Open Source and Inflection Points
Balancing Innovation with Stability

Dave McAllister

Sr. OSS Technologist

NGINX

So Why Worry?

- Definition of Open Source
  - "Open refers to a process that generates trust, permitting positive interdependence"
- Importance of Inflection Points
- The Challenge: Innovation vs. Stability

Innovation versus Stability

- Nature of Open Source
  - Innovation, based on collaboration and flow of ideas
  - Stability is often an afterthought
- Community
  - Innovation: continual improvements, features and directions
  - Stability: Dynamism can lead to issues
- Enterprise
  - Innovation: Prioritize the features that help them
  - Stability: Must be 110% reliable

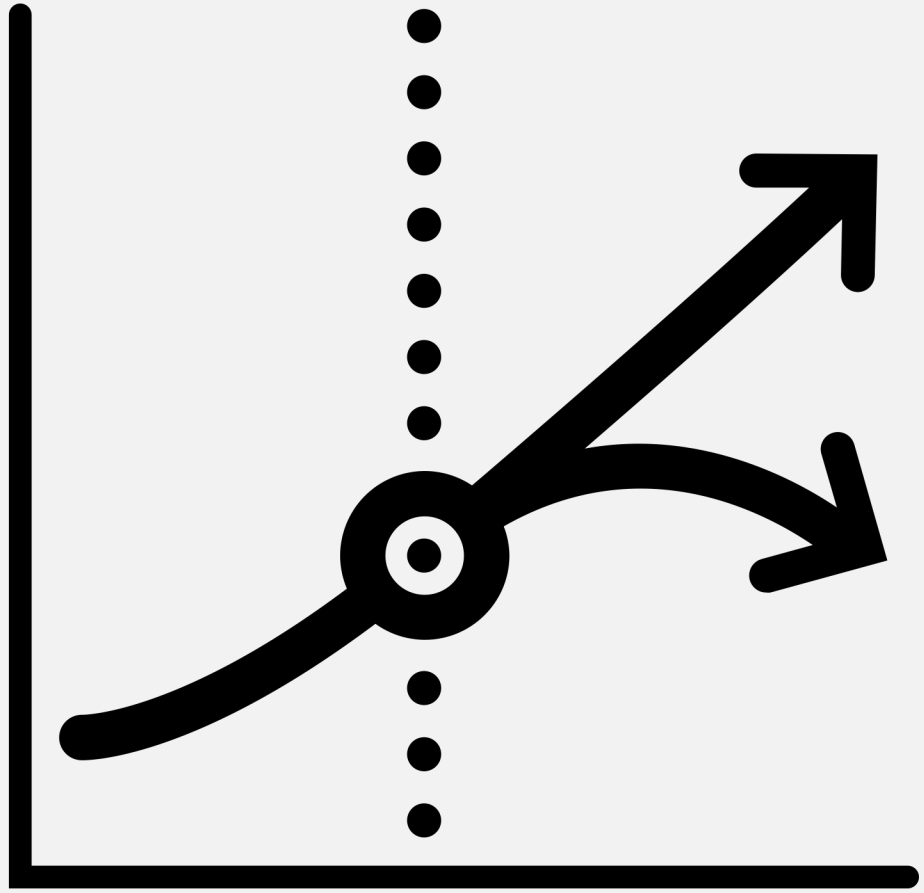
Balance can be tricky

Innovation without Stability: Leads to cutting-edge features but may introduce bugs, security vulnerabilities, or incompatibilities.

Stability without Innovation: While the software might be reliable, it may stagnate, missing out on critical updates, new features, or optimizations that the community or industry moves towards.

Inflection Point? Huh?

