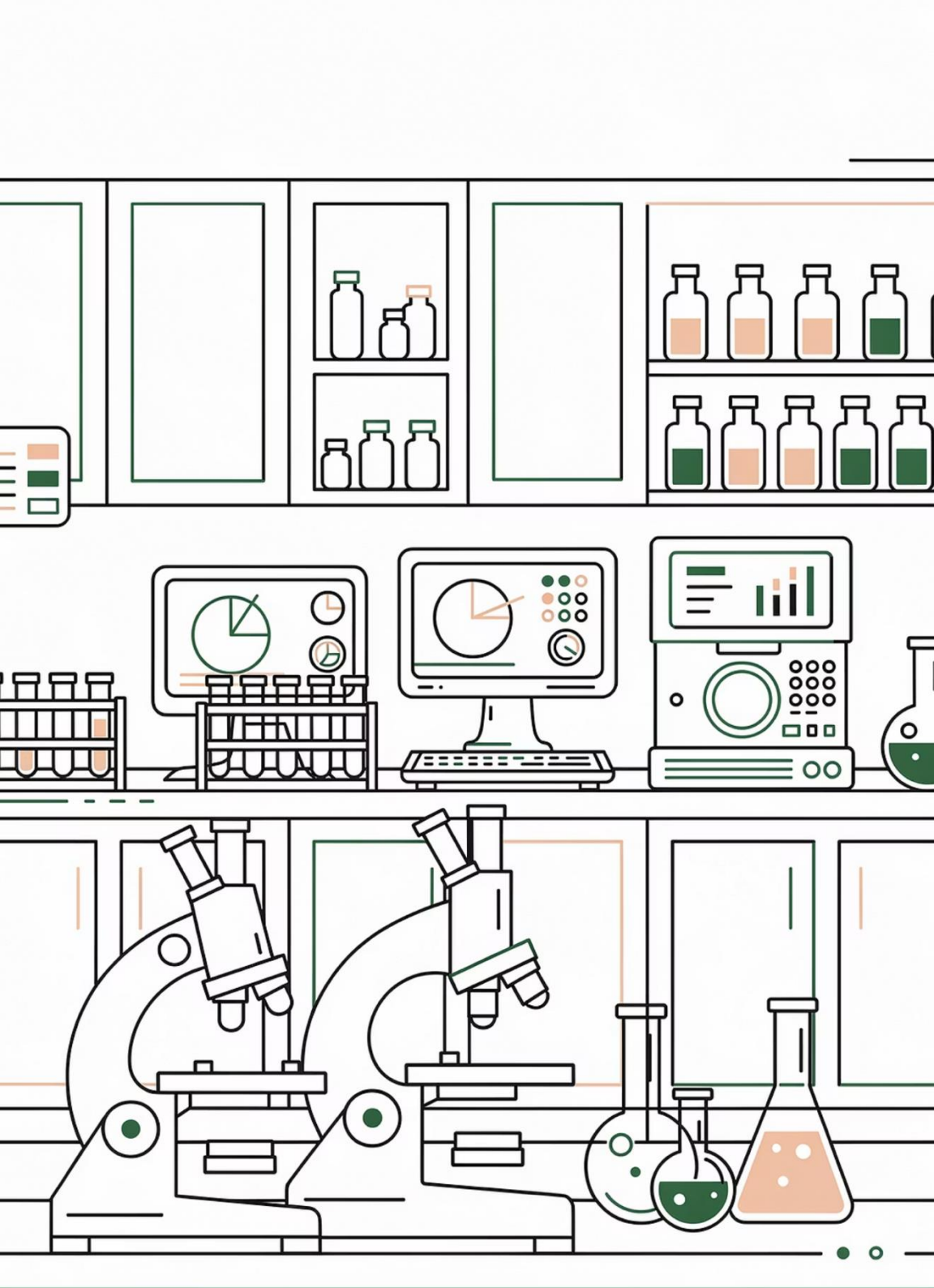




PHARMACEUTICAL INNOVATION

Git for Statistical Programming

Bridging proven technology with regulatory excellence in clinical trials

Rita Pecuch, Takeda, rita.pecuch@takeda.com

