



When AI Writes the TLFs

Transforming Clinical Reporting and Redefining the Programmer's Role

An end-to-end AI workflow for TLF development – and what it means for statistical programming

DISCLAIMER

Important Notice

The views expressed in this presentation are those of the presenter and are not endorsed by PHUSE or Jazz Pharmaceuticals.

Where We Are Today

AI in clinical reporting has moved from concept to practice



LLMs Are Capable Enough

Modern models can read a SAP, interpret analysis intent, and generate structured specs, shells, and code - with the right guardrails



The Tooling Exists

Cloud LLM platforms, RAG provides standards and SOP context, and secure enterprise environments make production use feasible today



Standards Are Aligning

CDISC ARS, machine-readable protocols, and metadata-driven pipelines create the structured inputs AI needs to work reliably



Sponsors & CROs Are Piloting

Across the industry, organizations are moving from proofs-of-concept to working prototypes and early production use



What Comes Next

We have the tools and infrastructure. The next challenge is integrating them deeply into day-to-day programming workflows.

- Identifying workflows that benefit most from automation
- Building solutions for those areas
- Evolving how our programmers work

The Statistical Programmer in 3 Years

From code writer to clinical data engineer

What Goes Away	What Stays	What's New/Rising
• Writing boilerplate SDTM/ADaM/TLF code • Manual shell creation from SAP • SAS syntax memorization • Manual checks for style and formatting	• CDISC standards & regulatory expertise • Clinical domain knowledge • Debugging & root cause analysis • Cross-functional collaboration	• Metadata design & specs that drive automation • AI tool proficiency & prompt engineering • Analytical QC & clinical plausibility review • Pipeline orchestration

The Mindset Shift: *From "I write code that produces this table" to "I own the quality of this table – however it gets produced"*

Managing the Risks

What we need to get right



Regulatory Traceability

Every AI-generated deliverable must be fully traceable. FDA 2025 guidance requires credibility assessment. Document when and how AI was used.



Data Privacy

Clinical data must never go to external LLMs. Enterprise deployments address this – but policy and training must be airtight.



Over-Reliance Without Review

The biggest quality risk: a programmer trusts AI output without scrutiny. Questioning AI output is expected, not optional



Tool Dependencies

AWS, MS Azure, Snowflake, Off-the-shelf SCEs, etc – if pricing or accessibility terms change, we need contingency. Open-source components reduce risk of being locked into single set of vendors.



Team Adaptation

The programming role is changing, not disappearing. Programmers who invest in new skills will be more valuable.



Double Programming

Two programmers relying on the same AI tools dilutes traditional independence. Consider Different toolchains, different prompts/models, multi-agent validation, metadata/spec QC.

Staffing Models & Future Demand

How AI efficiencies may reshape the shape of the profession



Efficiency per programmer rises

If a programmer can produce and QC far more output per unit time, the same volume of work requires fewer hours — placing downward pressure on headcount for routine production.



But total demand may also grow

Lower cost per analysis can expand what sponsors ask for — more exploratory work, faster iterations, more submissions — potentially absorbing freed capacity rather than eliminating it.



The net effect is genuinely uncertain. It will vary by organization, therapeutic area, and how aggressively AI is adopted — the honest answer is that roles will shift more predictably than total demand.

What is more certain

Skill mix reshapes

Demand tilts from production coding toward specification, validation, and oversight

Team shapes flatten

Fewer large pools of production programmers; more senior reviewers and architects

Value moves upstream

The people who define standards and metadata become the leverage points

How Organizations Can Prepare

Practical moves that hold up regardless of how demand evolves



Invest in Upskilling Now

Build R, Python, and AI-tool fluency across the team before it's urgent. The transition is smoother when it's proactive, not reactive.



Structure Data & Standards

AI works best on structured inputs. Machine-readable SAPs, metadata standards, and clean templates are prerequisites, not afterthoughts.



Redesign Roles Deliberately

Define what specification authors, validation strategists, and pipeline architects actually do — don't let the new roles emerge by accident.



Govern AI from the Start

Traceability, human oversight, and validation approaches for AI output should be designed in early, not retrofitted after a finding.

Three Takeaways

01 **AI can already transform the SAP-to-TLF workflow.**

A working prototype shows specs, shells, ADaM requirements, and code generated end-to-end — with humans reviewing at every stage. This is practical, not speculative.

02 **The programmer's role shifts from writing to orchestrating.**

As routine coding automates, value concentrates in specification, validation, domain judgment, and system design — skills that are harder to build and harder to automate.

03 **Headcount demand is unclear.**

Whether total headcount grows or shrinks is uncertain and context-dependent. What's clear is that the skill mix is changing — and the time to prepare is now.