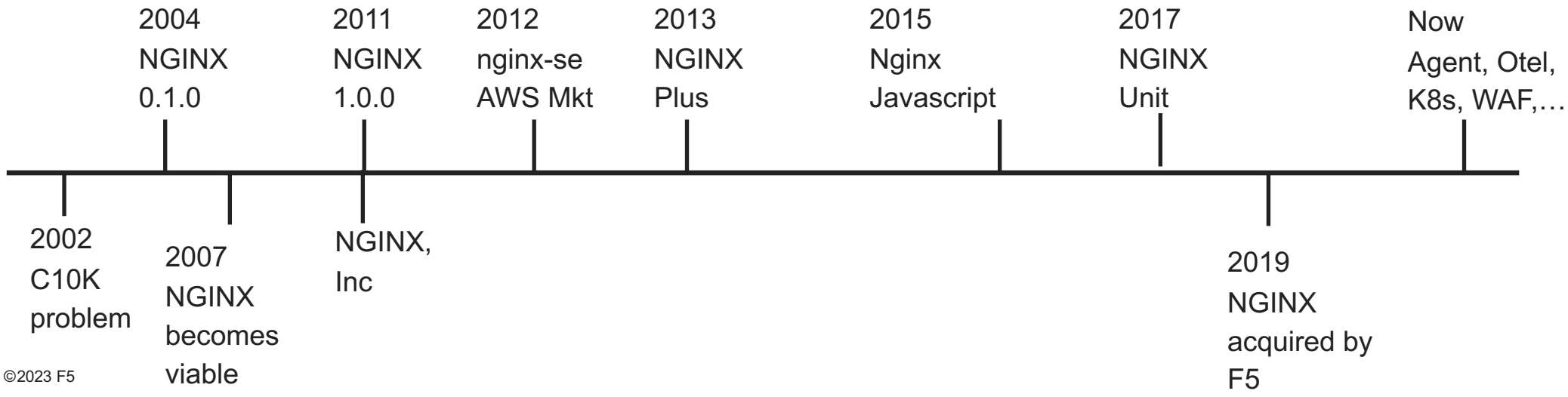
- Critical junctures in a project's lifecycle
- Influences on trajectory & technical direction
- Examples:
  - Initial releases
  - Governance decisions
  - Licenses
  - Business models



Critical Junctions

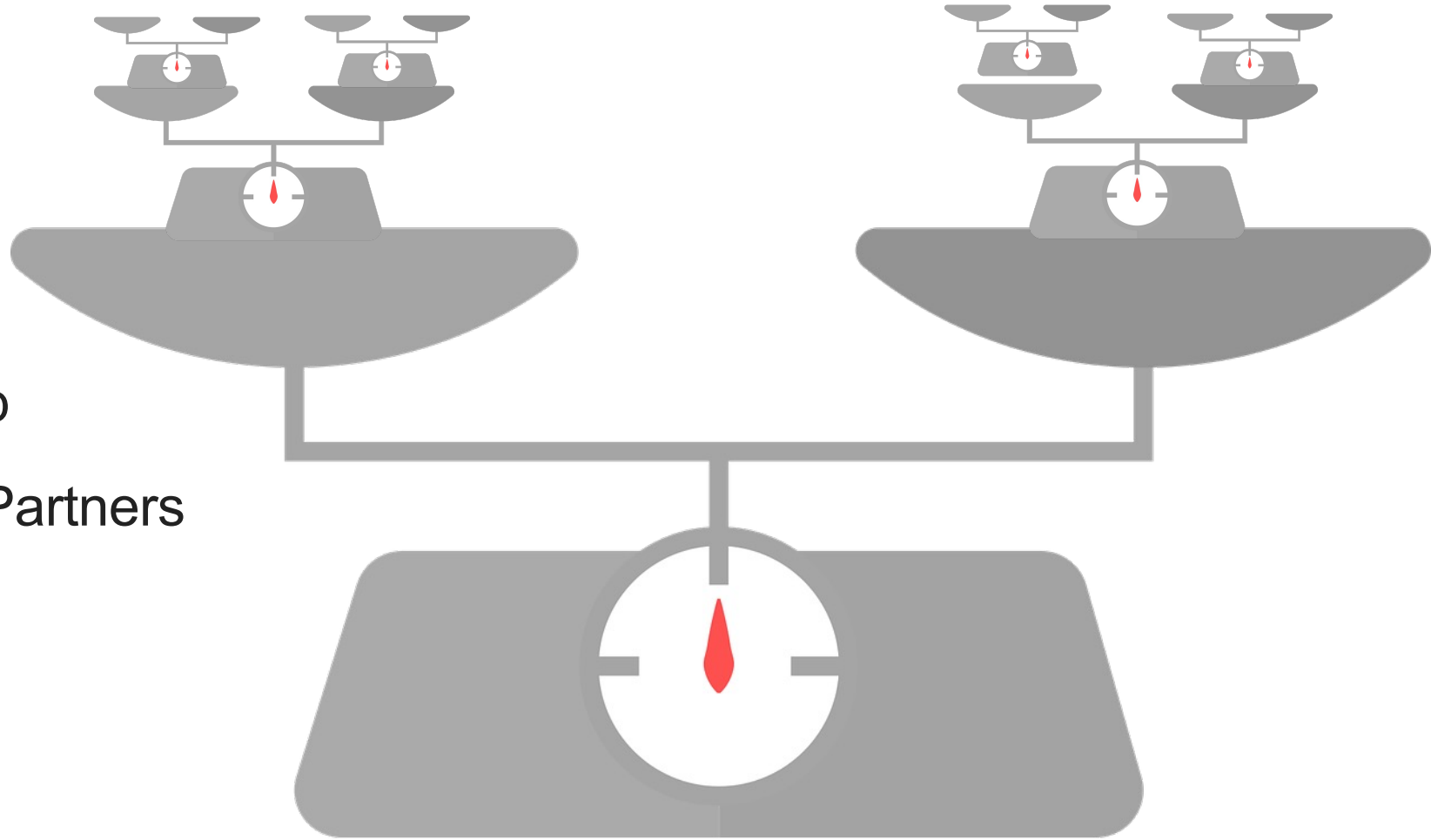
- Initial Release
- Stable Release
- Adoption
- LTS model
- Fiscal model/ commercial offering

