

Strategic Use of GenAI in Statistical Programming

A White Paper for Professionals in the Pharmaceutical Industry

PHUSE Special Project

phuse.global

Contents

Executive Summary 1

Introduction 1

Framework for Adoption: Principles, Scope, Controls and Roadmap 1

1. Scope and Design Principles 2
2. Where GenAI Adds Value in Statistical Programming 2
3. Adoption Controls: HITL, Risk-Based Governance, Guardrails, and Validation Philosophy 3
4. Regulatory Alignment and Inspection Readiness 4
5. Practical Adoption Roadmap: Start Small, Scale Responsibly 5

Worked Example: GenAI-Assisted ADaM Specification from a SAP 5

Future Considerations 6

Conclusion 8

Appendices: Supporting Detail and Implementation Guidance 9

- How to Use the Appendices 9
- Appendix A – Conceptual and Foundational Clarifications 9
- Appendix A1. The Core Distinction: Where Learning Happens vs. Where GenAI Is Used 9

Appendix B – GenAI Use Cases in Statistical Programming 10

- Appendix B1. Use Case Scope, Assumptions and Design Principles 10
- Appendix B2. Use Case Categorisation Across the Programming Lifecycle 10
- Appendix B3. Planning and Prioritisation Lenses 11
- Appendix B4. Detailed Use Case Catalogue 11
- Appendix B5. Positioning and Adoption Guidance 12

Appendix C – Technical Enablement and Architecture 13

- Appendix C1. AI Enablement and Implementation Considerations 13
- Appendix C2. Technical Platforms, Reference Architectures and Key Technology Considerations 13
- Appendix C3. Context Management as a Critical Enabler 14

Appendix D – Human, Organisational and Operating Model 15

- Appendix D1. Human-in-the-Loop (HITL) Framework 15
- Appendix D2. Organisational Considerations 16
- Appendix D3. Roles and Responsibilities 16
- Appendix D4. Operating Procedures (SOPLite) 16

Appendix E – Risk, Validation, Privacy and Ethics 17

- Appendix E1. Risk-Based Controls and Risk Management 17
- Appendix E2. Data Privacy, Security & Compliance 18
- Appendix E3. Validation and Quality Management (GxP Focus) 19
- Appendix E4. Ethics and Responsible AI Considerations 19

Appendix F – Regulatory Alignment & External Frameworks 19

- Appendix F1. Regulatory Alignment Overview 19
- Appendix F2. Key Regulatory and Standards References 19

Resources 22

Acronym List 23

Revision History

Version	Date	Summary
1.0	September 2026	Initial version

Authors: Sharad Chhetri (Takeda), Chadha Taranbirsingh (Astellas), Daniel Boisvert (Biogen), Jeffrey Meyers (Regeneron), Ryan Yu (Regeneron), Rashad Rasul (BMS), Gurubaran Veeravel (Merck), Ramesh Ayyappath (Otsuka), Matt Smith (Pfizer), Jiayi Qin (Novo Nordisk), Bhavin Vala (Novartis), Paul Bukowiec (Takeda), Yang Liu (AbbVie), Liming Jin (Amgen), Thiago Figueiredo (Merck Group), Harpreet Pruthi (Sanofi), Daniel Schramek (GSK), Mark Reynolds (AstraZeneca), Jakob Kaiser (Bayer), Shereen A. Khwajazada (PHUSE & Lotus CRO), Sam Parmar (Pfizer), Clark Peirce (Thermo Fisher Scientific), N. Ward (Amgen), Tadeusz Lewandowski (Johnson & Johnson), Mark Bynens (Johnson & Johnson).

Executive Summary

The opportunity. Generative AI is reshaping how clinical evidence is produced. Statistical programming organisations that adopt GenAI within disciplined, risk-based frameworks may improve efficiency gains in specification authoring, code development and quality assurance, while simultaneously raising output quality, consistency and scalability. The magnitude of these benefits will depend on organisational maturity, governance, and effective human oversight. As sponsors, CROs and regulators continue to evaluate and adopt AI-enabled capabilities. Organisations that fail to develop an AI strategy risk falling behind in productivity, innovation and talent attraction.

The challenge. GenAI is probabilistic, non-deterministic, and prone to confident-sounding errors. It cannot be validated in the traditional GxP sense. Its outputs must be governed through the same rigour applied to human-authored deliverables, with additional controls for AI-specific failure modes including hallucination, hidden assumptions and inappropriate defaults. Furthermore, as organisations increasingly adopt AI-driven development approaches, operational efficiency and cost management can become initial challenges. Token consumption, model utilisation and infrastructure costs must be actively monitored and optimised to balance innovation, performance and scalability with financial sustainability.[1] An additional challenge is workforce capability and role evolution. As AI becomes integrated into statistical programming workflows, there is a risk that programmers gain less hands-on experience developing the technical and domain expertise needed to critically evaluate outputs. Although roles may evolve towards AI-enabled development and oversight, strong knowledge of programming, data standards, clinical trials and regulatory expectations will remain essential to ensure quality, compliance and scientific integrity.

This framework. This paper provides a principle-based, inspection-ready framework for adopting GenAI across the end-to-end statistical programming lifecycle. It establishes five foundational principles: (1) GenAI augments human work; it does not replace it (2) all outputs must be static, persisted and reviewable (3) adoption must be risk-based and use-case-driven (4) data privacy and confidentiality are non-negotiable (5) validation targets the surrounding workflow, not the model itself.

The roadmap. We recommend a phased adoption approach: start with low-risk documentation and QA use cases to build governance muscle and practical experience, then expand into core data and output generation as standards, context management and team capability mature. Organisations should invest in AI-ready metadata and context infrastructure as a critical enabler of reliable, scalable GenAI performance.

The trajectory. The field is moving fast. In December 2025, the FDA deployed agentic AI capabilities across the agency. Multimodal models, autonomous agents and AI-native development environments are advancing rapidly. This paper is designed to be durable, anchored in regulatory principles that will outlast any specific technology generation, while preparing organisations for a future in which GenAI is not an optional add-on but a core capability of the statistical programming function.

Introduction

In December 2025, the U.S. Food and Drug Administration deployed agentic AI capabilities for all agency staff – reviewers, scientists and investigators – to accelerate drug evaluation. The same month, the EMA and the FDA jointly published guiding principles for AI in drug development, signalling that regulators are not waiting for industry to move first. They are moving themselves.

Statistical programming sits at the centre of this transformation. Every dataset, specification, table, figure and listing that supports a regulatory submission passes through statistical programming workflows. The quality, speed and scalability of these workflows directly affect how fast safe and effective treatments reach patients. Generative AI offers a step-change in all three dimensions – but only when deployed within disciplined, risk-based and inspection-ready frameworks.

This paper provides that framework. Developed by a cross-industry Working Group under PHUSE, it establishes the principles, governance structures, risk controls and practical adoption guidance needed to deploy GenAI responsibly across the statistical programming lifecycle. The framework is intentionally technology- and organisation-agnostic, designed to remain valid as GenAI capabilities evolve.

We make three core claims. First, GenAI is not merely a productivity tool that does existing work faster; it is a strategic capability that enables work previously impossible at scale, cross-study consistency analysis, real-time specification-to-output traceability, and automated regulatory intelligence. Second, realising this potential requires treating GenAI adoption as an organisational transformation, not a technology deployment: governance, skills, context management and workflow integration matter as much as model selection. Third, the regulatory landscape already supports responsible GenAI adoption; the principles of human oversight, traceability, reproducibility and risk-based governance are well established and directly applicable.

What this paper does not do is prescribe a single path. Organisations vary in maturity, risk tolerance, regulatory context and investment priorities. Instead, we provide a principled framework, a structured use-case catalogue with value and complexity assessments, and a phased adoption roadmap that each organisation can calibrate to its own context. The goal is clear: enable statistical programming organisations to adopt GenAI with confidence, at pace, and without compromising the quality and rigour that regulators and patients depend on.

Framework for Adoption: Principles, Scope, Controls and Roadmap

Purpose of this framework for end users

This framework is intended to help statistical programming teams, reviewers, quality partners and leaders decide where, when and how GenAI can be used responsibly. It translates the broader principles of human oversight, risk-based governance, traceability, privacy, and inspection readiness into practical guidance that end users can apply when selecting use cases,

generating artifacts, reviewing outputs and documenting decisions.

The framework is organised as a practical adoption guide: it clarifies what is in scope, identifies where GenAI adds value, defines the controls needed for different levels of risk, explains how to remain aligned with regulatory expectations, and provides a phased roadmap for starting small and scaling with confidence. Its purpose is not to prescribe one universal implementation model, but to give end users a clear, defensible way to apply GenAI while maintaining quality, accountability and trust.

