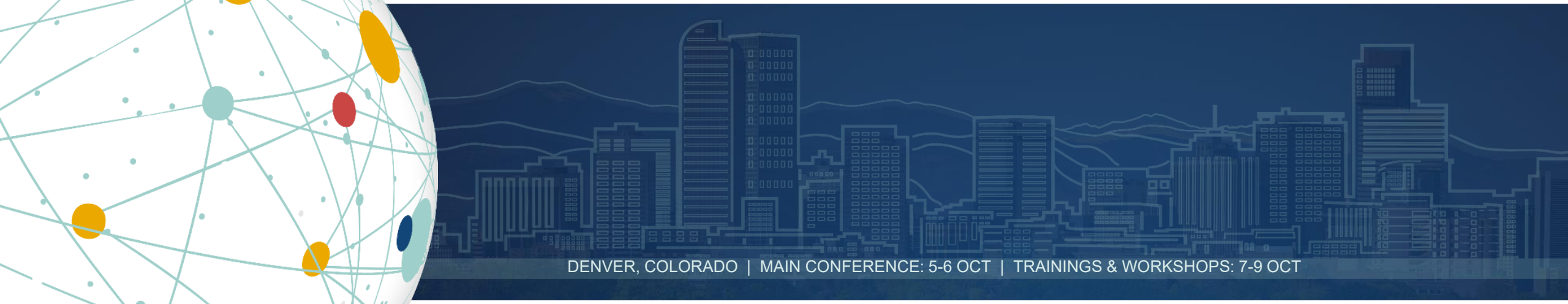


cdisc®

2026 US Interchange

THE FUTURE IS CONNECTED: STANDARDS AND AI POWERING DIGITAL TRANSFORMATION



Accelerating SDTM Engine Development and Adoption

with a Multi-Agent AI Framework and RAG-Based Assistant

Srini Malipeddi | Proma Debnath
Merck & Co., Inc., North Wales, PA, USA

INTEGRATING AI INTO A GOVERNED LIFECYCLE

The lifecycle we already had, and what AI changed

1. **Dividing the work**
   Four accountable roles and a fixed sequence of gates, defined before AI.

2. **Making AI assistance defensible**
   Rules are written into the standards; some authorities stay with humans.

3. **Widening access to the library**
   A grounded assistant answers from our own Git-based knowledgebase.

4. **What we found in practice**
   Faster development and wider access to expertise, review burden intact.

what we were willing to delegate, and what we were not.

THE SDTM ENGINE

One authoring step replaces the spreadsheet handoff

Conventional model: a hand-written Excel spec, then programmers translate it into code.

1 Select and configure

Pick the study; set IG and dictionary versions.

2 Author mappings

Choose validated functions instead of writing code.

3 Publish to preview

One action generates domains, logs and outputs.

4 Promote and run

The study lead promotes; the Engine runs it.

Intent, implementation and datasets stay linked in one traceable chain.

THE SEAM

Where the platform ends and this talk begins

1 Machine Understandable Specification

- Human-readable and machine-executable JSON
- Each TargetList entry names the function and its parameters
- A Functions array records version and execution sequence

2 The function contract

- A versioned Python module with a fixed signature
- A metadata.json declaring id, version, domains and files
- Exact name agreement across folder, module and metadata

One correct function serves every study — and a defect spreads as easily as an improvement.

THE DEVELOPMENT CHALLENGE

A library function must be correct for every study

Clinical correctness

- CDISC-conformant logic
- Spec approved before coding
- Defined outcome for misses
- Edge cases no study shows

The engine contract

- No exception may escape
- Return only the target
- No identifiers in logs

The evidence burden

- Every branch exercised
- Expected results from spec
- Three independent reviews

A function may take several rounds of spec revision and rework before release.

THE GOVERNED LIFECYCLE

Four accountable roles, defined before AI arrived

1. **Developer**
   Authors the specification, code and tests; achieves the quality gates.

2. **Reviewer**
   Peer-reviews code; verifies the diff contains only intended files.

3. **SME**
   Approves the spec; authors independent expected data; reviews logic.

4. **Release Owner**
   Approves and merges pull requests; owns promotion and sign-off.

In this talk, “developer” means a library developer — not a study programmer.

THE DEVELOPMENT WORKFLOW

Three standards every function must satisfy

Specification first

- SME-approved before any code
- Logic, sources and targets
- Reference files, edge cases
- Approved deviations

Structured logging

- Emitted via logging extra
- Queryable without parsing
- No exception may escape
- Omitting the log loses data

Test coverage

- Valid, empty, missing cols
- Null and blank values
- Reference hits and misses
- Duplicates, odd categories
- Every path, every except

Approved intent, observable behavior, and evidence across every code path.

VALIDATION GATES

No function deploys until every gate passes

1 Developer testing

Designs cases, confirms outcomes, signs as tester.

2 Code quality gates

black, flake8 and pylint clean; metadata checked.

3 Logging compliance

Self-reviewed against the logging standard.

4 Integration test

In UAT, on mock study data, reproducing expected output.

Any failure returns the function to development — not to the gate that caught it.

