



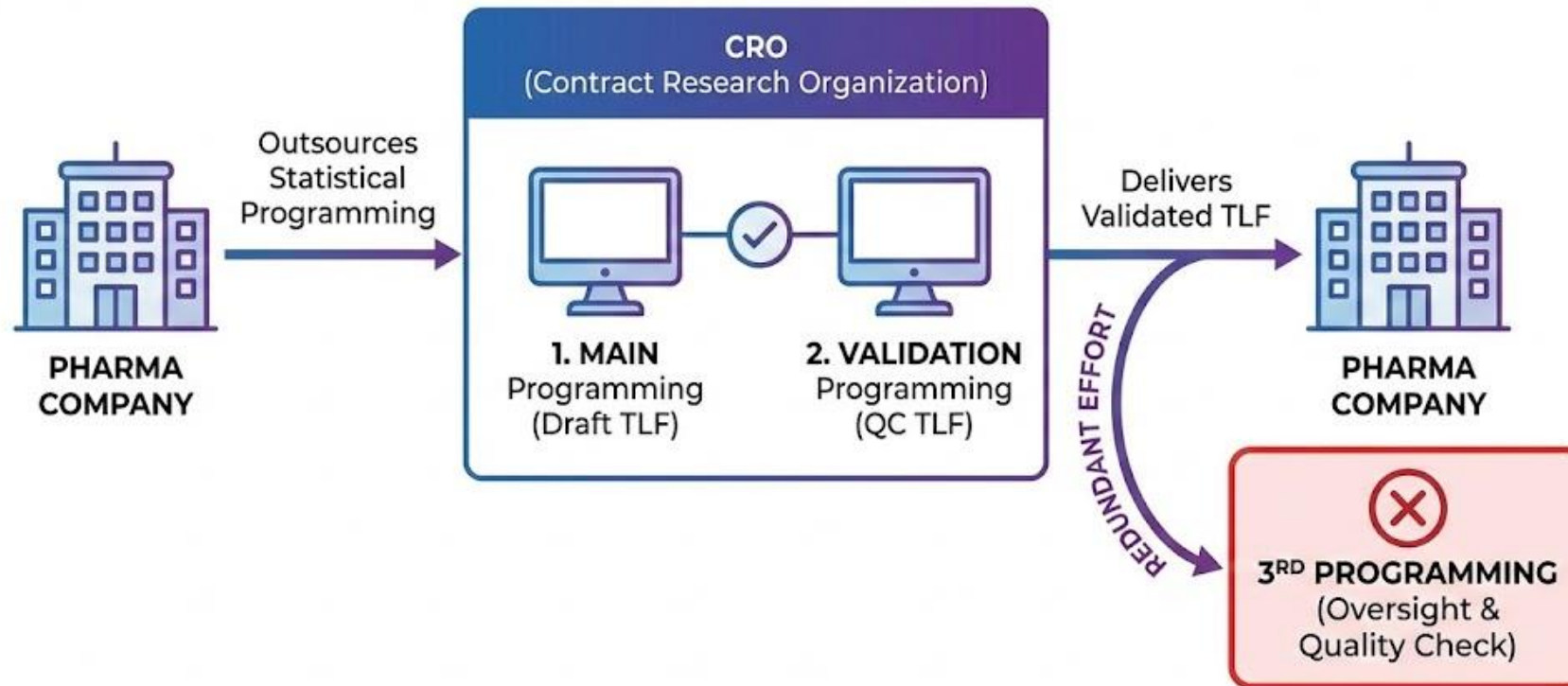
QC Without Code: ChatGPT Enterprise for Sponsor Oversight of Outsourced TLFs

ML24 - PHUSE US Connect 2026

Unnat Patel, Brad Moore, Jenny Price,
Brent Burger - UroGen Pharma

Kate Roth - LSU

Oversight



Current State: Pharma companies remain responsible for TLF quality, leading to a redundant third programming effort despite CRO's double programming.

Sponsor Quality Control Responsibility

- Deliverables frequently outsourced
- Sponsors remains accountable for deliverables
- Timeline and cost pressures



Why Consider Generative AI for QC?