1. Scope and Design Principles

What is in scope, what is not, and why

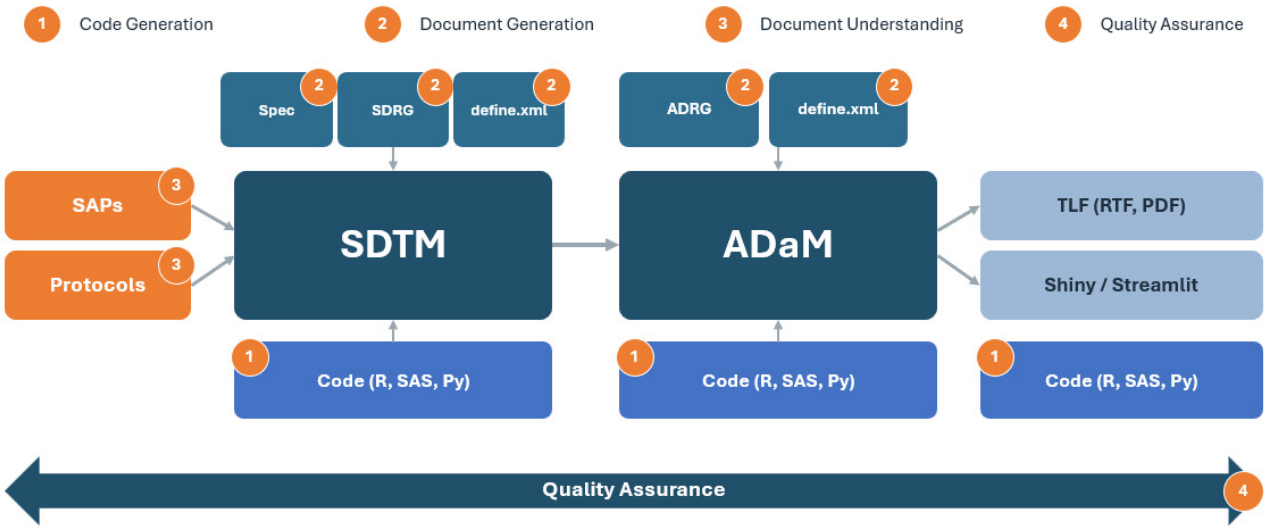
This framework applies to GenAI-assisted activities across the end-to-end statistical programming value chain, including programming specifications, SDTM and ADaM datasets, tables, figures, listings, validation artifacts, reviewer guides and supporting submission documentation. GenAI is positioned as a drafting, review and support capability within existing human-led workflows. It does not create a parallel process, replace established SOPs, or transfer accountability away from qualified professionals.

The design principles are intentionally simple and durable. GenAI augments human expertise rather than replacing it; outputs used in regulated workflows must be static, persisted, versioned and reviewable; adoption must be driven by intended use and risk; sensitive data must only be handled in approved environments; and autonomous or in-use learning is out of scope for regulated or confirmatory analyses. (See Appendix A1 for the learning distinction.)

2. Where GenAI Adds Value in Statistical Programming

Condensed, value-focused use cases

GenAI delivers the greatest value when applied to well-bounded, reviewable tasks across the full statistical programming lifecycle rather than being confined to isolated code generation. Priority use cases include drafting specifications and study documentation, extracting structured information from protocols and SAPs, supporting CDISC mapping and discrepancy detection, generating or refactoring code, assisting with TFL development, explaining code logic, generating unit tests, and checking consistency across specifications, code, datasets and outputs. (See Appendix B4 for the detailed use case catalogue.)



(Biometrics Process Traditional Workflow)

Use cases should be prioritised by expected value, complexity, operational readiness and risk. Early value is strongest where GenAI reduces repetitive authoring, improves consistency or identifies ambiguity before it causes downstream rework. Higher-impact use cases, such as dataset generation or submission-relevant output production, should follow only after standards, metadata, governance and team capability are sufficiently mature. (See Appendix B4 for prioritisation examples.)

Directional benefit and cost quantification

Leadership funding decisions should be supported by directional quantification, even when exact ROI cannot yet be proven. For each priority GenAI use case, teams should estimate both expected benefits and total cost of ownership, explicitly documenting assumptions, confidence level, and how benefits will be realised. External guidance from Gartner emphasises assessing both business value and cost for GenAI use cases, including the need to translate productivity gains into financial or operational outcomes and to continuously monitor value and cost as adoption scales.[23]

Benefit estimates should include time saved, rework avoided, cycle-time reduction, earlier issue detection, improved consistency, quality improvement, and capacity released for higher-value work. Cost estimates should include licences or API charges, token consumption, infrastructure, integration and context management, training and change management, governance, QA, validation evidence, monitoring, support and optimisation. Deloitte's work on AI token economics highlights that AI costs can be usage-driven, nonlinear and volatile, requiring active monitoring, forecasting, and FinOps-style governance rather than traditional static software cost models.[24]

A simple directional calculation can be used as a starting point: benefit value = volume × time saved per unit × fully loaded cost rate × adoption rate × realisation factor. Net value = benefit value − total cost of ownership. For example, if GenAI supports 100 programming specifications per year, saves 6 hours per specification, uses a fully loaded rate of €100 per hour, reaches 70% adoption, and realises 60% of the theoretical time savings as usable capacity, the directional annual productivity value is 100 × 6 × €100 × 0.70 × 0.60 = €25,200. This value should then be compared against annual TCO, including tooling, usage, support, quality, and governance costs.

Measurement should be built into pilots and scaled deployments from the start. McKinsey recommends linking technical performance, user adoption, operational change and financial impact, using baselines, control groups or staggered rollout where feasible to strengthen attribution.[25] KPMG's opportunity-sizing work also supports a workflow- and role-level approach to estimating GenAI value, noting that value capture depends on adoption, behavioural change, data readiness and trust.[26]

- **Improved efficiency:** Automates repetitive and manual tasks and supports standardisation, significantly reducing turnaround time and operational effort in otherwise highly accurate but resource-intensive workflows.
- **Quality support:** Assists with error detection, consistency checks and cross-artifact comparisons, helping identify potential issues earlier.
- **Increased scalability:** Enables rapid scaling of programming and validation activities, supporting delivery against demanding timelines without a linear increase in staffing.
- **Greater capacity for innovation:** Offloads repetitive tasks, allowing expert resources to focus on higher-value analytical, interpretive and strategic work rather than routine execution.

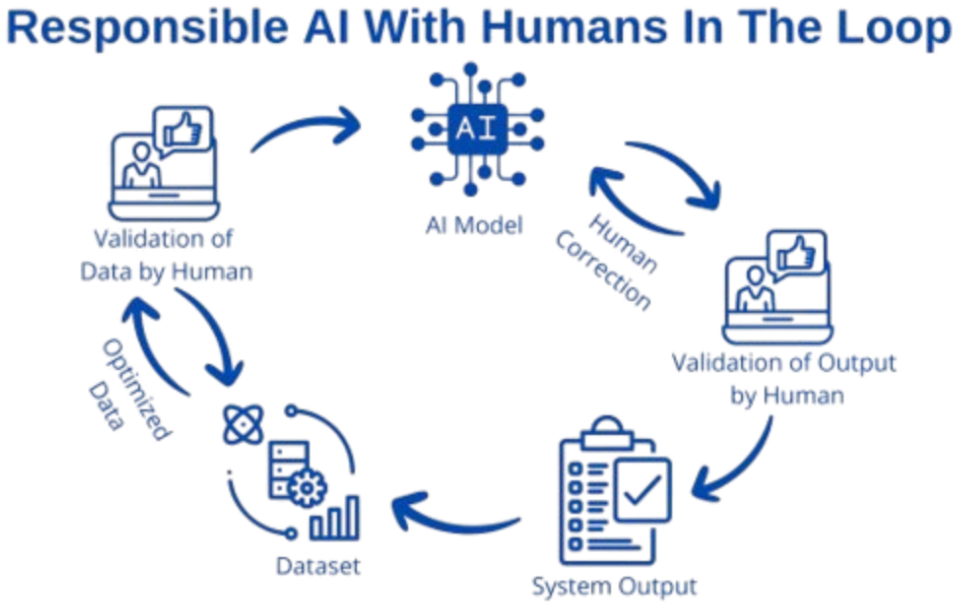
3. Adoption Controls: HITL, Risk-Based Governance, Guardrails, and Validation Philosophy

Human accountability, proportional controls and workflow-based validation

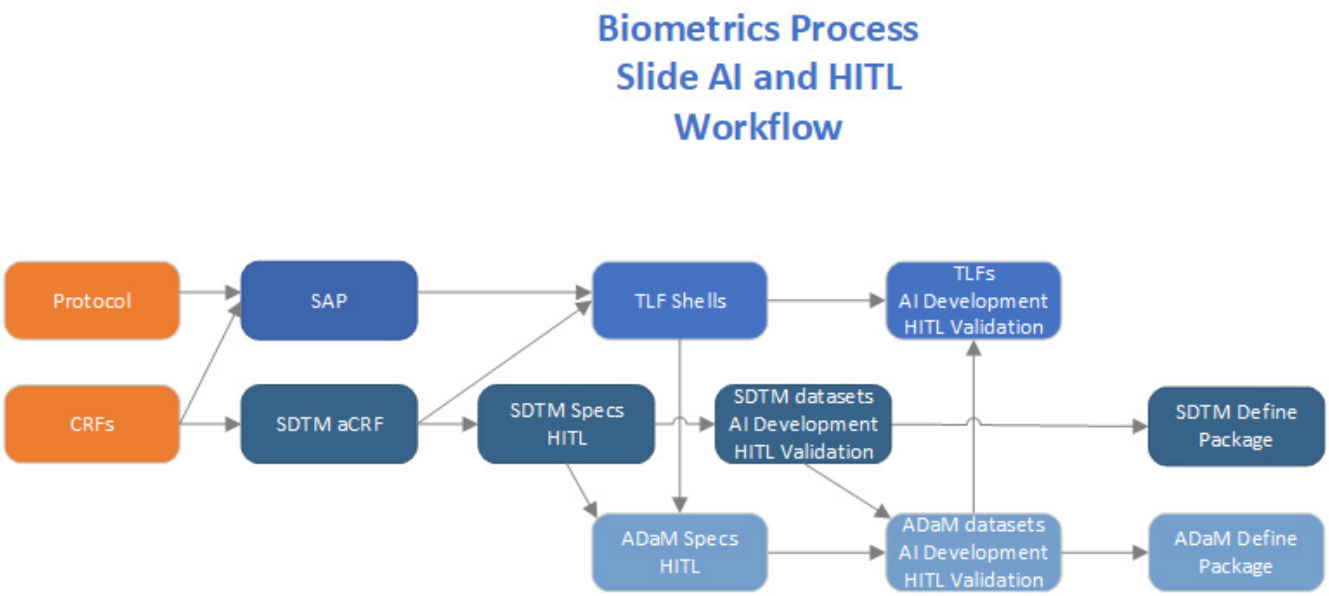
Human-in-the-loop review is the central control. GenAI may draft, suggest, compare, summarise or explain, but qualified humans remain accountable for analytical correctness, methodological appropriateness, regulatory fitness and final approval. Reviewers must assess the artifact against specifications, standards, tests and scientific intent; AI-generated rationale should not be considered evidence.

