



The AI Reckoning: Why Statistical Programmers Must Evolve Now

Paper & Presentation # ET06

PHUSE US Connect 2026

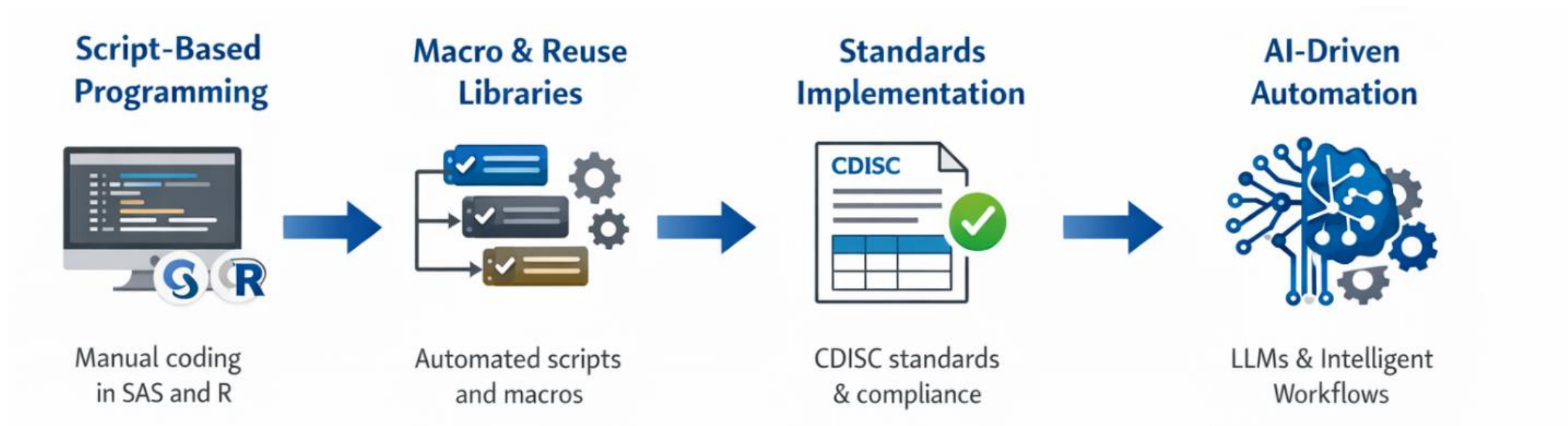
March 23rd, 2026

Bhavin Busa, Principal & Co-Founder, Clymb Clinical

Every Mission Has a Reckoning



Evolution of Statistical Programming



**AI is the new
superpower, not
the replacement.**



Key Pillars of the AI-Enabled Statistical Programming Framework

From Code-Centric
to Model-Centric
Programming

CDISC as the Enabler,
Not the Constraint

Where LLMs Fit,
and Where They
Do Not

A Practical
Architecture for
SAS, R, and AI
Co-Existence

Organizational
and Skill Challenges

The Evolving Role
of the Statistical
Programmer

Code-Centric vs. Model-Centric Statistical Programming

Code-Centric Programming



- Extensive manual coding
- Script-heavy, error-prone
- Difficult to scale
- Limited traceability & auditability

Model-Centric Programming



- Metadata-driven automation
- Metadata-based, reusable models
- Scalable & consistent
- Structured traceability & audit trail

CDISC As The Enabler, Not the Constraint

- CDISC is shifting from compliance to an automation backbone (CDISC 360i)
 - USDM models study design in a machine-readable way
 - ARS formalizes analysis definition, execution, and reporting with traceability
 - SDTM/ADaM remain core building blocks, but USDM/ARS model intent and amplify automation
- AI depends on structured, machine-readable inputs, not free-text specifications
- Structured models + JSON constraints make LLM outputs reproducible, auditable, and inspection-ready

