

DAILY ENGLISH READING

13 September 2026 - 30-minute active reading pack

TODAY'S QUESTION

Can you defend an AI-generated SDTM, ADaM, or TFL?

Difficulty: C1 | AI-native SCE | governance | provenance | ClinAgent

WHY SELECTED

The fresh production architecture separates AI proposal, human approval and deterministic execution. The key issue is not whether AI can create code or data, but whether the result can be reconstructed from controlled inputs, versions, rules, validation evidence and approvals.

READING ORDER

0-3 min Preview: distinguish proposal, approval and execution.

3-14 min Main: governance, pinned versions, snapshots and provenance.

14-20 min ClinAgent: thin tools, thick skills and rule engines.

20-25 min CDISC: machine-readable standards-based workflows.

25-30 min Output: redesign one SP task for defensibility.

CORE ARCHITECTURE

source data -> AI proposal -> human approval -> deterministic execution

-> automated validation -> computational provenance -> controlled release

Probabilistic reasoning and controlled execution should be separate.

Every approved version should be reconstructable.

Vendor performance claims are signals, not independent validation.

WHAT TO FREEZE

source-data snapshot; specification / metadata version; ADaM version

program or executable-rule version; skill / model version; environment state

validation results; reviewer decision; approval timestamp

The final artifact is only one part of the evidence package.

KEY VOCABULARY

defensible output	可辩护输出	result that can be reconstructed and justified
computational provenance	计算谱系	data code environment and transformation history
immutable snapshot	不可变快照	preserved state that cannot silently change
governed pipeline	受治理流程	controlled permissions versions and transitions
human approval	人工批准	explicit accept reject or escalation decision
deterministic execution	确定性执行	same controlled inputs produce same result
auto-mapping	自动映射	automatic source-to-standard proposal
auto-remediation	自动修复	system-generated correction under controls
version pinning	版本锁定	fix workflow to exact approved versions
audit trail	审计追踪	chronological actions changes and approvals
exception-based review	异常审阅	humans focus on uncertain or failed cases
domain logic	领域逻辑	CDISC SAS statistical and sponsor rules
stateless tool	无状态工具	narrow input-output operation
compliance infrastructure	合规基础设施	logging masking access and validation controls
agent lock-in	智能体锁定	dependency on one model or agent host

USEFUL PHRASES

1. the central question is whether the output can be defended
2. AI should propose while controlled systems execute
3. every approved version should be reconstructable
4. human review should concentrate on exceptions and judgment
5. domain logic should be separated from data access
6. the same input should reproduce the same approved output
7. a model suggestion is not the same as an executed transformation
8. automation must preserve provenance across handoffs
9. vendor efficiency claims require independent validation
10. production readiness depends on governance as much as capability

COMPREHENSION QUESTIONS

1. Why separate AI proposal, human approval and deterministic execution?
2. What belongs in the computational provenance of a final TFL?
3. How does Thin MCP, Thick Skills separate access from expertise?
4. Why are vendor efficiency claims different from independent evidence?
5. Which SP activities fit exception-based review and which remain human-led?

RETELLING PROMPTS

30 sec: explain generation versus defensibility.

45 sec: source -> proposal -> approval -> execution -> validation -> provenance.

60 sec: divide responsibility among LLM, skill, rule engine, SCE and programmer.

5-MINUTE SPEAKING / OUTPUT TASK

Minute 1: choose SDTM, ADaM, log QC, TFL programming, TFL QC or eSubmission.

Minutes 2-3: assign AI, skill, execution, validation and human responsibilities.

Minute 4: define the exact provenance package to freeze or log.

Minute 5: give a production go/no-go decision.

ONE SENTENCE TO KEEP

A production-grade SP agent should not merely create the right output once; its data, rules, execution, validation, versions and approvals should all be reconstructable and defensible.