Controls should scale with risk. Low-risk uses may rely on standard review and version control. Moderate-risk uses require defined acceptance criteria, traceability, and independent review. High-impact or submission-relevant use cases require enhanced evidence, explicit context of use, robust testing and documented signoff. Guardrails should define permitted use cases, prohibited uses, approved environments, input and output restrictions, review checkpoints, documentation expectations and escalation paths.

The validation philosophy is workflow- and control-focused. The general LLM itself is not treated as a traditionally validated GxP system; instead, validation focuses on the surrounding process, tools, environments, access controls, versioning, logging, review procedures, testing practices, change control and retained evidence. GenAI-assisted artifacts should flow through the same SDLC, QC, validation and approval pathways as human-authored artifacts.



[22] Anderson, A. (2023, 6 September). *Responsible AI use with Human-in-the-loop (HITL).*



Biometrics Process AI with HITL workflow

4. Regulatory Alignment and Inspection Readiness

Why this approach is inspection-ready

This adoption model aligns with principle-based regulatory expectations for AI in drug development: human oversight, proportional risk management, transparency, reproducibility, data integrity and clear accountability. GenAI-assisted deliverables remain inspection-ready when they are static, versioned, traceable to inputs and decisions, independently reviewed, and archived through existing quality and records processes.

The practical implication is that GenAI should not require a separate compliance universe. Instead, it should be embedded into existing SDLC, QC, validation, data privacy and audit frameworks, with targeted additions for GenAI-specific risks such as hallucination, hidden assumptions, nondeterminism, data leakage, inappropriate defaults and unsupported references. The human user remains accountable for the final deliverable and is responsible for ensuring it meets all applicable compliance requirements.

In this context, the framework aligns with widely cited regulatory principles for AI in regulated environments, including:

Principle	Summary
Human-centric by design	AI development aligns with ethical, human-centred values.
Risk-based approach	AI follows proportionate validation, mitigation and oversight based on context and model risk.
Adherence to standards	AI complies with legal, ethical, technical, scientific, cybersecurity and regulatory standards (including GxP).
Clear context of use	AI has a well-defined role and purpose for why it is being used.
Multidisciplinary expertise	Experts in AI and its application domain are integrated throughout the technology lifecycle.
Data governance and documentation	Data provenance, processing and decisions are traceable, verifiable and aligned with GxP; privacy is protected.
Model design and development practices	AI uses best practices in design, engineering and data quality to ensure transparency, reliability and robustness.
Risk-based performance assessment	Performance is evaluated using fit-for-use data, appropriate metrics and validation of predictive performance.
Life cycle management	AI undergoes continuous monitoring, issue management, and periodic re-evaluation (e.g. for data drift).
Clear, essential information	Plain-language communication ensures users understand context, performance, limitations, data and updates.

(Adapted from the U.S. Food and Drug Administration, 2026) [5]

5. Practical Adoption Roadmap: Start Small, Scale Responsibly

Start small, learn safely, and expand with maturity

Adoption should begin with a foundation layer: approved tools and environments, privacy and security controls, defined risk tiers, role-based accountability, training, logging expectations and clear guidance on permitted and prohibited uses. This foundation allows early experimentation without creating unmanaged risk.

The first wave should focus on low-risk documentation, review, and QA use cases such as drafting specifications, summarising requirements, explaining code, generating comments, supporting unit test creation and checking consistency. These use cases build practical experience, trust, and reviewer skill while keeping regulatory exposure contained.

The second wave should strengthen standards, templates, metadata, reusable context, and integration with existing statistical computing environments. As organisational maturity grows, GenAI can then be embedded into development and review workflows, with outputs flowing through the same version control, testing, QC and approval pathways as human-authored artifacts. (See Appendix C3 for context management approaches.)

Only after these foundations are in place should organisations expand into higher-impact activities such as SDTM and ADaM generation, CDISC validation support, TFL generation and cross-output consistency checks. At that stage, stronger HITL controls, documented acceptance criteria, enhanced traceability and risk-based validation evidence is expected. This phased approach preserves rigour, accountability and inspection readiness while allowing GenAI value to scale responsibly.

Worked Example: GenAI-Assisted ADaM Specification from a SAP

Scenario. A statistical programmer is tasked with drafting the ADaM specification for ADTTE (time-to-event analysis dataset) based on an approved SAP and protocol. The team has adopted GenAI for specification authoring under their low-to-moderate risk tier.

Step 1: Context Assembly

The programmer assembles the input context for the GenAI system: the relevant SAP sections (endpoints, analysis populations, censoring rules), the organisation's study-specific SDTM metadata (spec and datasets), the organisation's ADaM template and naming conventions, the CDISC ADaM Implementation Guide v1.3 extract, and the study protocol's visit schedule and treatment arms. This context is provided via the organisation's approved context management infrastructure (e.g. MCP server connected to the standards repository). (See Appendix C3 for context management guidance.)

Step 2: Prompt and Generation

Using an approved prompt template, the programmer instructs the GenAI system: *"Based on the attached SAP sections and ADaM template, draft a complete ADaM specification for ADTTE. Include all variable-level metadata, derivation logic for AVAL, STARTDT, ADT, CNSR, and PARAM. Apply the censoring rules specified in SAP Section 4.3. Flag any ambiguities in the SAP that require clarification."*

The GenAI system generates a complete draft specification, including variable definitions, derivation logic, value-level metadata and three flagged ambiguities where the SAP language could support multiple interpretations.

Step 3: Human Review and Refinement

The programmer reviews the draft ADaM specification against the SAP, protocol and ADaM IG. In this example, the review identifies one derivation where the GenAI applied a default censoring rule not specified in the SAP (corrected), one variable naming inconsistency with the organisation's standard (corrected), and the three flagged ambiguities (escalated to the statistician for clarification). The programmer documents all modifications in the version control system with clear annotation of what was AI-generated versus human-modified.

Step 4: Independent Review and Approval

An independent reviewer evaluates the ADaM specification against the SAP, verifying derivation logic, completeness, and CDISC compliance. The reviewer assesses the specification on its merits – not relying on the AI's reasoning or 'chain of thought' and confirms the flagged ambiguities have been resolved. The approved specification is versioned and archived with full provenance (prompt, generated output, human modifications, review decisions).

Outcome

The GenAI-Assisted ADaM specification from the SAP example demonstrated that GenAI can produce a substantially complete, high-quality first draft of an ADaM specification, creating a stronger starting point. This allowed review effort to focus on targeted refinement, confirmation of derivation logic, and resolution of exceptions rather than broad initial authoring. The AI's identification of three SAP ambiguities also surfaced potential issues early, supporting fewer downstream review cycles and reducing the risk of rework later in the programming cycle. After human review, correction and independent approval, the final specification met the same quality standard as a human-authored specification and remained fully inspection-ready.

Key Observations from This Example

Principle	How It Applied
Human accountability	Programmer reviewed and corrected all ADaM specifications; reviewer independently verified; statistician resolved ambiguities
Static, persisted outputs	Generated specs were saved, versioned and archived with full provenance trail
Risk-based controls	Low-to-moderate risk tier: standard QC process, no additional validation required beyond existing SOPs
Data privacy	SAP and protocol provided through approved, access-controlled context infrastructure; no PHI involved
Workflow validation	Review, testing and approval processes unchanged from human-authored workflow; GenAI involvement documented
Value beyond speed	AI surfaced three SAP ambiguities that might have been missed, preventing downstream rework

Future Considerations

Where This Is Going

The GenAI capabilities described in this paper represent the starting point, not the destination. Several developments are likely to reshape statistical programming within the next five years. Organisations that understand these trajectories can position themselves to lead.

From Assistants to Agents

Today's GenAI operates primarily as a drafting assistant: a human prompts, the model generates, the human reviews. The next phase is agentic AI – systems that autonomously execute multi-step workflows, use tools, consult references and produce complete work packages for human review. The FDA's December 2025 agentic AI deployment signals that regulators view this trajectory as viable and valuable. Within statistical programming, agentic workflows could orchestrate end-to-end processes from SAP to table shells to ADaM specifications to TFL code, with human review concentrated at defined checkpoints rather than distributed across every micro-task. This shift will require stronger governance, clearer acceptance criteria, and more robust sandboxing – but it will also fundamentally change the ratio of review time to authoring time.

Multimodal AI and Document Intelligence

Current use cases focus primarily on text and code. Emerging multimodal models can process images, tables, annotated PDFs and structured forms natively. For statistical programming, this means GenAI that can read annotated CRFs, interpret complex table layouts in legacy SAPs, extract information from scanned regulatory correspondence and validate TFL formatting against visual specifications – tasks that currently require significant manual effort. Organisations should begin assessing the potential and which workflows involve unstructured visual inputs and consider how multimodal capabilities could reduce manual interpretation steps.