- Partnerships
- Major releases
- Acquisitions
- Governance
- External factors



Inflections on Trajectory

- Community Engagement
- Market Needs and Trends
- Governance and Leadership
- External Collaboration and Partners
- Regulatory and Licensing
- Iteration and Feedback
- Financial Stability

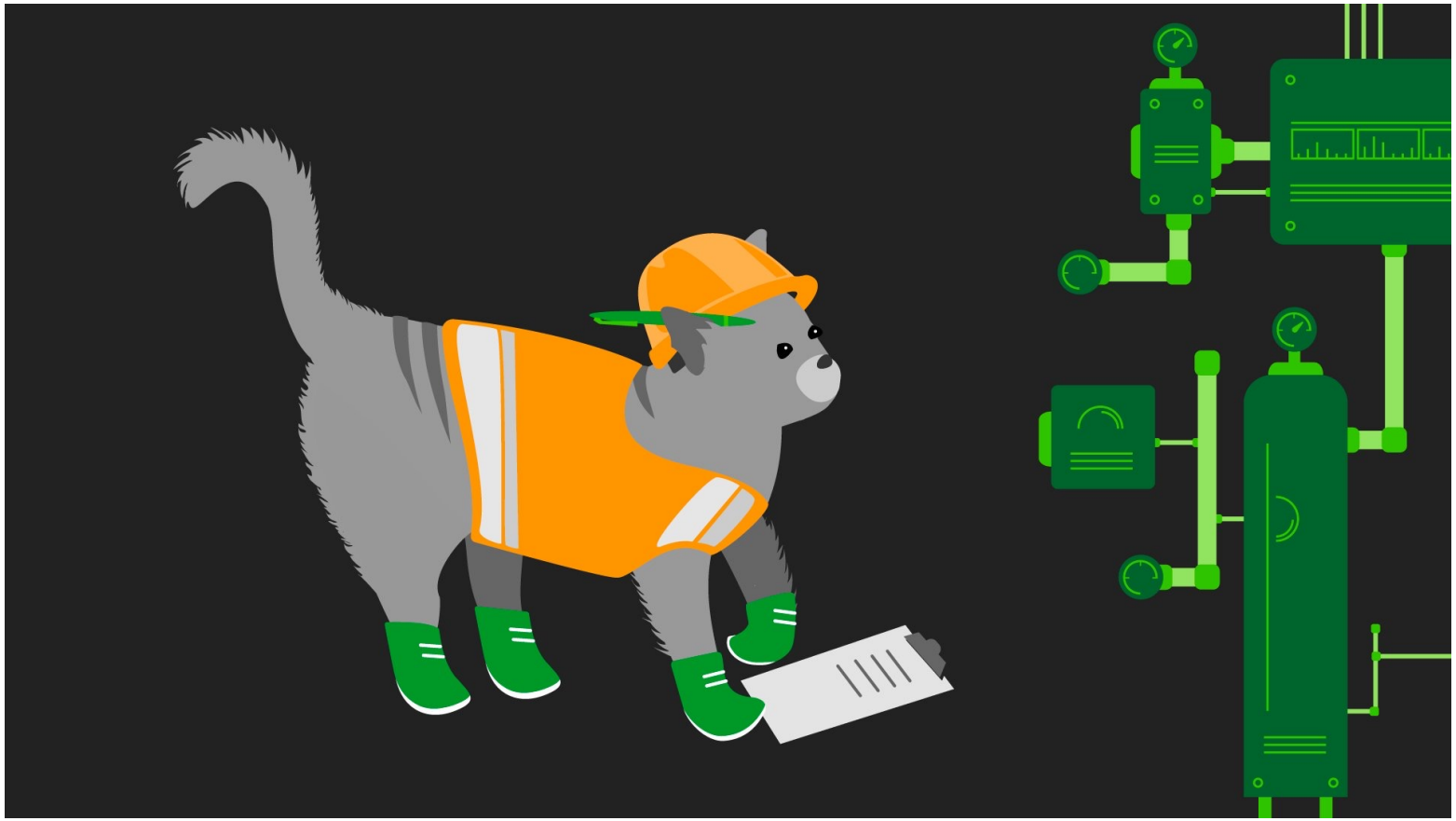


Innovation rocks!

- Drives project evolution
- Enhances relevance & adaptability