- Large Language Models (LLM) can interpret datasets and written specifications
- No-code workflows reduce traditional programming dependency
- Potential to expand QC coverage with less resources

Core Questions and Objectives

Can ChatGPT Enterprise:

- independently generate TLFs from ADaM?
- reliably identify discrepancies with production outputs?

Can this be done in a controlled, governed workflow?

Method

Design Principles

- ADaM-only data foundation
- Independent, no-code approach
- Human-in-the-loop oversight

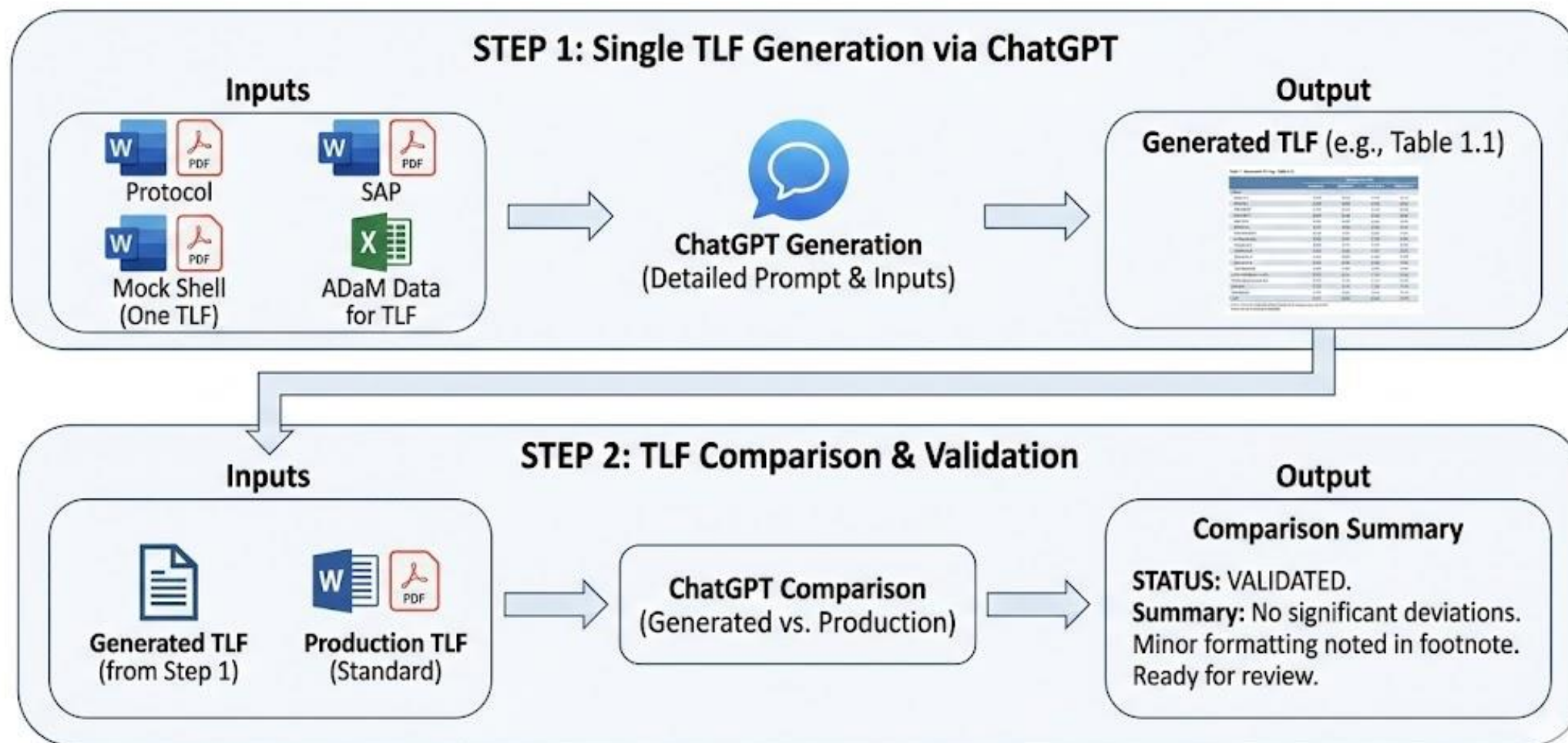
Stage I – Proof of Concept

- Single completed study
- Demographics table and Kaplan–Meier plot
- Free-text prompts in ChatGPT Enterprise
- Goal: assess basic feasibility