Standards as AI-Native Infrastructure

The convergence of CDISC standards with LLM-consumable metadata formats represents one of the most consequential near-term developments. When Define-XML, controlled terminology, and standard code libraries are structured as machine-readable context that GenAI systems can reliably consume, the quality and consistency of AI-generated outputs will improve dramatically. This is not a future aspiration – it is an active area of development and aligns with broader industry initiatives such as CDISC's Unified Study Definitions Model (USDM) and Digital Data Flow, which are advancing machine-readable study definitions and greater interoperability across the clinical development lifecycle. Organisations should prioritise making their standards, specifications and metadata 'AI-ready': structured, versioned, governed and accessible through standardised interfaces such as the Model Context Protocol (MCP). This investment pays dividends regardless of which GenAI tools are adopted.

Workforce Transformation

GenAI will reshape the statistical programming workforce. Demand for routine, manual programming tasks – boilerplate SDTM mappings, repetitive TFL code, standard dataset derivations – will decline as GenAI automates initial drafting. Demand for senior programmers who can architect GenAI workflows, critically evaluate AI-generated outputs, exercise scientific judgement on edge cases, and design robust validation strategies will increase. Prompt engineering will not become a standalone role in statistical programming; rather, it will become an embedded skill for domain experts, similar to how version control or code review became standard competencies over the past two decades. Organisations must invest now in upskilling programmers who develop these capabilities across their teams.

CRO and Sponsor Dynamics

GenAI will alter the outsourcing model for statistical programming. Sponsors will increasingly expect CROs to demonstrate GenAI capability, governance maturity and

measurable efficiency gains. CROs that can deliver GenAI-augmented programming with transparent quality controls will command stronger positioning. Conversely, CROs that cannot demonstrate AI readiness may face margin pressure as manual-intensive delivery models become less competitive. This dynamic creates both urgency and opportunity for organisations across the ecosystem.

Regulatory Evolution

Regulatory expectations for AI transparency and governance will intensify. The EU AI Act imposes obligations on general-purpose AI providers that directly affect the tools statistical programmers use. The FDA's draft credibility framework distinguishes between operational efficiency uses and analytical uses that affect the reliability of results. Organisations should anticipate increasing expectations for documentation of GenAI use, evidence of risk management, and clarity on how AI-related risks are controlled. The principle-based approach in this paper is designed to be resilient to these developments, but organisations should actively monitor regulatory evolution and adapt their governance accordingly.

The Cost of Inaction

GenAI adoption is not cost-free: enterprise API agreements, platform licensing, compute infrastructure, token consumption, training programmers, and governance overhead all require investment. Token costs can grow quickly as usage scales across teams, studies, large context windows, repeated prompt iterations, and agentic workflows; they therefore need active monitoring, budgeting and optimisation rather than being treated as an invisible utility cost. However, the cost of inaction is higher. Organisations that delay adoption risk falling behind competitors who deliver faster and at lower cost, losing talent to more AI-forward employers, and being unprepared when regulatory expectations for AI transparency crystallise. The phased adoption roadmap in this paper is designed to deliver value early while containing risk, enabling organisations to build the business case for continued investment through demonstrated results.

For a contract research organisation (CRO), integrating AI into workflows requires additional considerations beyond standard business AI governance because of the highly regulated nature of clinical research and the expectations of sponsors and regulatory agencies. AI implementation should prioritise quality, compliance, data integrity, patient safety and transparency.

Here are considerations that are uniquely or disproportionately important for CROs:

CRO-Specific Consideration	Why It Matters
Sponsor-Specific AI Requirements	Sponsors may have different AI policies. A CRO needs a process to identify, document and comply with each sponsor's requirements rather than applying one organisation-wide approach.
Contractual Obligations	Statements of Work (SOWs), Master Service Agreements (MSAs) and Confidential Disclosure Agreements (CDAs) may restrict AI use or require sponsor notification or approval before AI is used in deliverables.
Multi-Sponsor Environment	CROs simultaneously support many sponsors. AI workflows must prevent any possibility of sponsor information being used across projects or influencing outputs for another client. Strong data segregation is essential.
Service Consistency	AI should improve efficiency while ensuring deliverables remain consistent across different project teams, therapeutic areas, and clients. Standardised prompts, templates and review procedures become operational assets.
Client Transparency	Decide when and how AI use should be disclosed. Some sponsors may expect disclosure in proposals, governance meetings, quality agreements, or study documentation.
Competitive Differentiation	AI can be used to improve turnaround time, quality and consistency, but it must not compromise scientific rigour. The focus is on enhancing service delivery rather than creating proprietary drug-development insights.
Resource Allocation	AI changes staffing models by automating repetitive work. CROs should evaluate how to redeploy staff towards higher-value activities such as client communication, strategy, and quality oversight.
Cross-Functional Standardisation	AI governance should span biostatistics, statistical programming, data management, medical writing, regulatory affairs and clinical operations to avoid fragmented practices.
Deliverable Ownership	Although AI may assist in preparing a statistical analysis plan, TFL shells, reviewer guides, or CSR sections, the CRO remains accountable for the quality of every client deliverable. Responsibility cannot be delegated to AI.
Client Audits	Sponsors increasingly audit vendors' AI governance. CROs should be prepared to demonstrate approved tools, policies, training, validation activities, and quality oversight during qualification and routine audits.
Vendor Qualification of AI Tools	AI providers become part of the CRO's vendor ecosystem and should be assessed alongside other critical vendors for security, reliability and support.
Knowledge Management	AI can help capture institutional knowledge, SOP guidance, programming standards and lessons learned across studies while ensuring sponsor-specific information remains isolated.
Business Continuity	CROs need contingency plans if an AI platform changes, becomes unavailable, or produces inconsistent outputs, since disruptions could affect multiple sponsor projects simultaneously.

A CRO-Centric Principle

A useful way to frame AI governance for a CRO is:

AI should strengthen the CRO's ability to deliver consistent, high-quality, sponsor-specific services while protecting client confidentiality, maintaining regulatory compliance, and preserving scientific accountability.

This differs from a pharmaceutical company's objective, which is primarily focused on accelerating drug discovery and development. For a CRO, AI is first and foremost a service delivery capability. Success depends not only on regulatory compliance but also on meeting diverse sponsor expectations, protecting confidential information across multiple clients, and maintaining the trust that underpins long-term partnerships.

Conclusion

Act now: turn GenAI from experimentation into controlled capability

GenAI is no longer a distant innovation topic for statistical programming; it is a capability that organisations should start governing, testing and scaling now. The question is not whether GenAI can support statistical programming, but whether it can be introduced in a way that preserves scientific judgement, data integrity, human accountability and inspection readiness. The answer is yes, provided organisations move deliberately: start with bounded use cases, apply risk-based controls, keep humans accountable, and make every GenAI-assisted output traceable, reviewable and defensible.

The immediate next steps depend on organisational maturity, but every organisation can take practical action now:

- For organisations just starting: define an approved GenAI use policy, confirm which tools and environments may be used, and identify low-risk documentation or QA use cases for controlled pilots.
- For organisations running pilots: formalise risk tiers, prompt and output retention, and escalation paths for errors, hallucinations, privacy concerns, or unsupported outputs.
- For organisations ready to deploy and scale: standardise templates, metadata, reusable context, and acceptance criteria so GenAI can produce higher-quality, more consistent outputs across studies and teams.
- For mature organisations: integrate GenAI into existing SDLC, QC, validation, audit and records processes, with clear evidence of traceability, review decisions, version control and accountable human approval.
- For all organisations: establish a feedback loop to monitor output quality, capture lessons learned, update governance, and reassess use cases as technology, regulations and organisational capability evolve.

Used this way, GenAI becomes more than an efficiency tool.

AI is rapidly becoming a strategic differentiator across the clinical research industry. Organisations that proactively explore and implement responsible AI solutions today will not only improve operational performance but also shape the future of drug development. Whether you are a sponsor driving innovation or a CRO delivering execution, the time to begin your AI journey is now. Together, we can build a more efficient, data-driven and patient-centred future for clinical research.

Appendices: Supporting Detail and Implementation Guidance

How to Use the Appendices

The main body of this paper provides the core strategic framework and recommendations for responsible GenAI adoption in statistical programming. The appendices that follow are intended as supplemental reference material. They provide deeper explanations, use-case detail, technical considerations, governance examples and regulatory context for readers who want to explore specific topics in more depth.

These appendices should be used selectively and should not be interpreted as additional mandatory requirements beyond the core guidance in the main paper. They are written in a more explanatory and illustrative style, offering clarifications, examples, design rationales and implementation considerations that may help organisations adapt the framework to their own contexts.

Readers may consult the appendices based on their roles, maturities and areas of interest, for example using the use case catalogue for prioritisation, the technical appendices for architecture considerations, or the governance and operating model sections for implementation planning. Named contributors are retained for each appendix to acknowledge subject matter ownership and to facilitate follow-up discussions where clarification or deeper engagement may be helpful.

Appendix A – Conceptual and Foundational Clarifications

Appendix A1. The Core Distinction: Where Learning Happens vs. Where GenAI Is Used

The key to resolving this apparent tension is to clearly separate the three layers:

- Model learning and improvement (upstream, outside GxP workflows)
- Controlled deployment of a fixed model version with appropriate prompting and context (inside workflows)
- Human learning and process improvement (continuous, everywhere)

How GenAI Actually Learns (and Continues to Improve)

GenAI systems learn primarily through offline, controlled processes, not during live use in regulated workflows.

Model Training and Retraining (Offline)

- Large language models are trained or fine tuned on curated datasets outside clinical workflows.
- Improvements occur through:
  - New training data
  - Model architecture changes
  - Fine tuning or reinforcement learning
- These updates are versioned, documented, and released deliberately, similar to software upgrades.