Jim Coates, KSM Technology Partners, jcoates@ksmpartners.com

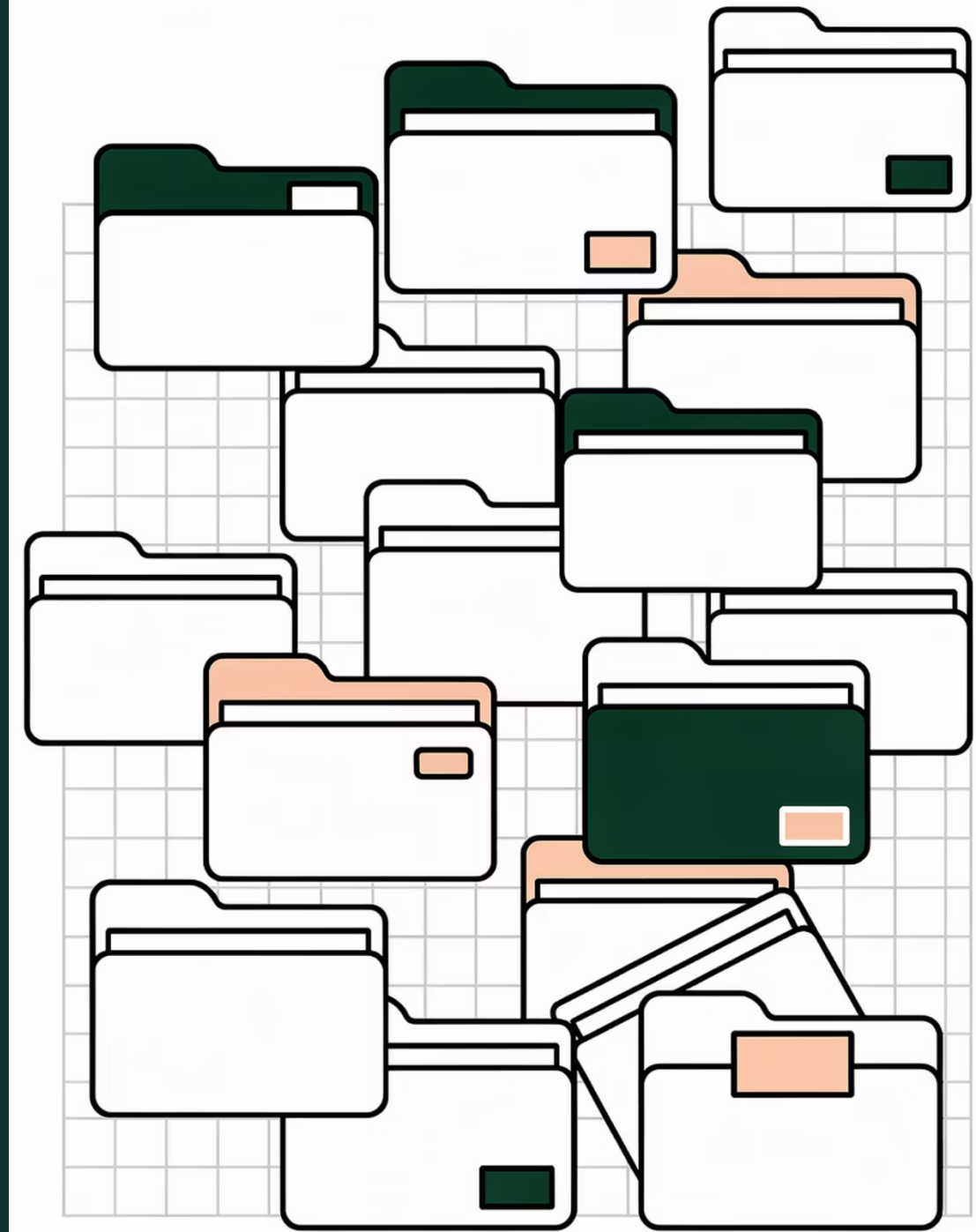
The Version Control Crisis

Today's Reality

- ADSL_v2_final_revised.sas
- ADSL_v3_USE_THIS.sas
- Manual audit trails
- Email-based QC reviews

