

PHUSE US CONNECT 2026

# Beyond Box Checking

Five Best Practices for Embedding QC in Modern SCEs

Matthew Tendler · Head of Life Sciences Product Management · Domino Data Lab

# What we heard from your peers

**53%**

have NO dedicated  
QC tracking tool

**51%**

rely on manual  
documentation  
for traceability

**79%**

double-program  
ADaM 100%

**3.2/5**

traceability  
self-assessment

*Senior, SAS-dominant, Experienced professionals - still fighting manual processes.*

THE REALITY CHECK

# QC today is a parallel process, not an integrated one

## How it works today

- Email-driven approvals and Excel trackers
- QC bolted on at the end — after outputs are "done"
- Double programming everything by default
- Audit confidence depends on individuals, not systems

## How it should work

- Metadata-driven planning with risk-based categorization
- QC embedded where programmers already work
- Automated validation and reproducible workflows
- Full traceability via Git, lineage, and audit trails

SURVEY: When asked what features matter most - version history (55%), ease of use (51%), and audit trail (47%) topped the list.

The gap is real - but lessons from our customers prove it is closable. Our survey confirms the pain is widespread.

WHY THIS MATTERS NOW

# Four forces accelerating the need for change

## Regulatory Pressure

Agencies expect full traceability - but our survey shows Pharma orgs score just 2.6/5 on accessible traceability.

## Exploding Data

RWD, decentralized trials, complex endpoints - ADaM QC is the #1 effort bottleneck (cited by 77% of respondents).

## The R Transition

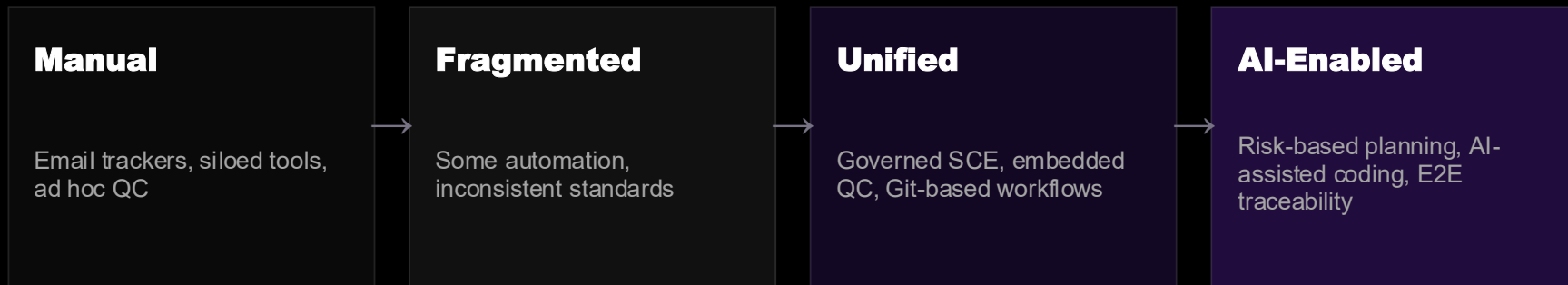
38% of surveyed programmers already use R alongside SAS. The multi-language SCE is non-negotiable.

## AI Readiness Gap

62% say "not at all" or "not sure." The foundation isn't there yet.

THE JOURNEY

# Where are you on the modernization curve?



Most teams are in Stage 1 - 2. Majority have no QC tool and use manual traceability · Less than half use Agile methods.

In our work with Large Pharma customers the order matters. Skipping from Stage 1 to Stage 4 is where modernization fails.

# Five best practices for embedding QC

1

## **Git-First Workflows**

Version control as the backbone of traceability

2

## **Automation by Default**

Outputs land in the right place, every time

3

## **Risk-Based Categorization**

Stop double-programming everything

4

## **End-to-End Collaboration**

Extend QC tracking to medical writers and beyond

5

## **AI-Ready Architecture**

Build today for the intelligence layer of tomorrow

*Each practice builds on the last — the sequence is deliberate.*

# Git-First Workflows

## What it means

- Every script, spec, and config lives in version control
- Branching strategies for production vs. exploratory work
- Immutable audit trails via commit history
- Code review as a QC gate - not a bolt-on

## Evidence

- Majority customers say version history is their #1 critical feature
- Majority still use manual documentation for traceability

Prescription: If you do one thing after this talk, put your statistical programs under Git. Highest-ROI modernization step.