This is where 'learning' is expected and encouraged.

Crucially:

- Learning happens before deployment, not during execution.
- A specific model version is frozen when used for regulated work.

Controlled Model Versioning (Deployment Discipline)

In regulated statistical programming:

- A specific model version is selected and pinned.
- Prompts, instructions, static context (e.g. AGENTS.md/ documentation) and dynamic context (e.g. tools, MCP) are controlled and versioned.[7][8][9][10][11][12]

While model context is inherently dynamic and may be refined over time to improve performance, such refinements are treated as controlled changes; in practice, teams may iteratively curate cumulative context artifacts (e.g. evolving AGENTS. md files) to capture institutional knowledge, recognising that the effectiveness of these approaches is still an active area of research and subject to ongoing evaluation.[13]

- Outputs are static and reviewable.

When the model improves:

- It is introduced as a new version.
- Subject to:
  - Risk assessment
  - Qualification/acceptance testing (fit-for-purpose)
  - Controlled rollout

This mirrors how SAS versions, R packages and validation tools are handled today.

The system learns between versions, not within a live analysis.

What 'Learning in Use' Means and Why It's Out of Scope

'Learning in use' does not mean:

- Using GenAI frequently
- Refining prompts
- Improving workflows over time.

It specifically refers to autonomous adaptation during execution, such as:

- The model updates its parameters based on:
  - Generated outputs
  - Execution results
  - User behaviour.
- The same input produces measurably different outputs because the model has changed itself. It is acknowledged that stochastic (non deterministic) models may produce different outputs for identical inputs, even when the model itself has not changed. For this reason, governance does not attempt to enforce identical outputs across runs. Rather, it focuses on preventing autonomous model adaptation during regulated execution by means of controlled model versioning, fixed execution boundaries, and human oversight.

Additional related considerations include:

- Hallucination drift, where the nature, frequency or severity of incorrect or fabricated outputs changes over time despite stable workflows. Appropriate mechanisms should exist to detect and monitor such patterns, for example through human review feedback, issue logging, periodic spot checks, and trend analysis.
- Differences in verifiability by artifact type: programming artifacts are generally easier to verify through execution,

testing, and comparison to specifications, whereas narrative documentation is more difficult to validate and more susceptible to issues such as fabricated references or unsupported statements, requiring heightened reviewer scrutiny.
- Behaviour that cannot be reliably reconstructed after the fact, due to evolving model behaviour or uncontrolled context changes.

Why is this a problem in regulated analyses?
Such behaviour:
- Breaks reproducibility (you can't re run and get the same result)
- Breaks traceability (you can't explain which logic was applied)
- Breaks auditability (you can't show regulators a stable process).

This is why regulators (EMA/FDA) explicitly reject learning systems in confirmatory settings.

The issue is not AI learning, but uncontrolled learning during regulated execution.

How Learning Happens Safely Inside the Organisation
Even while excluding learning in use, learning still happens continuously in three safe ways.

Human Learning
Humans learn from GenAI outputs:
- Better specifications
- Improved standards
- Refined prompts
- Better templates and metadata

These improvements are:
- Documented
- Reviewed
- Reused deliberately.

This is fully compliant and often the highest-value form of learning.

Workflow Learning (Process Improvement)
Organisations can:
- Analyse logged prompts and outputs
- Identify recurring errors or gaps
- Improve based on fit-for-purpose, measurable indicators, such as:
  - Context sources
  - Prompt templates
  - QA checks
  - Guardrails.

None of this requires the model to self-modify during execution.

Model Updates as Controlled Change
When GenAI capabilities improve:
- A new model version is introduced.
- Prior analyses remain reproducible.
- Change impact is assessed. A typical mechanism for this is 'evals'.[15]

This aligns perfectly with:
- Change control
- Validation principles
- Inspection expectations.

SMEs: (Thiago Figueiredo, Mark Bynens)

Appendix B – GenAI Use Cases in Statistical Programming

Appendix B1. Use Case Scope, Assumptions and Design Principles

This guidance focuses on the end-to-end statistical programming value chain as the foundation for the proposed GenAI framework. The value chain encompasses the creation of SDTM and ADaM datasets, associated programming deliverables, and analytical outputs, including tables, figures, listings and reports, that support clinical trial decision-making.

A core design assumption is that all GenAI-enabled steps produce static, persistent outputs to ensure reproducibility, traceability and inspection readiness in regulated workflows.

Within this framework, GenAI is positioned as an augmentation layer applied across existing, standard-driven statistical programming workflows. Its role is to reduce time and effort in the creation and pipelining of deliverables, while maintaining established practices for quality control, human review, and accountability.

Appendix B2. Use Case Categorisation Across the Programming Lifecycle

The GenAI use cases described are organised into six primary categories, aligned with the statistical programming lifecycle, from study planning through to submission and inspection readiness:
1. Code Generation
2. Programming Specification and Document Generation
3. Document Understanding
4. CDISC Data Generation and Validation
5. Output (Tables, Figures, Listings) Generation and Validation
6. Quality Assurance

Together, these categories represent a comprehensive view of how GenAI can be applied across the end-to-end statistical programming value chain. Importantly, this categorisation is intentionally broader than a focus on 'code generation' alone. Statistical programming organisations play a strategic role in clinical development, contributing not only to code production but also to study design interpretation, standards implementation, data quality, analytical rigour and regulatory readiness. Within this context, GenAI is positioned as an enabler of end-to-end value creation across the statistical programming lifecycle, rather than as a tool limited to automating code production.

Appendix B3. Planning and Prioritisation Lenses

To help organisations plan, sequence and scale the adoption of GenAI in statistical programming, two complementary lenses across all use cases can be applied: Value and Architecture Complexity. Together, these lenses support informed decision-making by balancing expected benefit against implementation effort and risk.

Value

The Value lens describes the degree to which a GenAI use case is expected to improve efficiency, quality or delivery timelines within statistical programming workflows.

- Low Value
  Represents marginal or situational benefits. These use cases typically provide incremental assistance for narrow tasks and do not materially change overall productivity, quality or timelines.
- Medium Value
  Delivers noticeable, context-dependent improvements in efficiency or quality. These use cases can meaningfully reduce effort for specific activities or study phases, but still require substantial human involvement and do not fundamentally alter the end-to-end workflow.
- High Value
  Provides broad, repeatable benefits across studies and programmes. These use cases materially reduce delivery timelines, scale effectively across teams, or significantly improve consistency and quality, resulting in a measurable shift in how statistical programming work is performed.

Architecture Complexity

The Architecture Complexity lens reflects the level of technical investment, integration and governance required to safely deploy a given GenAI use case in a regulated environment.

- Low Complexity
  Requires minimal new infrastructure and can be implemented within existing tools and environments (e.g. IDE-embedded assistance). Governance and security requirements largely align with current workflows.
- Medium Complexity
  Requires additional integrations, such as centralised logging, monitoring, controlled model access, or basic orchestration. These use cases introduce new governance considerations but remain manageable within standard enterprise platforms.
- High Complexity
  Requires enterprise-grade architecture, including orchestration of multiple services or agents, strong security and access controls, comprehensive auditability, and alignment with regulatory and compliance frameworks. These use cases are typically associated with large-scale or higher-impact workflows.

How These Lenses Are Used

These two lenses are applied consistently throughout the use case section to help teams:

- Identify early, low-risk wins that deliver immediate value
- Distinguish incremental improvements from transformational opportunities
- Plan a phased adoption roadmap, progressing from simpler use cases towards more advanced, higher-impact implementations as technical and organisational maturity increase.

By explicitly considering both value and complexity, organisations can adopt GenAI strategically, at scale, and in alignment with regulatory expectations, rather than opportunistically or tool-driven.

Appendix B4. Detailed Use Case Catalogue

Code Generation

Intent

Support the creation of static programming artifacts for SDTM, ADaM, and reporting workflows, reducing authoring time while enforcing standards and improving code readability and maintainability.

Key Specific Use Cases

- LLM-First Code Generation for SDTM/ADaM/Reporting
  The LLM generates a complete first draft of production code based on approved specifications and standards. Human programmers review, test, modify and approve the code through mandatory human-in-the-loop (HITL) processes. Provenance of inputs, prompts, generated outputs and subsequent human changes is retained.
- Human-First Programming Supported by GenAI
  The programmer drives the design and core logic, while GenAI assists with boilerplate generation, refactoring, code completion, translation and documentation. Human authorship, review, testing and signoff remain unchanged.
- Language Translation (e.g. SAS → R/Python)
  Legacy code is translated into modern languages with functional equivalence verified through unit and integration tests.
- Synthetic Data Generation
  GenAI creates realistic synthetic datasets based on specifications to enable early programming and detection of edge cases before live data is available.

Value/Architecture Complexity

- Value: medium to high
- Architecture Complexity: low to medium (high for fully LLM-first or agentic implementations)

Programming Specification and Document Generation

Intent

Reduce the manual effort required to author and maintain programming specifications and supporting documents, which are often on the critical path to study start and submission.

Representative Use Cases

- Generation of Programming Specifications
  Drafting standardised analysis plans, programming specifications, and table shells using natural language prompts combined with study metadata and standard templates.
- Drafting of Study and Programming Documents
  First draft creation of documents such as:
  - Programming plans
  - Validation summaries
  - Change logs
  - SDRG/ADRG narrative sections.
- Template-Driven Standardisation
  Enforcing consistency across studies by generating documents from approved templates and controlled vocabularies.

Value/Architecture Complexity
- Value: medium to high (reduces rework and delays upstream of coding)
- Architecture Complexity: low to medium

Key Controls
- Human review and approval
- Versioning and traceability to standards and metadata

Document Understanding

Intent
Enable automation without requiring full standardisation of source documents by extracting structured information from unstructured text.