The Cost

- Lost productivity
- Overwritten work
- Audit uncertainties
- Regulatory risk



Three Forces Driving Change

01

Efficiency Imperative

More trials, faster timelines, same resources—automation is no longer optional

02

Vendor Independence

Open-source portability protects your investment and workflow ownership

03

Industry Standardization

90%+ developer adoption, major pharma leaders, next-gen talent expectations

💡 KEY INSIGHT

What Git Actually Is

Tracks Every Change

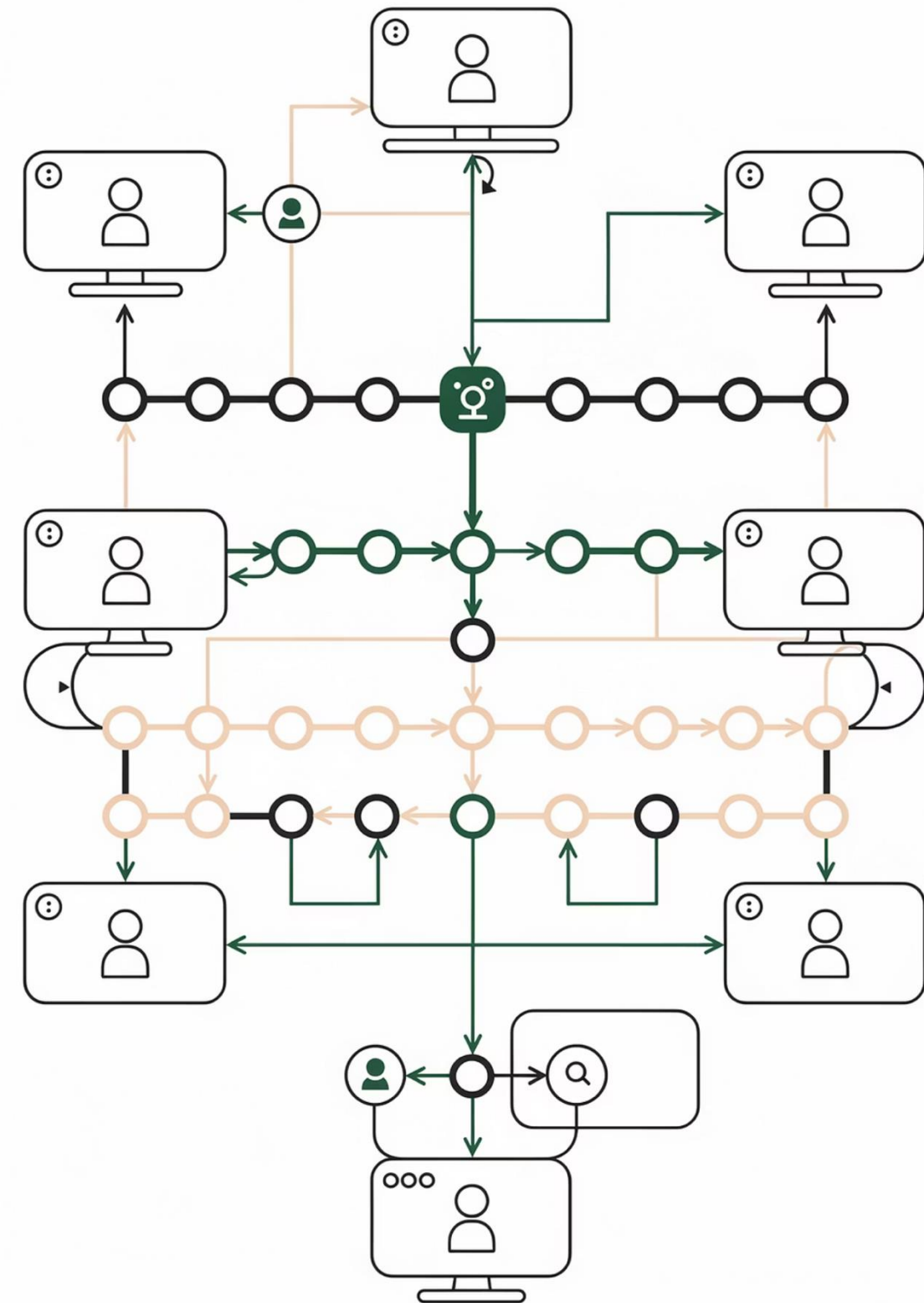
Automatic audit trail: who, what, when, why