Where LLMs Fit, and Where They Do Not

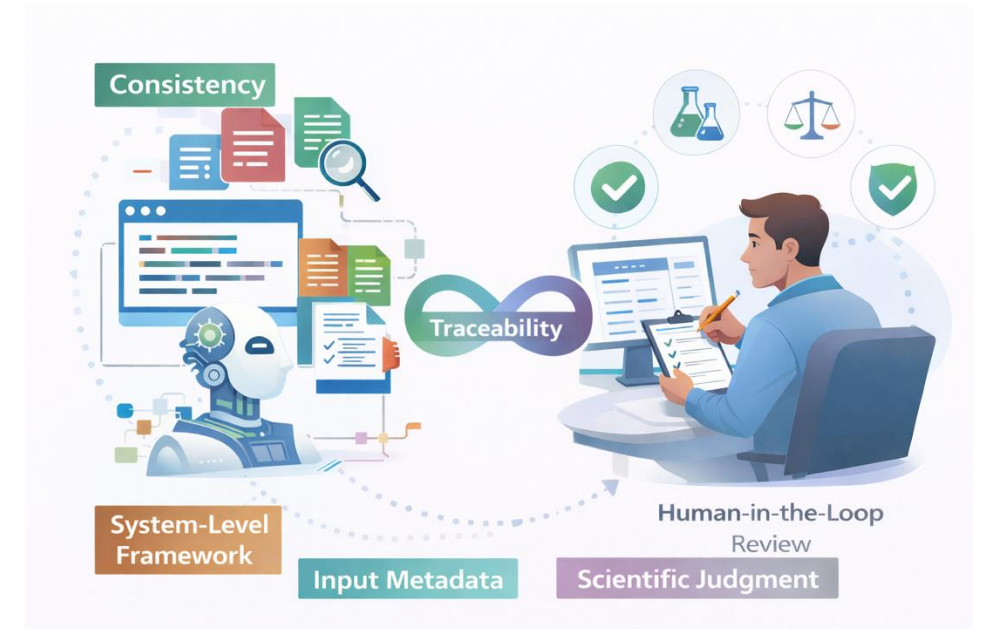
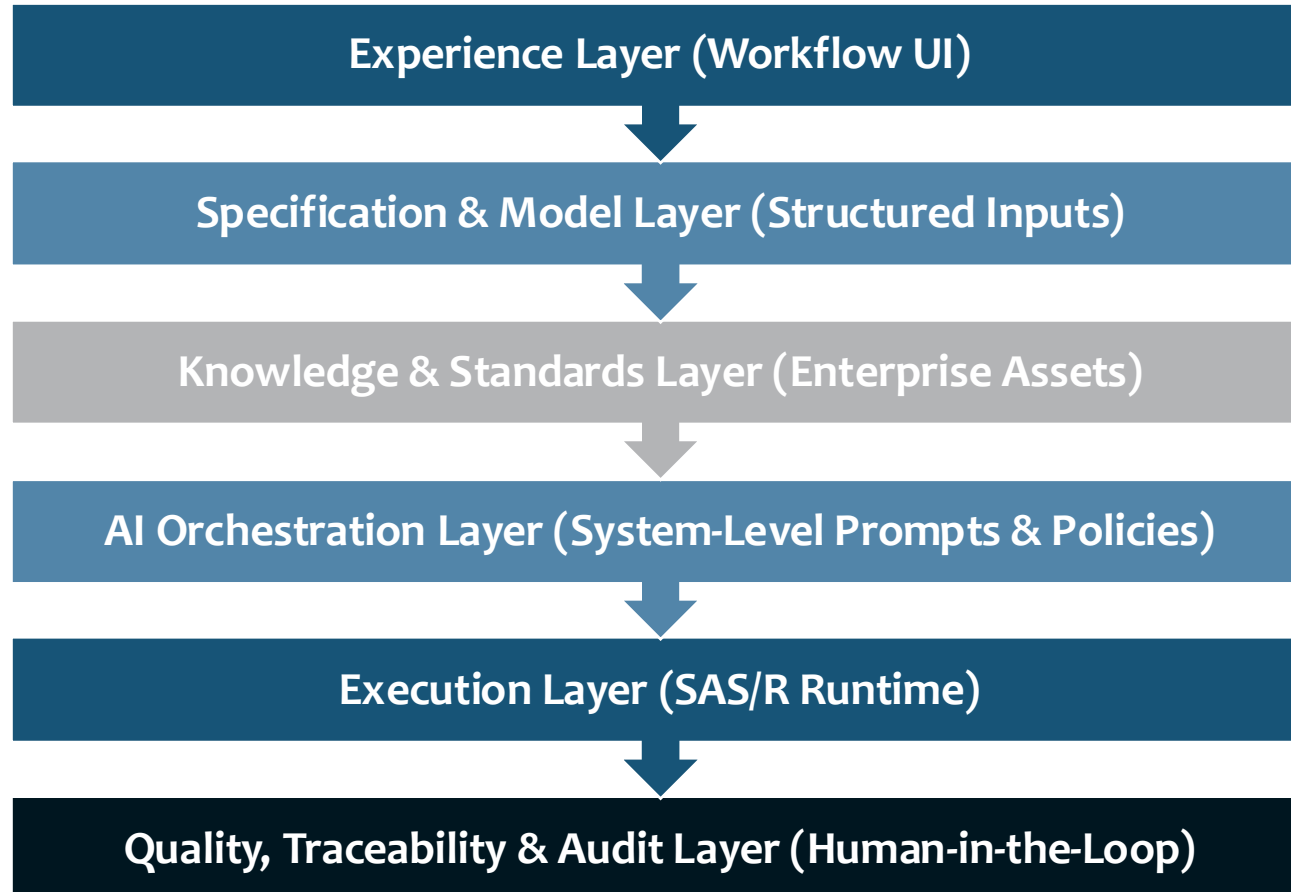
■ Appropriate use cases

- Generate SAS and R code from structured, model-driven metadata
- Support meta-programming and standardized logic
- Assist with mapping specifications and consistency checks