# Automation by Default

## What it means

- Outputs land where they need to — no manual file shuffling
- Batch execution with dependency tracking
- Metadata-driven TFL generation pipelines
- Auto-snapshot datasets tied to jobs that produced them

## Survey evidence

- Only 37% agree automation has reduced their QC time
- 43% want dependency tracking as a critical feature
- 51% cite ease of use as most critical — automation should be invisible
- Upstream data changes must auto-flag downstream outputs as stale

Prescription: Automation isn't about removing humans — it's about removing the friction that keeps them from high-value work.

# Flexible, Risk-Based Categorization

79%

double-program ADaM 100%  
regardless of criticality

70%

for TFLs

49%

for ALL three domains

## What respondents told us

*"Dual programming hardwired into SOPs"*

*"Changes to older expectations of 100% independent programming"*

*"A moving away from outdated double programming that has become a tick box"*

## The modern approach

- Risk categories tied to output criticality
- QC depth scaled: double-program critical, review others
- AI-driven risk categorization suggestions
- Full version history of QC policy decisions

Prescription: Challenge the default. Not every deliverable deserves the same QC intensity. Smart planning saves weeks per study.

# End-to-End Collaboration Models

## Extend QC tracking beyond programming

- Medical writers need real-time TFL status visibility, not email updates
- Biostatisticians review QC plans before programming starts and feedback is auto-tracked
- Clinical data managers need to see upstream data change ripple effects
- Self-service dashboards replace status meetings and tracker spreadsheets

SURVEY: 30% cite cross-team coordination as a top 3 effort area. ADaM QC (77%) and TFL QC (60%) top the effort list — these are collaboration-heavy stages.

Prescription: If your QC tracker is an Excel file emailed between stakeholders, you are not tracking — you are hoping.

# AI-Ready Architecture

## Where AI pays its dividend

- Not just coding assistants — the real value is end-to-end orchestration
- Protocol → SAP → metadata → output generation
- AI-driven risk categorization and smart QC planning
- SAS and R remain non-negotiable for submission — LLMs augment, not replace

## Survey: what builds confidence?

- Industry standards
- Validation frameworks
- Clearer regulatory guidance
- Better audit trail/traceability
- Internal skills/training

Prescription: Build your SCE for where AI will be in 3 years. 45% said better traceability increases AI confidence - Best Practices #1-4 ARE the AI foundation.

# Where LLMs fit in Big Pharma today

## High Value Today

Code translation (SAS→R), spec drafting, log summarization, knowledge retrieval, synthetic data for testing

## Emerging (1–2 yrs)

AI-assisted ADaM/TFL generation, AI-driven QC risk categorization, NL data exploration, automated CSR drafting

## Transformative (3+ yrs)

End-to-end protocol→submission orchestration, cross-study intelligence, AI project managers for deliverables

For stat programmers: your role evolves from writing every line to orchestrating, validating, and governing AI-assisted workflows. Deterministic statistical packages aren't going anywhere but your role will evolve.

# What to take home

1

## **Embed QC upstream**

Quality built into workflows beats quality checked at the end

2

## **Git is your foundation**

55% of your peers say version history is their #1 need — give it to them

3

## **Automate relentlessly**

Only 37% see automation ROI today — because piecemeal doesn't work

4

## **Challenge default double-programming**

79% double-program ADaM 100% — even your peers say SOPs haven't caught up

5

## **Build for AI now**

45% say better traceability increases AI confidence — BPs #1-4 ARE the foundation

The organizations that will lead in 2028 are the ones investing in their SCE foundations today.

THANK YOU

# Q&A

Matthew Tandler

Head of Life Sciences Product Management · Domino Data Lab

[matthew.tandler@dominodatalab.com](mailto:matthew.tandler@dominodatalab.com)