Enables Collaboration

Multiple programmers, simultaneous work, zero conflicts

Instant Recovery

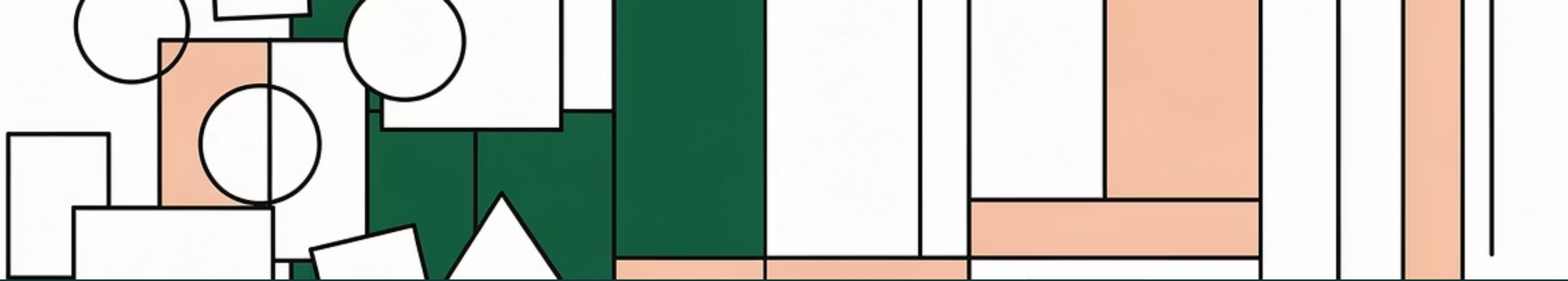
Any previous version accessible with one command



What Git Is *Not*



- Not Vendor-Owned
Open-source technology, not vendor-locked
- Not Only for Developers
Designed for regulated environments requiring audit trails
- Not an SCE Replacement
Works within your existing statistical computing environment
- Not Requiring Cloud
Works locally first; cloud hosting optional but recommended



Before Git vs. With Git

1

Traditional Approach

- Multiple "final" versions
- Unclear what's current
- Manual change documentation
- Detective work for attribution
- Storage bloat from duplicates

2

Git Workflow

- Single source of truth
- Always clear current state
- Automatic audit trail
- Built-in attribution
- Efficient delta storage

Git Enables New Capabilities



Safe Experimentation

Branch, test new approaches, abandon without risk—parallel development without fear