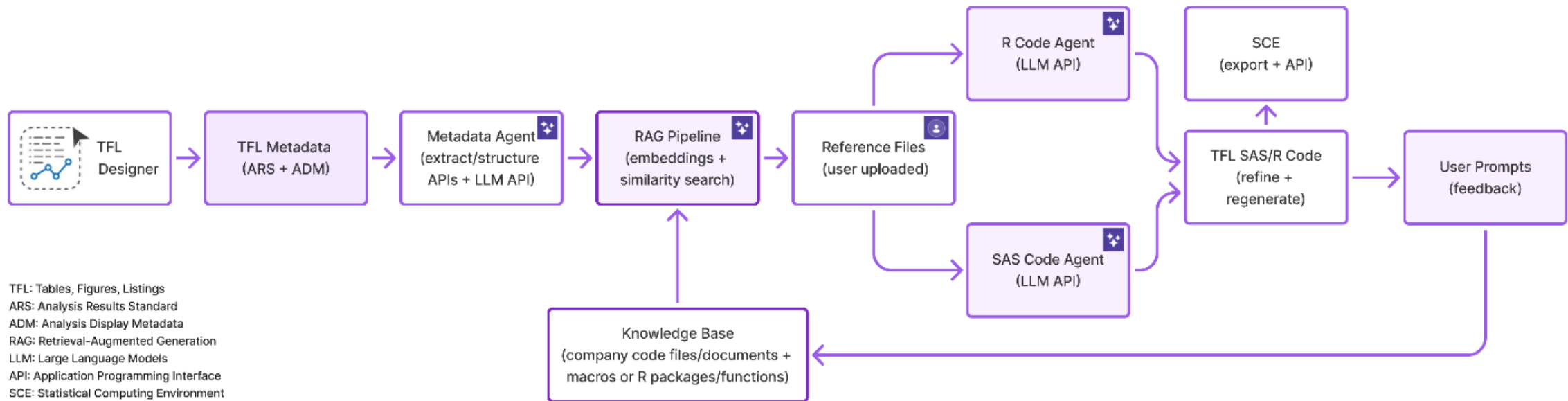
■ Poor use cases

- Autonomous definition of analysis intent
- High-stakes statistical or regulatory decision-making
- Unconstrained interpretation of free-text specifications

A Practical Architecture for SAS, R, and AI Co-Existence



Reference Implementation – AI Code Generator



Reference Implementation – AI Code Generator

The screenshot displays the TFL Designer interface, specifically the 'AI Code Generator' window. The window is titled 'Table 14.1.1.3.1 (Current Shell)' and shows the 'Summary of Demographic and Baseline Characteristics' table. The code generated is SAS code, and the user has selected 'GPT-4.1' as the model. The code includes comments for Program Name, Purpose, Project, Input Datasets, Output, Created By, Date Created, Table ID, Table Title, and Shell Source. The user has added two reference codes: 'eventfreq.sas' and 'univars.sas'. The code is displayed in a text area with a 'Copy' button and an 'Export' button. Below the code area is a 'Refine the code with custom prompt' text area and a 'Refine Code' button. The background shows the 'Clymb Pilot 302' table structure with columns for Characteristics, Age, Gender, and Age Group.

Table 14.1.1.3.1 (Current Shell)
Summary of Demographic and Baseline Characteristics

SAS Code: GPT-4.1

eventfreq.sas univars.sas

```
/* ***** */
/* Program Name : t14113_sum_demo.sas */
/* Purpose : Generate Table for Summary of Demographic and Baseline Characteristics */
/* Project : Clymb Pilot 302 */
/* Input Datasets : ADaM: ADSL */
/* Output : &out./t14113_sum_demo.rtf */
/* Created By : Bhavin Busa */
/* Date Created : 02FEB2026 */
/* Table ID : Table 14.1.1.3.1 */
/* Table Title : Summary of Demographic and Baseline Characteristics */
/* Shell Source : TFL Designer Enterprise */
/* ***** */
```

Copy Export

Refine the code with custom prompt

Show System Prompt Clear Keep Refine Code

© Copyright Clymb Clinical LLC

Organizational and Skill Challenges



Evolving Skill Profiles



Governance and Control Models



Validation and Quality Frameworks



Change Management and Adoption



Accountability and Human Oversight

The Evolving Role of the Statistical Programmer



**Specification
Ownership**



**Model and
Metadata
Stewardship**



**Enterprise
Standards
Alignment**



**AI-Guided
Programming
Oversight**



**Quality and
Traceability
Accountability**



**Cross-
Functional
Collaboration**

In an AI-enabled world, Statistical Programmers become architects of workflows, not just authors of code.

The AI Reckoning

Ad hoc Code

Free-Text Specs

Siloed Tools

The Reckoning



Workflow UI



Structured Specifications

Models & Metadata

SAS / R Execution

Human Review

This is the moment where we decide how AI is used.

Thank you!





Bhavin Busa
Principal & Co-Founder
bhavin@clymbclinical.com



EMAIL

info@clymbclinical.com



WEBSITE

www.clymbclinical.com



PHONE

781-692-3613