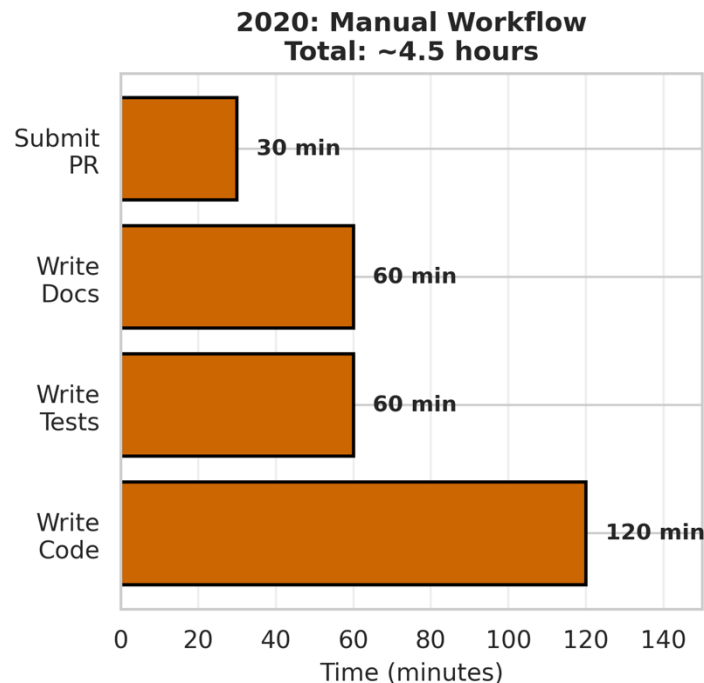


- **Barrier reduction:**

- Code generation speeds tasks 3-5x
- Lower entry barriers for new contributors
- Faster research prototyping (hours vs. days)

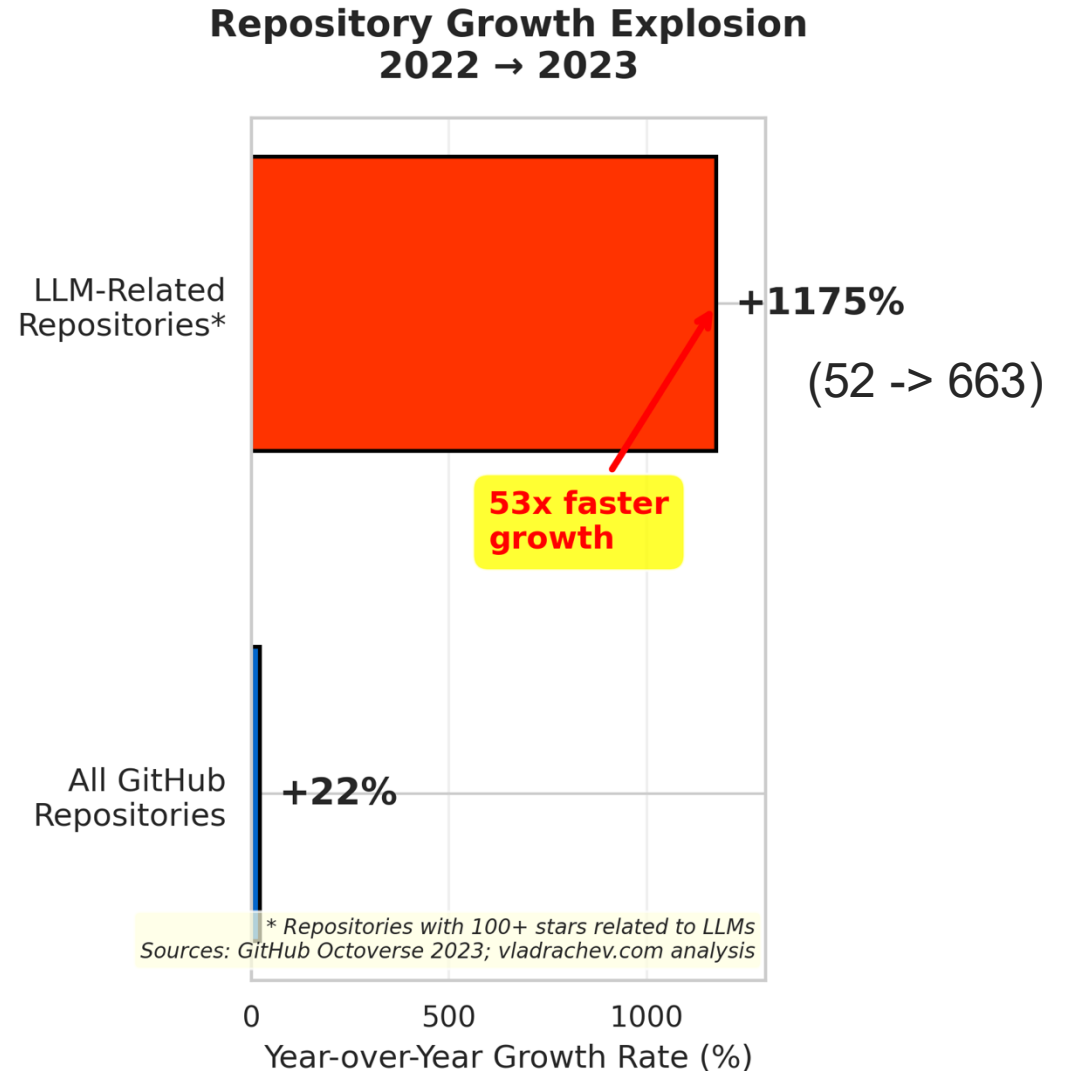
- **Quality improvements:**

- AI-assisted linting, test generation
- Better documentation (auto-generated)
- Code review assistance