Precise QC Reviews

Compare exact code changes line-by-line rather than reviewing entire files



Compliance by Design

Regulatory requirements satisfied automatically through immutable history

21 CFR Part 11 Alignment



Authority Checks

Access controls determine who can commit to which branches



Operational Checks

Pull requests enforce mandatory review before production deployment



Audit Trail

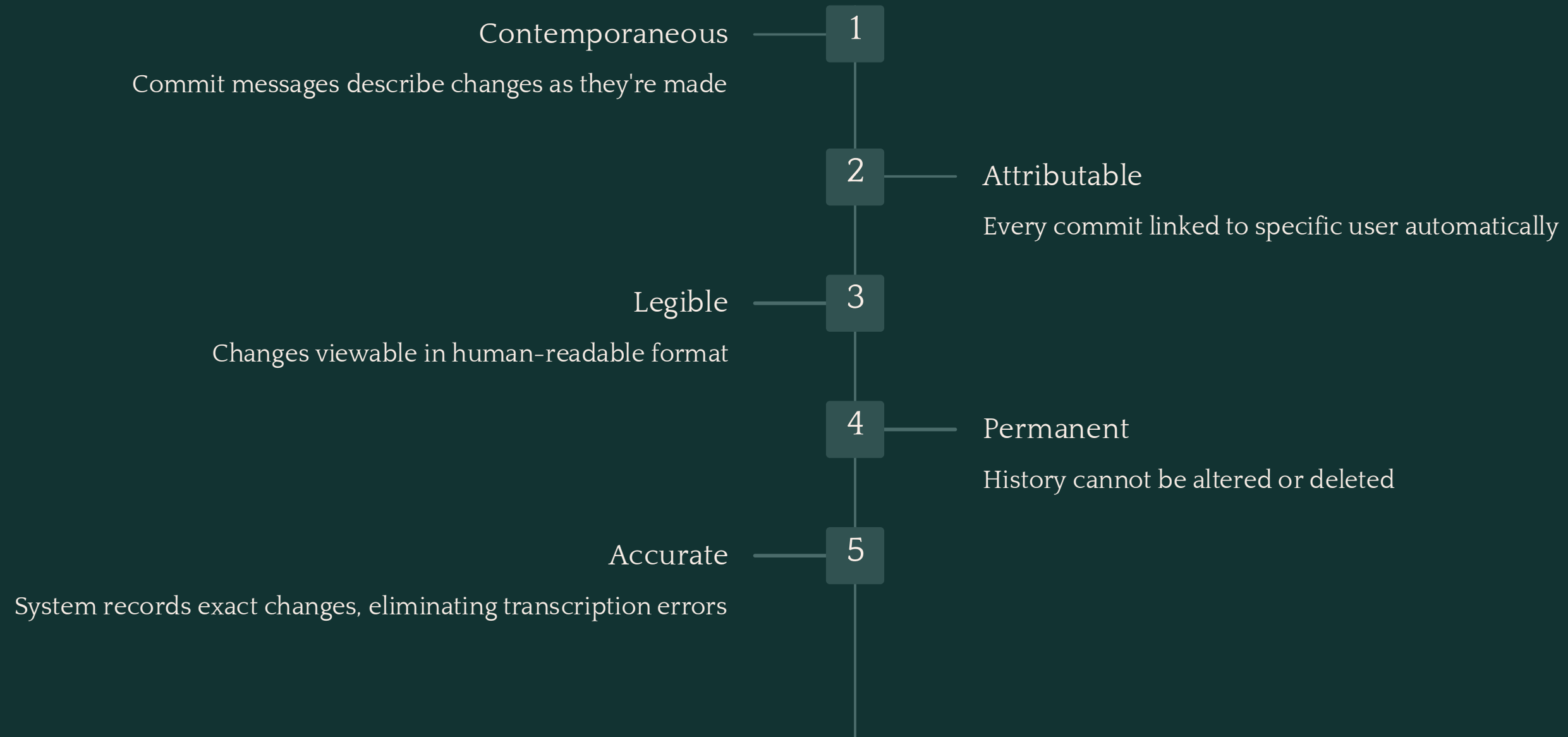
Immutable record of all changes with user ID, timestamp, rationale