Representative Use Cases
- SAP to Table Shells
  Extracting populations, endpoints and analysis descriptions from SAPs to generate initial table shells.
- Protocol to Trial Design Domains
  Interpreting protocol text to derive trial design structures (arms, visits, epochs).
- Natural Language to Transformation or Analysis Logic
  Translating narrative descriptions in SAPs or Define-XML documentation into executable transformation or analysis code for human review.

Value/Architecture Complexity
- Value: high
- Architecture Complexity: medium

CDISC Data Generation and Validation

Intent
Assist statistical programmers in the mapping, conversion and validation of CDISC-compliant datasets, improving efficiency and consistency while maintaining regulatory rigour.

Representative Use Cases
- SDTM and ADaM Code Generation
  Generating initial SDTM and ADaM programs based on mapping specifications, metadata, and standards libraries.
- Discrepancy Detection and Suggestion
  Identifying mismatches between specifications, metadata and datasets (e.g. variable derivations, controlled terminology).
- Validation Support
  Assisting with:
  - Identifying potential compliance issues
  - Suggestions for corrections or clarifications
  - Drafting explanations for reviewer guides.

Value/Architecture Complexity
- Value: high (core to submission readiness)
- Architecture Complexity: medium to high

Key Controls
- HITL review of all mappings and derivations
- Alignment with CDISC standards and internal validation SOPs

Output (TFL) Generation and Validation

Intent
Automate and scale the creation and review of tables, figures and listings (TFLs) while ensuring consistency, correctness and compliance.

Representative Use Cases
- Automated TFL Code Generation
  Generating reporting code directly from table shells and specifications.
- Cross-TFL Consistency Checks
  Validating that repeated statistics (e.g. Ns, counts, percentages) are consistent across outputs.
- Specification-to-Output Alignment
  Reviewing TFLs against specifications to detect ambiguities, contradictions or deviations.
- Formatting and Presentation Validation
  Checking adherence to layout, formatting, spelling and style standards for submissions.

Value/Architecture Complexity
- Value: medium to high (reduces late-stage rework)
- Architecture Complexity: medium

Quality Assurance

Intent
Apply GenAI as an additional QA layer after deliverables are created, improving quality and reducing manual, non-analysis effort.

Representative QA Use Cases
- Log & code review
- Error correction and refactoring with documented diffs
- Code explanation and logic clarification
- Automated unit test generation
- Robustness checks for missing data and outliers
- Program header generation and validation

Value/Architecture Complexity
- Value: low to medium individually; high when scaled across studies
- Architecture Complexity: low to medium

Appendix B5. Positioning and Adoption Guidance

Across all categories, consider:
- Starting with low to medium complexity QA and documentation use cases to build confidence
- Expanding into data generation, validation and output automation as governance and skills mature
- Treating GenAI as a strategic accelerator across the lifecycle, not a point solution for coding.

SMEs: Daniel Boisvert, Matt Smith, Jeffrey Meyers, Yang Liu, Ramesh Ayyappath, Tadeusz Lewandowski, Liming Jin, Sam Parmar, Clark Peirce

Appendix C – Technical Enablement and Architecture

Appendix C1. AI Enablement and Implementation Considerations

For statistical programming organisations, successful AI implementation requires a thorough review of the current end-to-end technology stack and reporting capabilities. Teams should assess how their existing infrastructure, including data storage, processing platforms, workflow automation tools, and reporting systems, can support and integrate with AI solutions. Evaluating compatibility, interoperability and readiness for AI-driven processes helps organisations identify gaps and opportunities for enhancement.

It is important to understand any AI-related implementation plans or roadmaps from third-party vendors whose software is licensed by the company, as these may impact on future integration strategies and overall technology alignment.

Additionally, it is critical to understand enterprise-wide AI capabilities. Organisations should catalogue available resources such as internal AI platforms, governance frameworks and support teams to ensure new AI initiatives align with broader corporate strategies and standards. This approach allows statistical programming teams to leverage existing investments, avoid redundant efforts, and accelerate adoption. Establishing clear governance, roles and responsibilities for AI enablement further ensures an efficient and compliant implementation that maximises value across the ecosystem.

Appendix C2. Technical Platforms, Reference Architectures and Key Technology Considerations

This section describes the technical foundations required to safely and compliantly enable Generative AI (GenAI) across the statistical programming lifecycle safely, at scale. The focus is intentionally on reference architectures and enabling capabilities, rather than on specific products or vendors.

The goal is to support:

- Human-in-the-loop (HITL) workflows that are suitable for regulated environments
- Advanced agentic workflows that automate multi step tasks while preserving auditability, traceability and accountability.

Reference Architecture Patterns

GenAI adoption represents:

- *"Doing what we do today, but faster"*
- *A new way of working in GenAI-native environments.*

Accordingly, two complementary reference architectures are defined.

HITL-Centric Reference Architecture (Baseline)

This architecture is suitable for most current regulatory contexts.
Key characteristics:

- GenAI assists with authoring, review and QA.
- All outputs are static artifacts (code, datasets, documents).
- Human review and approval are mandatory.
- GenAI does not execute production code autonomously.

Logical flow
User/IDE ➔ Prompt & Context Layer ➔ LLM ➔ Tooling (tests, linting) ➔ SDLC systems ➔ Audit & Governance

This pattern aligns well with existing GxP, Part 11 and validation frameworks.

Agentic Reference Architecture (Advanced)

This architecture supports multistep automation (e.g. SDTM ➔ ADaM ➔ TFL traceability, large-scale QA) while maintaining control. The project-level configuration file ensures consistency and standardisation across multiple artifacts.
Key characteristics:

- GenAI agents perform bounded tasks in sandboxed environments.
- Agents operate with least privilege permissions.
- Outputs are returned as reports, or review packages.
- No direct writes to validated production systems.

Logical flow
User/Orchestrator ➔ Agent Runtime ➔ Context & Tool Servers ➔ LLM(s) ➔ Results ➔ Human Review
This pattern reflects a future-ready SCE but requires stronger governance and maturity.

Key Technology Considerations

AI Model Selection

GenAI model selection must consider:

- Suitability for pharmaceutical terminology and statistical reasoning
- Support for enterprise controls (logging, isolation, versioning)
- Stability and reproducibility (model version pinning).

Importantly:

- The LLM itself is not validated as a GxP system
- The process and system using the LLM (review, testing, approvals, traceability) must be validated per company policy.

This distinction was explicitly raised in the Working Group comments and is foundational to scalable adoption.

Integration with Existing Statistical Computing Environments (SCEs)

GenAI must integrate seamlessly into established environments, including:

- SAS, R, Python
- IDEs such as VS Code, SAS EG/Viya, RStudio, Positron.

Integration patterns include:

- Embedded assistants (interactive prompting)
- CLI-based tools for advanced workflows
- APIs interfacing programming languages (e.g. R, Python) with LLM services
- SDLC integration via version control and CI/CD
- Context-aware assistants that leverage approved coding standards, repository-level instructions (e.g. custom IDE or repository guidance) and standard code libraries (such as validated SAS macros or reusable functions).

A key design principle is that GenAI outputs flow through the same SDLC and validation processes as human written artifacts.

Data Security and Secure Inference

Security considerations include:

- Use of enterprise, allow listed GenAI interfaces only
- No PHI/PII exposure to public models
- Encrypted inference and storage
- Prompt/output logging aligned with DLP policies
- Network isolation and sandboxing for agents.

For agentic workflows, an explicit design choice must be made between:

- Execution based on user credentials, and
- Non-human identities with gated exits (preferred for higher-risk use cases).

Scalability and Performance

As GenAI adoption scales, platforms must support:

- Multiple concurrent users
- Large code bases and datasets
- Batch QA and validation tasks.

Key considerations:

- Latency and throughput
- Cost and quota management
- Monitoring and operational alerting
- Elastic compute scaling

Scalability becomes critical when moving from pilot use cases to portfolio-level adoption.

Appendix C3. Context Management as a Critical Enabler

Effective context management is one of the most important, and often underestimated, technical requirements for GenAI in statistical programming.

Context includes:

- Study metadata
- SAPs, protocols, table shells
- Programming standards and templates
- Approved macros and libraries
- Prior outputs and lineage information.

Without disciplined context management, GenAI outputs become inconsistent, hard to reproduce and difficult to audit.

To address these limitations in practice, organisations typically adopt approaches from manual prompt stuffing to retrieval augmented generation (RAG) over governed knowledge sources.

Context Management Approaches

Common approaches include:

- Prompt stuffing (small-scale, manual)
- RAG using curated knowledge sources
- Structured context services for larger-scale and agentic workflows.

This is where MCP servers become relevant.

MCP Servers and Tooling Ecosystem

What MCP Servers Enable

Model context protocol (MCP) servers provide a standardised interface for GenAI systems to securely access tools, data and context.

In a statistical programming setting, MCP-style servers provide:

- Controlled access to standards, metadata and specifications
- Read-only access to datasets or code repositories
- Execution of approved tools (e.g. linters, validators, consistency checks)
- Structured, auditable context injection into LLM prompts.

Crucially, MCP servers help avoid:

- Ad hoc prompt engineering
- Manual copy/paste of sensitive context
- Undocumented context changes that break reproducibility.

MCP in HITL vs Agentic Architectures

In HITL architectures:

- MCP servers act as context providers
- The human remains the orchestrator
- Context is transparent and reviewable.

In agentic architectures:

- MCP servers define what an agent can see and do
- Tool access is explicitly scoped
- Outputs are returned for human approval.