Stage I – Key Observations

- Demographics tables reproduced accurately
- KM plots required prompt iteration
- Prompt specificity drove output quality

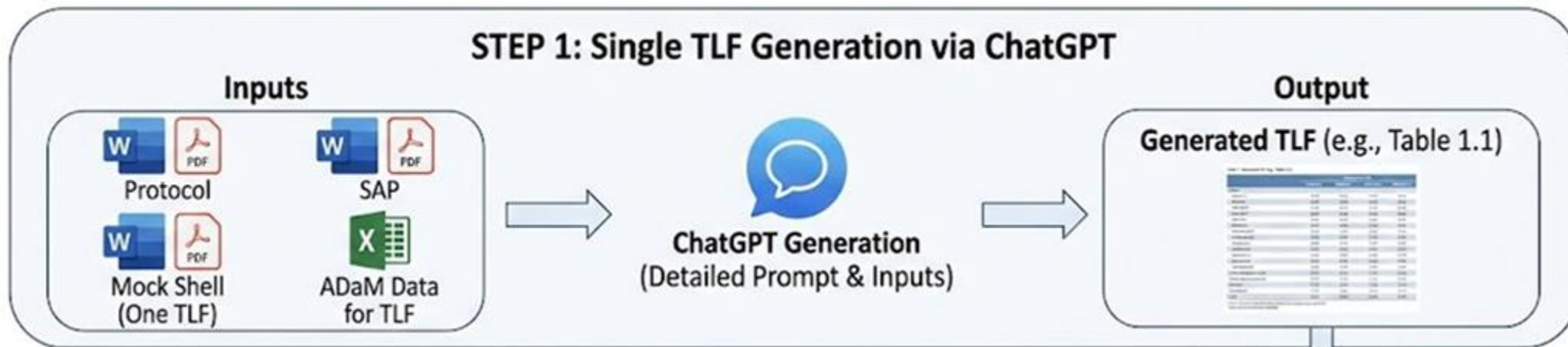
AI-Assisted Single TLF Generation and Validation Workflow



**Process is repeated iteratively for each individual TLF.*

Stage II – Generic Prompt Engineering

- Develop a reusable prompt template
- Explicit constraints and stepwise workflow
- Designed for repeatability and validation



Stage II – Validation Strategy

- Control
 - INPUT (Step 1): Study docs and validated ADaM extracts
 - OUTPUT (Step 1)/INPUT (Step 2): ChatGPT-generated TFLs
 - OUTPUT (Step 2): Standardized TLF comparison reports
- Repeatable test scripts
- Multiple runs under identical conditions
- Formal exception logging

Stage III – Workflow Integration

- Pilot on interim reporting deliverables
- Execution records archived
- Embedded into risk-based QC framework
- Liaise with Quality Assurance team - process documentation

Discussion

Four Categories of Issues Observed

- 1) Documentation gaps
- 2) ADaM coding/derivation issues
- 3) Prompt/template defects
- 4) LLM numeric/logical slips

Accuracy and Feasibility Results

- High consistency with vendor outputs
- Stable results across repeated runs
- Most issues resolved via clarification, not rework
- Supports feasibility of no-code QC

Efficiency Gains

- Reduced need for independent programming
- Shifted time programming to structured review
- Lowered incremental effort per unique TLF
- Expanded more QC coverage to lower-risk tables

Documentation and Traceability

- Prompts, data versions, and model versions logged
- Structured QC summaries generated
- More consistent documentation across studies

Limitations

- Does not replace upstream ADaM validation
- Large datasets may surpass context limits
- Model behavior can change over time
- Complex statistical concepts need more instruction

Conclusions

- ChatGPT can function as an independent “virtual programmer”
- AI-assisted QC complements traditional validation
- Governance and human oversight remain essential
- Practical, scalable approach for sponsors

Future Directions

- AI-assisted TLF development
- SDTM-based cross-checks for selected TLFs
- Broader portfolio-level QC coverage
- Agentic workflows?

Thank You!

Special thanks to **Unnat Patel, Jenny Price, Brent Burger** from UroGen Pharma and **Kate Roth** from Louisiana State University