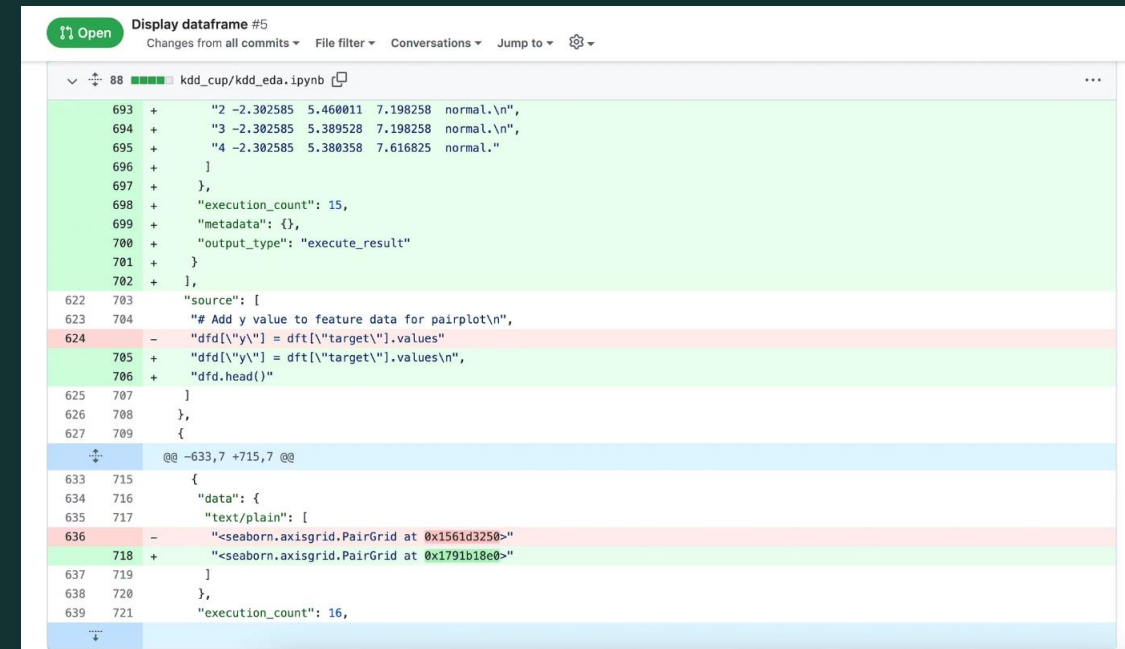
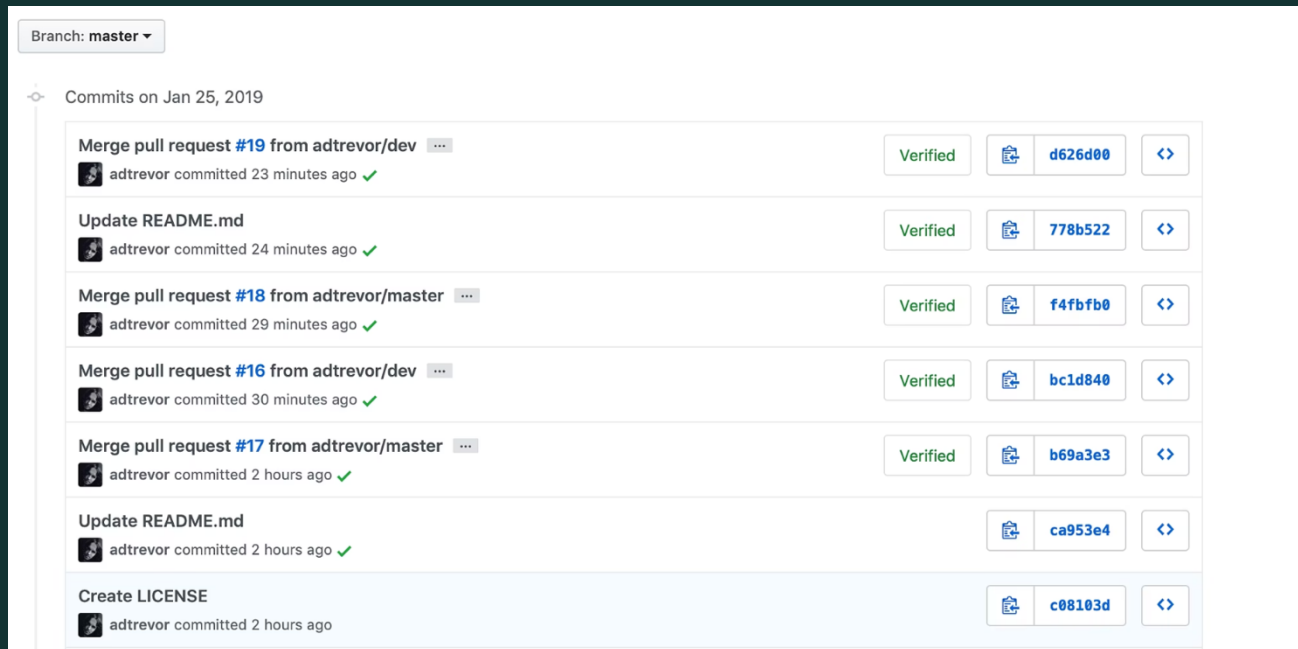
System Validation