*Charts are approximate and for illustration only*

- **The proliferation problem:**
  - Active maintainers up only 5% (general repos)
  - Easy to start, hard to sustain
  - Attention fragments across projects
- **Quality signal weakening:**
  - AI generates competent, not excellent code
  - "Good enough" floods the ecosystem
  - Harder to identify skilled contributors
- **Human bottlenecks remain:**
  - Design, architecture, community management



<https://coolest-gadgets.com/github-statistics/>

<https://vladrachev.com/notes/2024/repo-analysis/>

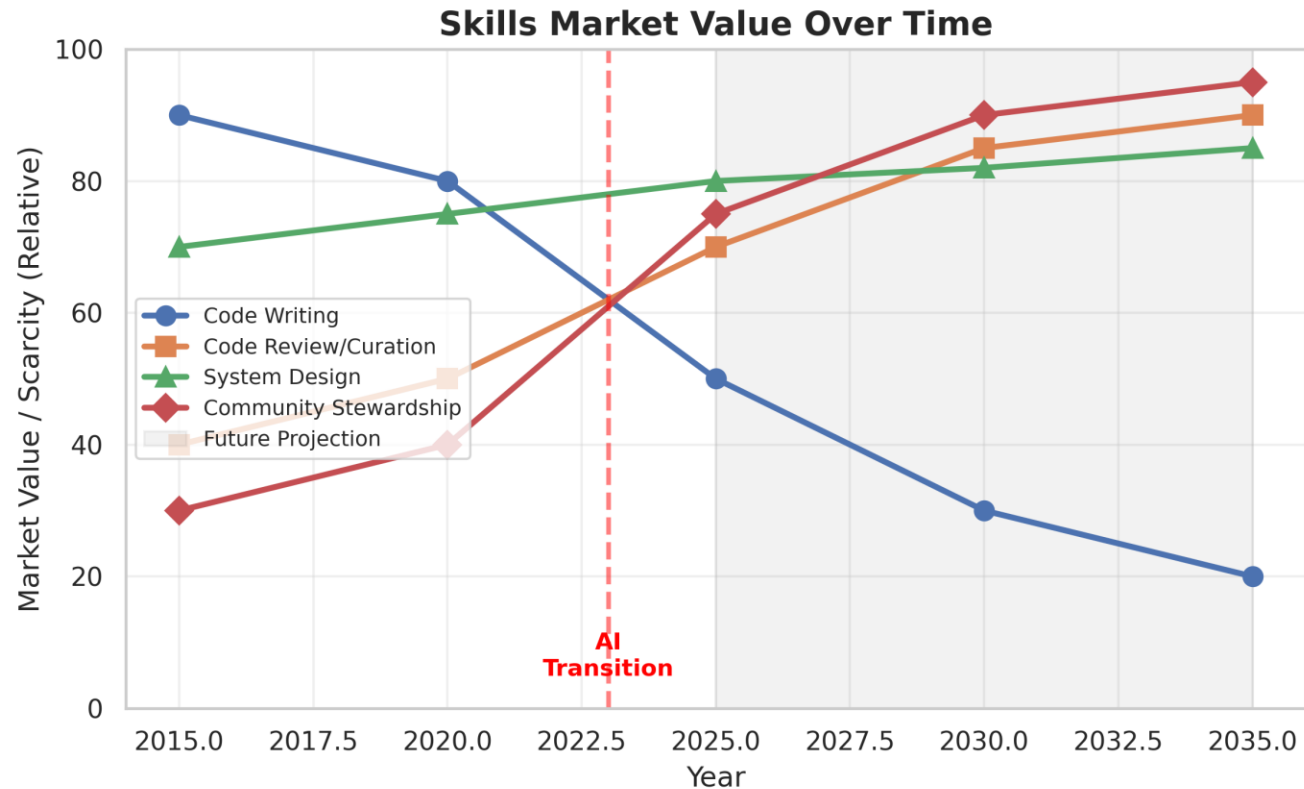


- **What AI changes:**

- Writing code: increasingly automated
- Reading/understanding: still requires humans
- Designing systems: AI assists, humans decide
- Maintaining communities: fundamentally human

- **The new scarcity:**

- Scarce in 2015: Python programmers
- Scarce in 2025: Curators and stewards
- AI generates options; humans choose wisely



**"In the age of AI, the scarce resource isn't code; it's curation."**

*Charts are approximate and for illustration only*

- **Business model:**
  - Hybrid (open core + proprietary)
  - Dual licensing, services revenue
- **Governance:**
  - BDFL -> stewards as project matures
  - Copyright clarity, contributor agreements
- **Community:**
  - Lower barriers, recognize non-code work
  - Pay maintainers when possible
- **AI era:**
  - Embrace amplification, curate aggressively
  - Measure engagement, not just volume

**Sustainability Model Selection Framework**