AI-Ready Context and Governance[14]

Much enterprise data and context, broadly defined to include datasets, metadata, documents and supporting artifacts, exists today in formats such as:

- Word documents
- Excel spreadsheets
- Free-text formats.

To be safely usable by GenAI, this context must become AI-ready, meaning:

- Structured or markdown-compliant
- Versioned and governed
- Clearly scoped for access.

MCP servers (or equivalent context services) provide a natural place to enforce:

- Governance rules
- Access controls
- Audit logging
- Reproducibility guarantees

A practical next step for organisations is to define an 'AI-ready context checklist' (formats, ownership, approval, retention).

Platform Options and Deployment Models

Organisations typically adopt:

- Managed AI platforms (fastest time to value)
- Enterprise self-managed platforms (maximum control)
- Hybrid models (most common).

Decision criteria include:

- Data sensitivity and regulatory impact
- SCE integration complexity
- Audit and traceability requirements
- Readiness for agentic workflows.

Agentic capabilities, often enabled through MCP-like architectures, are increasingly viewed as core future features of statistical computing environments.

Required Capabilities vs Opinionated Best Practices

It is useful to distinguish:

Required Best Practices (regulatory/quality-driven)

- Human-in-the-loop review
- Static outputs
- Traceability and audit logging
- Secure inference and access control

Opinionated Best Practices (maturity-driven)

- Agent sandboxing
- MCP-based context services
- Prompt and context versioning
- RAG over governed knowledge sources.

Together, these capabilities ensure GenAI adoption in statistical programming is safe by design, scalable, and futureproof, while remaining aligned with regulatory expectations and the strategic evolution of the function.

SMEs: Sharad Chhetri, Jiayi Qin, Thiago Figueiredo, Matt Smith, Yang Liu, Tadeusz Lewandowski, Sam Parmar

Appendix D – Human, Organisational and Operating Model

Appendix D1. Human-in-the-Loop (HITL) Framework

The strategic use of Generative AI (GenAI) in statistical programming requires a robust human-in-the-loop (HITL) approach to ensure scientific integrity, regulatory compliance, and accountability. HITL does not replace established authoring, independent review, or validation processes; rather, it augments existing practices to reflect the realities of GenAI-assisted development. GenAI workflow should be designed to cater to HITL, meaning it should expose capabilities and limitations and provide tools for human overseers. The aim is for the human to see into the black box of the LLM.

At its core, HITL ensures qualified humans remain accountable for all analytical outputs, regardless of the degree of GenAI assistance. All GenAI-generated code, analyses and results must be reviewed, approved and signed off by appropriately trained individuals in accordance with applicable organisational SOPs.

Principles of HITL in Statistical Programming

HITL for GenAI-assisted statistical programming is guided by the following principles:

- Human accountability: A named, qualified individual retains responsibility for correctness, interpretability and fitness for purpose of all deliverables.
- Risk-based oversight: The intensity of human review scales with the intended use and risk of the output (e.g. exploratory analyses vs. regulatory submissions).
- Transparency and traceability: The use of GenAI to generate GxP-compliant outputs should be transparently documented in proportion to its scale, impact and risk.
- Independence of judgement: Reviewers must critically assess outputs and not rely on AI-generated rationale or 'chain of thought'.

Review and Quality Control

Rather than prescribing a fixed 'dual review' model, HITL aligns with existing QC and independent review SOPs, adapted for AI-assisted workflows. This represents a 'new normal' for QC, where GenAI is treated as a powerful productivity tool but not an autonomous contributor.

Roles and Responsibilities

Clear role definition is essential for effective HITL. Typical roles may include:

- Prompter/Developer: Uses GenAI to draft or modify code.
- Reviewer: Independently evaluates AI-assisted outputs against specifications, standards and expectations.
- Statistical or Domain Expert: Interprets outputs and confirms scientific validity.
- Approver: Provides final sign off and assumes accountability.

These roles may be combined or separated depending on risk, but independence of review and accountability must be preserved.

Documentation, Audit Trail and Traceability

To support transparency, HITL requires:

- Documentation of:
  - GenAI model/provider and version (where applicable)
  - Prompts or prompt templates used
  - Generated outputs and subsequent human modifications
  - Tests performed and review decisions
- Maintenance of audit trails capturing human interventions and rationale for acceptance, rejection or override
- Optional tagging or annotation of AI-generated code in repositories to support traceability and explainability.

Validation, Monitoring, and Feedback Loops

HITL extends beyond initial code generation:

- Validation and verification activities remain mandatory and are performed by humans.
- Continuous monitoring should be applied when AI tools are used repeatedly, to detect performance degradation, unexpected behaviour, or changes in output quality.
- User feedback mechanisms should enable programmers and statisticians to flag errors, ambiguities or systematic issues in AI outputs.
- Where models are updated or retrained, traceability must be preserved.

Transparency, Explainability and Bias Awareness

Humans must be able to understand and justify AI-assisted outputs:

- Outputs should be interpretable and explainable to the extent required for the use case.
- Reviews should explicitly consider potential bias, especially for analyses affecting patient subpopulations.
- AI generated rationale should not be accepted uncritically; evidence and statistical reasoning must stand on their own.

Training and Awareness

Effective HITL depends on informed users. Training should cover:

- Capabilities and limitations of GenAI tools
- Prompting best practices
- Data privacy and security requirements
- Documentation and inspection readiness expectations
- Common failure modes and cognitive biases associated with AI use.

Acceptable HITL Workflows

Multiple GenAI-assisted workflows may be acceptable when governed by risk-appropriate HITL controls, for example:

- From scratch generation: AI generates code from a specification; human reviews code; statistician reviews outputs.
- Reuse focused workflows: AI assembles existing validated libraries or macros; human reviews parameterisation and results.
- AI-assisted validation: AI supports consistency checks between specifications, code and outputs, with humans retaining final judgement.

The appropriateness of conservative versus more liberal use of GenAI should be determined through organisational consensus and risk assessment, not tool capability alone.

Appendix D2. Organisational Considerations

Successful adoption of Generative AI (GenAI) in statistical programming is not solely a technology challenge; it is fundamentally an organisational and cultural transformation. Even when technical platforms and governance frameworks are well designed, GenAI will only deliver sustained value if teams understand how it fits into their day-to-day work and trust it as a disciplined, well-controlled capability. As such, organisational readiness must be addressed explicitly alongside technical and regulatory considerations.

Change management is a critical first component. GenAI introduces new ways of working, including AI-assisted drafting, more iterative review cycles, and increased emphasis on upfront specification quality. Organisations should proactively prepare teams for these changes by clearly communicating intent, scope and boundaries. This includes reinforcing that GenAI is an augmentation tool, not a replacement for expertise, and that accountability remains unchanged. Transparent messaging, visible leadership support, and early involvement of end users help reduce uncertainty and resistance, enabling smoother adoption.

Equally important is the development of new and evolving skill sets within statistical programming organisations. While deep AI expertise is not required for most users, teams must build practical proficiency in areas such as effective prompting, interpreting GenAI outputs, recognising common failure modes, and responsibly handling data and context. In parallel, selected roles may require deeper expertise in AI enablement, such as prompt pattern development, context management, workflow orchestration and data governance. Training programmes should be role-based, pragmatic and continuously updated as tools and use cases mature.

As GenAI adoption scales, organisations must also invest in AI literacy and awareness across the broader function. This includes educating teams on limitations such as hallucination, bias and nondeterminism, as well as reinforcing expectations for documentation, review and inspection readiness. Building this shared understanding helps prevent overreliance on AI outputs and supports consistent, high-quality human-in-the-loop review practices.

Cross-functional collaboration is another key enabler of value realisation. GenAI use in statistical programming sits at the intersection of statistics, programming, data science, IT, quality, privacy and compliance. Close collaboration between these groups is essential to align on acceptable use cases, risk tiers, validation approaches and operational guardrails. Early and ongoing involvement of QA, regulatory and compliance partners ensures GenAI-assisted workflows are inspection ready by design, rather than retrofitted after the fact.

Finally, organisations should view GenAI adoption as an evolving capability, not a one-time rollout. Operating models, roles and governance structures may need to adapt as GenAI use expands from low-risk efficiency gains into more central programming activities. Establishing feedback loops, communities of practice and shared learning mechanisms allows organisations to capture lessons learned, refine standards and continuously improve how GenAI is used. When supported by strong change management, targeted skill development and cross-functional collaboration, GenAI becomes a sustainable organisational capability that enhances statistical programming performance while preserving quality, trust and compliance.

Appendix D3. Roles and Responsibilities

Maintain existing accountability structures, with targeted additions for GenAI enablement and governance. Some organisations may designate roles like AI subsystem owner, tailored to your scale and risk profile.

Reviewer (independent): performs spec-to-code-to-output verification; documents rationale for acceptance/rejection; avoids reliance on AI 'chain of thought'.

Enablement Roles (as applicable): prompt engineer/AI workflow designer (maintains prompt patterns and evaluation harnesses); knowledge/RAG engineer (connects models to standards); AI platform engineer (performance, cost, integration).

Operations & Change
Training lead (curriculum on safe use, prompting, review standards), DevOps/MLOps (CI/CD, environments, artifact management), release/config manager (traceable releases), and support (access, incident triage).

Appendix D4. Operating Procedures (SOPLite)

A concise, implementable SOPLite is defined to complement existing SOPs by clearly specifying who performs which activities when, and what evidence must be retained, while deliberately avoiding overprescription of QC mechanics addressed in detailed SOPs and work instructions.

Eligibility & Setup
Permitted Use Cases: Define approved use cases and risk tiers (exploratory → submission).
Approved Tools: Publish an allowlist of GenAI interfaces/models and versions; document constraints (no PHI/PII; no confidential data in public tools).
Access & Controls: Enforce SSO, least privilege, logging, and DLP; keep endpoints behind firewall where required.