Deterministic behavior and extensive documentation simplify validation

Good Documentation Practices



Visualizing Git's Strengths



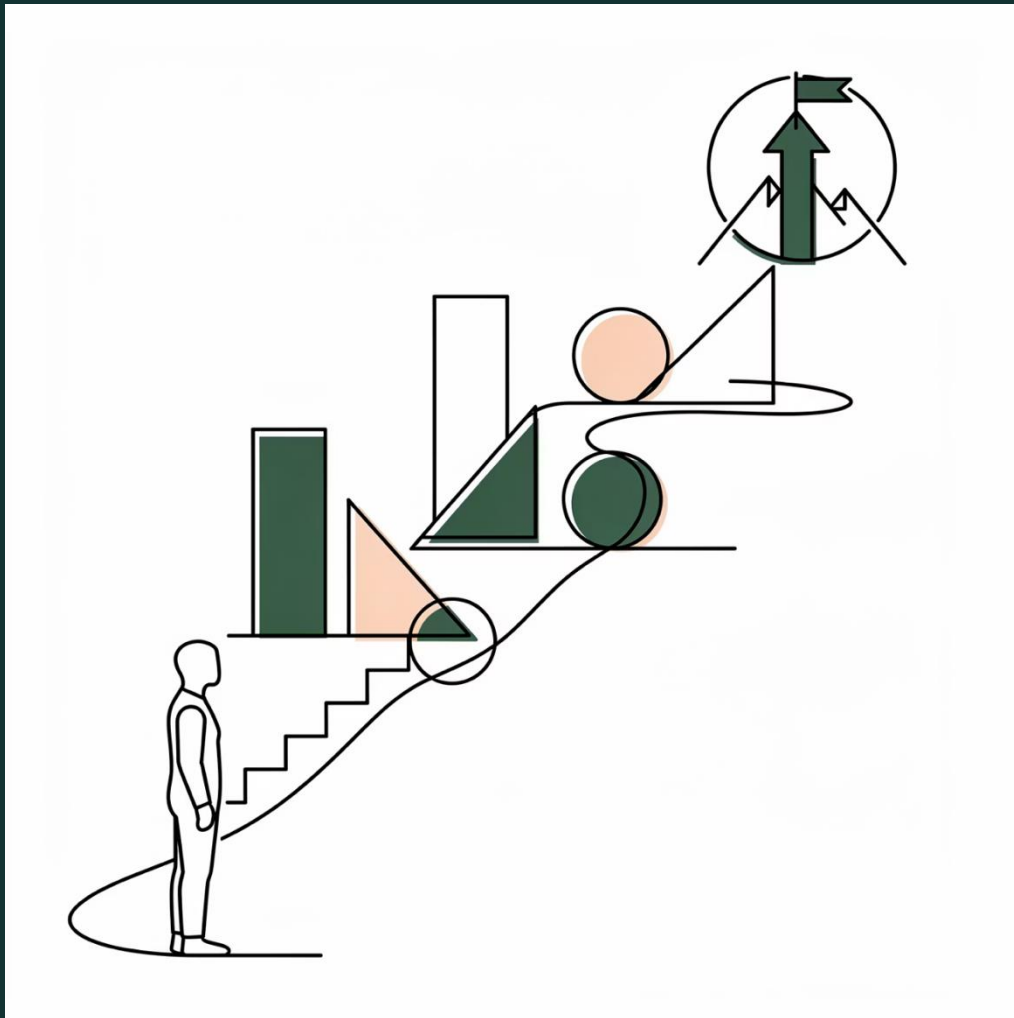
Commit History

Every change is recorded with who, what, and when. This creates an **immutable audit trail** for regulatory compliance and historical tracking.

Code Comparison

Precisely review **line-by-line code changes**, enabling thorough QC and efficient collaborative feedback through pull requests.

Addressing Common Concerns



"Too technical for statisticians"

Modern interfaces designed for non-developers. Basic proficiency develops within weeks.