- Encourages community engagement

Community and Innovation

- Opportunity for Contribution
- Learning and new directions
- Problem solving, together
- Recognition and expansion
- Dynamic feedback
- Adaptive evolution
- Networking and cross disciplines

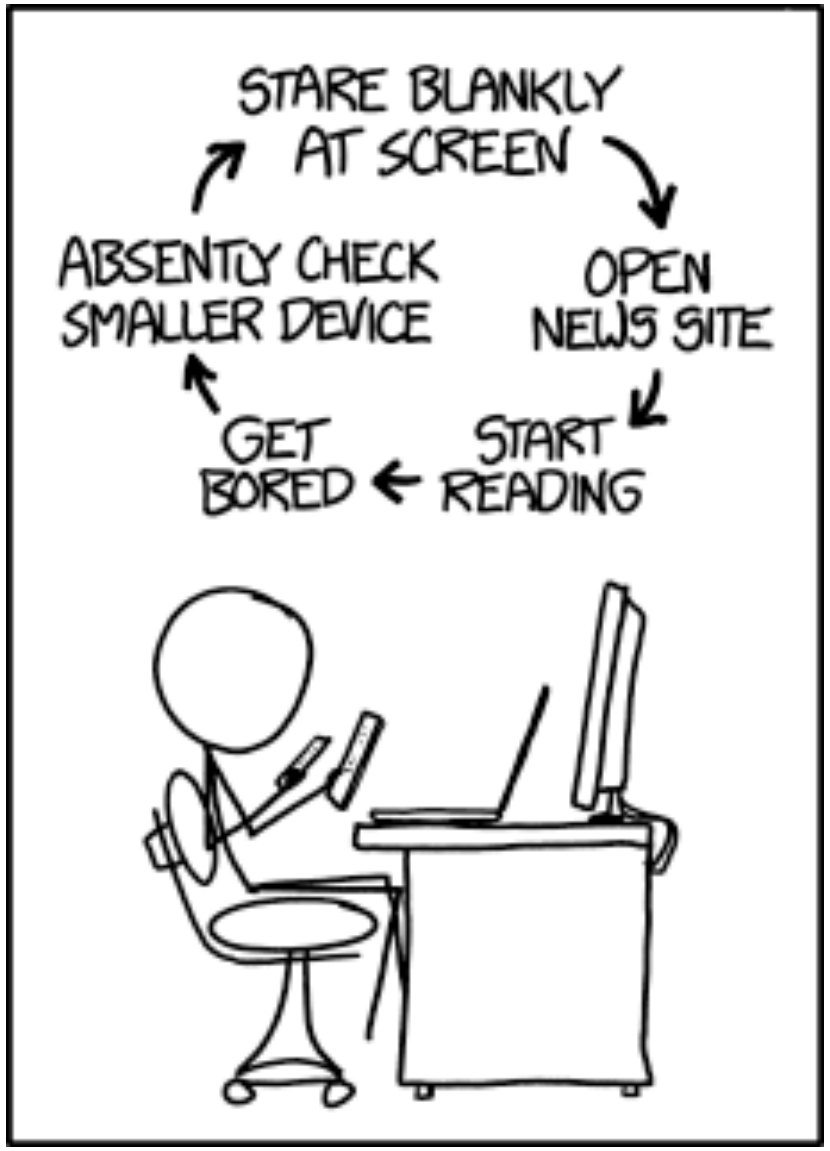


Stability Rules

- Ensures project reliability
- Builds trust for large-scale implementations
- Reduces operational risks