THE MULTI-AGENT FRAMEWORK

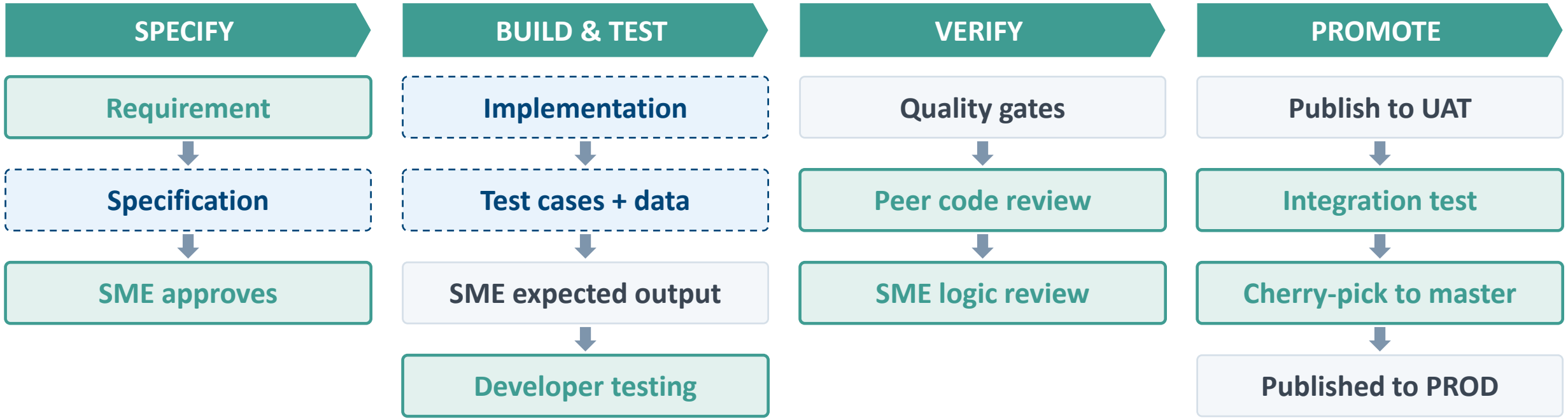
An agent never spans a governance boundary

1. **Requirements Interpreter** Execution plan, edge cases, questions → Developer
2. **Spec Writer** Formal specification → SME, before any code
3. **Python Programmer** Function module and metadata.json → Developer, Reviewer
4. **Test Plan Writer** Objectives and expected results from the spec → Developer
5. **Test Data Generator** Input data reaching every branch → Developer

Expected results never come from the code under test. Nothing an agent produces is self-approving.

WHERE AI SITS IN THE LIFECYCLE

Agents draft at three points — none of them is a gate



Any gate failure returns the function to development — never to the gate that caught it.

THE MASTER PROMPT

Two kinds of rule, and the defaults that must be negated

Mechanically checkable

- Naming identities bound to declared metadata
- Version folder matching the declared version
- Reference files loaded matching those declared

Semantically checkable only by a human

- Does the derivation implement the CDISC rule?
- Is edge-case handling clinically appropriate?
- On a reference miss: blank, or propagate?

Model defaults that must be explicitly negated

1. **Logging the record processed**
   Ensure no inadvertent unblinding occurs.
2. **Raising on an unrecoverable condition**
   An escaped exception aborts the run
3. **Iterating row-wise by habit**
   Vectorized alternatives are preferred
4. **Leaving intermediate columns behind**
   The contract returns only the target

An agent produces both reliably. Only the first can be confirmed by a script.

HOW CONTROL IS LAYERED

Three tiers, and the authority that stays with people

1 Tier 1 — Standards

- In the master prompt
- Nothing left implicit
- Model defaults negated

2 Tier 2 — Developer-led

- Every artifact reviewed
- Plans, code, tests, logs
- Plus mechanical gates

3 Tier 3 — Independent

- SME logic review
- UAT integration testing
- Recorded in the tracker

Authority reserved to humans

- Approve a specification (SME)
- Advance a function past any gate
- Merge and promote (Release Owner)

Regulatory posture

- Every artifact has an accountable owner
- Prompts are versioned alongside the code
- Agents never see production study data

Every “Definition of Done” checklist is completed by a named human.

THE RAG-BASED ASSISTANT

Four capabilities for everyone who uses the library

1 **Function Assistant**

Answers from the library's own source, docstrings and metadata.

2 **Standards Knowledge Base**

Answers SDTM questions from curated guidance, with citations.

3 **Spec Drafter**

Clarifies underspecified requirements, then drafts for approval.

4 **Parameter Builder**

Turns intent into structured parameters. Reduces errors, not eliminates.

The agents serve developers building functions. The assistant serves everyone who uses them.

HOW GROUNDING WORKS

Hybrid retrieval, a Git-tracked corpus, and honest limits

Hybrid retrieval

- Vector search for behaviour
- Keyword search for names
- Deduplicated and ranked

Corpus and sync

- Library source plus guidance
- Git-backed synchronization
- Chat over a Python API

Known limitations

- Retrieval not benchmarked
- Citations aid, not guarantee
- Unsynced work is invisible

An advisory aid, not a validated component. It satisfies no validation gate.

RESULTS

Gains in consistency, robustness and speed

100+
versioned functions in the library

130+
releases of the functions

3–6 → 1–3
weeks to validated completion

6
coverage categories, every branch

Consistency
Structure is settled, so review goes to logic.

Test robustness
Synthetic data builds the condition on demand.

Cycle time
Drafting fell to minutes. Review did not.

Access to expertise
Routine questions answered with citations.

WHAT GENERALIZES

The governance model was not relaxed for AI

1. **Build in what a machine can check; review what it cannot**
   Naming and contract rules can be checked. Clinical logic cannot.

2. **Explicitly negate model defaults**
   The most dangerous generated code is idiomatic everywhere else.

3. **Specification-first keeps test evidence anchored**
   Expected outputs come from a human-approved reference.

4. **Expect the writing to shorten, not the reviewing**
   Governance sets the remaining cycle length — and its floor.

5. **Grounded retrieval widens access without widening authority**
   Study programmers resolve routine questions without SME time.

Questions?



srinivas.malipeddi1@merck.com
proma.debnath@merck.com