"Our SCE has version control"

Vendor-specific versioning locks you in. Git provides universal, portable control.

"Learning curve will slow us down"

3-month adoption period. Programmers report wondering how they worked without it.



PHUSE Working Group Initiative



Cross-Industry Representatives

Major pharma, CROs, technology vendors collaborating on shared challenges



Shared Mission

Accelerate Git adoption through documented best practices and lessons learned



Growing Resources

Blog series, templates, case studies, training curricula, white papers

Our Deliverables

Strategic White Paper

Organizational adoption guidance

Case Studies

Real implementations at various adoption stages



Blog Series

Practical examples and code snippets

Implementation Templates

Repository structures, branching strategies

Training Materials

Curricula for varying technical backgrounds

The Decision Point

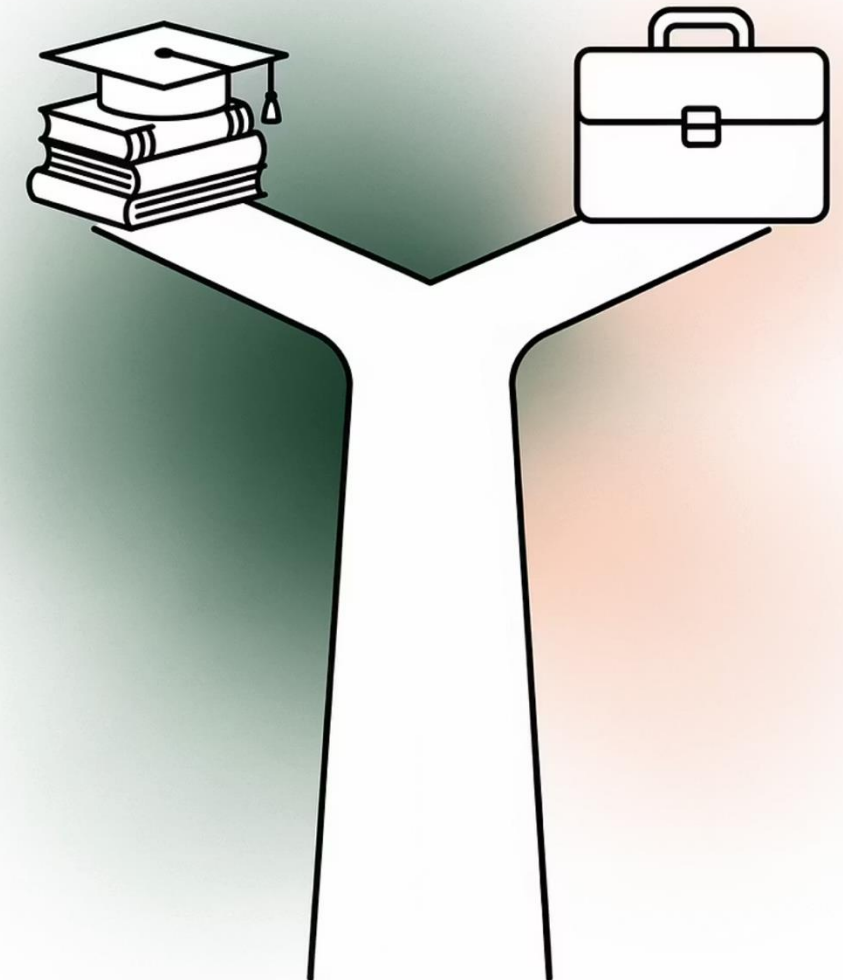
Proactive Adoption

- Deliberate planning
- Fits your culture
- Controlled timeline
- Lower switching costs

Reactive Adoption

- Business pressure driven
- Hurried transition
- Less control
- Higher costs

Question: Not *if* but **when** and **how**



Join the Movement

90%+

Developer
Adoption

Industry-standard
version control across
technology sectors

25+

Organizations
Collaborating

With representation in
The Use of Git in
Statistical
Programming working
group

3

Months to
Proficiency

Typical timeline from
training to productive
Git workflows

Ready to transform your statistical programming? Connect with the PHUSE Emerging Trends and Innovation Working Group for resources, guidance, and community support.