Walking the tightrope

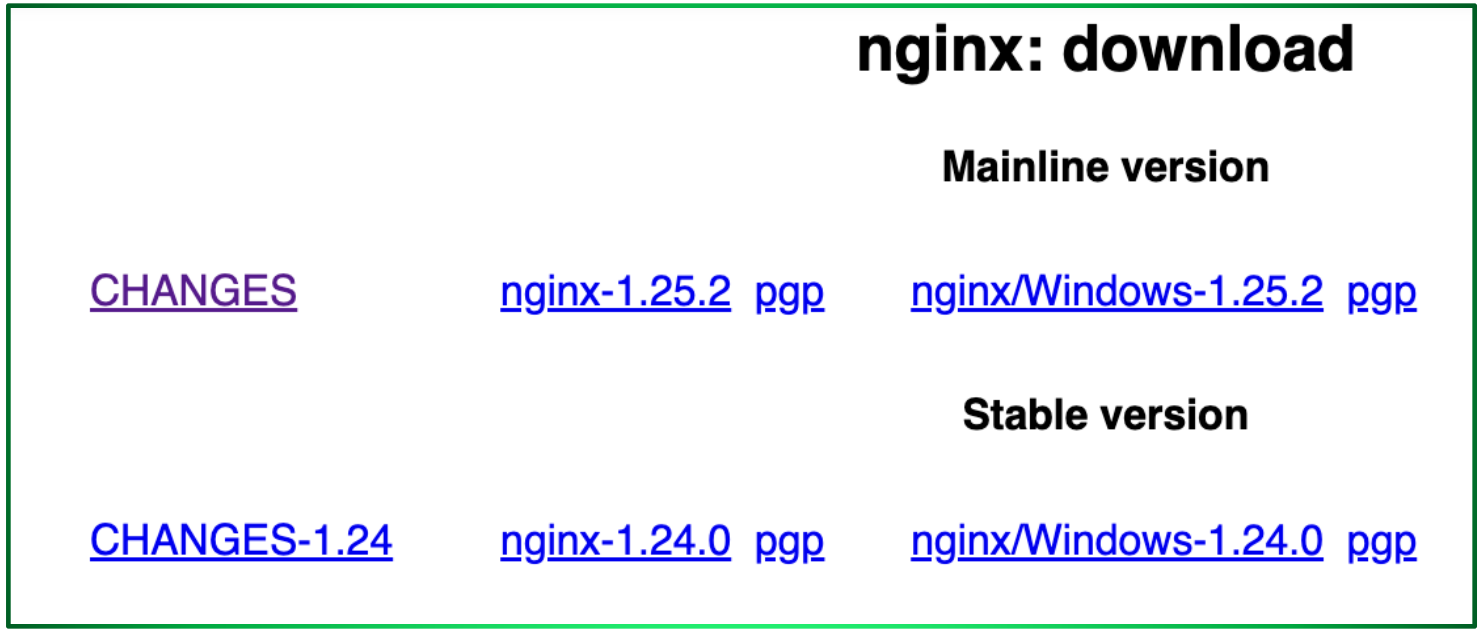
- Understanding community & enterprise needs
- Encouraging feedback loops
- Clear roadmap & versioning



loop.png (229×320) (xkcd.com)

But there are challenges

- Adopting strict versioning (e.g., Semantic Versioning)
- Offering Long-Term Support (LTS)
- Encouraging enterprise participation in the community



And even more challenges

- Licenses
  - Can you depend on the license not changing?
- Transparency
  - Does transparency lock the future stream?
- ”Unfair” competition
  - Circa 2017: The Public Cloud has killed open source"

Summing it up

- The Duel of Innovation vs. Stability
- Transparency and communication:
- Achieving Balance