Prompting & Logging
Use approved prompt templates or patterns; record model/provider/version, prompt(s), outputs and key human decisions (store in the project repo or designated registry). For higher-risk tiers, attach acceptance criteria and link to spec/test artifacts (e.g. unit tests).

Development & Review
Decompose tasks into small, reviewable diffs; enforce independent human review before merge (no auto merge from GenAI branches).
Validate code as per existing SOPs: unit/integration tests, spec-

to-output checks, reproducibility, and change control.

Release & Provenance
Tag and archive releases with provenance (specs, prompts, outputs, tests, approvals), ensuring auditability in the Part 11 scoped systems where execution/records reside.

Monitoring & Exceptions
Perform periodic checks for drift/hallucinations, or for systematic issues; review incident trends and open CAPAs where needed.
Exception Handling: If anomalies are detected (e.g. unexpected methods, insecure code), pause use, escalate to QA/InfoSec as appropriate, and document decisions and remediation.

SMEs: Sharad Chhetri, Jiayi Qin, Mark Reynolds, Jacob Kaiser, Shereen Khwajazada

Appendix E – Risk, Validation, Privacy and Ethics

Appendix E1. Risk-Based Controls and Risk Management

The implementation of Generative AI (GenAI) in statistical programming introduces new risks that must be systematically identified, assessed and managed. Rather than prescribing a single set of controls, organisations should establish a GenAI risk assessment framework that evaluates each use case based on its context of use, data sensitivity, and potential impact on regulated outputs.

Risk Assessment Framework
Organisations should assess GenAI use cases along the following core dimensions:

- Context of use
  Process that the GenAI system will support, replace or augment (e.g. drafting code vs supporting regulatory decision-making)
- Model characteristics
  Model the type, training and tuning data used, the version stability, and transparency
- Users and consumers of outputs
  Internal use only vs. broader organisational or external consumption
- Potential impact
  Impact on confidentiality, integrity and availability (CIA) of data and results

These factors together determine the risk tier of a given GenAI use case and the level of governance required.

Risk Categorisation
Based on the assessment dimensions above, GenAI use cases can be categorised into three broad risk tiers:

Low Risk – Operational Efficiency Use Cases
Low-risk use cases are limited to internal productivity and process optimisation and do not directly impact on regulated decision-making.

Characteristics
- Used within a single department or team and not shared or applied externally
- Not for any business-critical processes
- Outputs are reviewed and validated using existing SOPs
- No direct regulatory submission dependency
- No personal or confidential data exposed to unapproved systems

Examples
- Code explanation and refactoring
- Drafting comments or documentation
- Generating unit tests
- Creating synthetic data for development or testing

Controls
- Internal organisational guidelines
- Human-in-the-loop review
- Standard version control and QC practices

Moderate Risk – Business-Critical but Nonregulatory Use Cases
Moderate risk use cases support business-critical workflows or are shared more broadly across teams, but do not independently determine regulatory outcomes.

Characteristics
- Outputs inform important internal decisions
- May be shared across departments or functions
- Still subject to human review before use

Examples
- Cross-TFL consistency checks
- Programming specification generation
- SDTM/ADaM mapping assistance

Controls
- Robust internal organisational GenAI standards
- Centralised logging and traceability
- Alignment with industry best practices and internal governance frameworks

High Risk – Regulatory or High-Impact Use Cases
High-risk use cases are those that support regulatory submissions, influence patient safety or materially affect the reliability of clinical trial results.

Characteristics
- Outputs contribute directly to submission artifacts
- Potential regulatory or patient impact if incorrect
- May involve sensitive or confidential data

Examples
- GenAI-assisted analysis programs used for submissions
- Automated generation of submission documentation without adequate review

Controls
- Alignment with applicable regulatory frameworks (e.g. FDA Credibility Assessment Framework, EMA Reflection Paper on AI, EU AI Act)
- Explicit definition of context of use (COU), model risks, etc.
- Enhanced validation, documentation, and audit readiness
- Clear accountability and signoff by qualified individuals

Operational Efficiency vs Reliability of Results
In many statistical programming scenarios, such as generating static code files for subsequent review and QC, GenAI functions as a productivity accelerator, not an autonomous analytical system. In these cases, risk is primarily mitigated through existing review, testing and validation practices, even if the initial draft is produced by AI.

Key Risk Areas and Mitigation Considerations

The following risk areas should be considered across all GenAI use cases, with mitigation strategies tailored to the assessed risk level.

Bias and Hallucinations

GenAI models may produce incorrect or fabricated outputs due to limitations or biases in training data.

Mitigations

- Mandatory human review
- Periodic unit and integration testing
- Periodic audits of outputs
- User training to recognise AI-generated errors

Legal and Intellectual Property Risk

GenAI systems may inadvertently reproduce copyrighted or proprietary material breaching compliance with laws and regulations.

Mitigations

- Use of enterprise-approved models
- Understanding provider training data disclosures
- Clear policies on reuse and attribution

Reproducibility and Consistency

Non deterministic models and evolving context can lead to inconsistent outputs across runs.

Mitigations

- Static outputs
- Prompt, context, and model versioning
- Controlled context injection
- Avoidance of 'learning-in-use' in regulated workflows

Data Privacy and Security

There is a risk of unintentionally exposing sensitive or confidential data.

Mitigations

- No PHI/PII in public models
- Secure, allowlisted inference endpoints
- Data minimisation and access controls
- Network isolation and sandboxing for agents

Transparency and Accountability

Overreliance on GenAI without understanding its underlying logic or limitations can lead to undetected errors.

Mitigations

- Clear documentation of intended use
- Defined roles and responsibilities
- Explicit human signoff and accountability
- Avoidance of reliance on AI 'chain of thought' as justification

Risk Management Lifecycle (NIST AI RMF)

Risk management should be treated as a continuous lifecycle, rather than a one-time assessment. Organisations are encouraged to align with the NIST AI Risk Management Framework (AI RMF) and GenAI Profile:

- Govern – Establish oversight, policies and accountability
- Map – Identify GenAI use cases, risks and contexts
- Measure – Assess the likelihood and impact of identified risks
- Manage – Mitigate, monitor and respond to risks over time

A dedicated governance body should periodically review, update and audit the risk framework as GenAI capabilities, regulations and use cases evolve.

Summary

A risk-based approach enables organisations to safely scale GenAI adoption in statistical programming without unnecessarily constraining low-risk productivity gains or undergoverning high-impact use cases. By aligning risk controls with the context of use, regulatory impact and existing quality practices, GenAI can be integrated as a trusted, auditable and value-adding component of the statistical programming function.

Appendix E2. Data Privacy, Security & Compliance

The following considerations are to be implemented in line with each organisation's policies, procedures and legal guidance (do not supersede local SOPs). If no AI policy exists, consult Legal/Compliance, Privacy, InfoSec and QA to establish interim guardrails.

Use of Public vs. Enterprise GenAI

Prefer enterprise-secured GenAI (controls for data residency, access, logging, retention). Public LLMs may be acceptable for low-risk, nonconfidential lookups (akin to googling or checking community forums), provided no PHI/PII or confidential data is shared.

Understand gradations of security:

- Least secure: direct use of public tools/APIs
- More secure: enterprise cloud with dedicated infrastructure and safeguarded connectivity
- Potentially secure: requires strong governance, local LLMs with audited OSS stacks and sandboxing

Handling of Sensitive Data

Never enter PHI/PII or company confidential datasets into public LLMs. Apply DLP and access controls in enterprise channels, and follow approved retention and logging practices.

Regulatory & Policy Alignment

For regulated work, segregate GxP vs. non-GxP environments and apply risk-based validation and audit trails consistent with 21 CFR Part 11 expectations in the systems of record where code is executed and controlled. (Discussion point recognised: code generation may sit outside Part 11; the execution and record-keeping environment is typically the scoped Part 11 system.)

Maintain logs of model/provider/version, prompts/outputs, human decisions, and retention aligned with corporate policy and evolving obligations (e.g. transparency for general-purpose AI).

Where internet access is restricted, keep AI endpoints behind firewalls and prevent unsolicited uploads/downloads.

Third-Party and IP Considerations

Confirm licence terms for generated code and third-party packages; ensure usage and redistribution conform to corporate standards (coordinate with Legal/Compliance).

Appendix E3. Validation and Quality Management (GxP Focus)

Scope. Emphasise what is required beyond typical controls when GenAI assists code creation. Align with existing validation SOPs; avoid treating the LLM like a classical CSV component unless it is truly part of the validated system boundary.

System vs. Artifact View

- System view: If the LLM or its orchestration becomes part of a regulated 'system', follow your software validation procedures (e.g. GAMP-aligned) with documented roles, change control, and evidence. Acknowledge that classically validating a general LLM is challenging; focus validation on the workflow and controls around it.
- Artifact view: If the LLM creates static code artifacts (e.g. SAS/R/Python files) that are then run within the validated environment, treat those artifacts like human written code under existing risk-based validation (unit/integration tests, independent review, reproducibility, and traceable approvals).

Risk-Based Program Validation

Apply risk tiering by intended use (exploratory/internal vs. decision informing vs. regulatory submission). Use test-driven development where feasible; require independent code review and spec-to-output verification; manage changes via version control and change records.

Tool Qualification & Evidence

Qualify approved GenAI interfaces and constraints (who can use, for what tasks, which models, with what logging). Capture IQ/OQ/PQ appropriate evidence for the workflow (not the general model internals) and demonstrate reproducibility and traceability of analysis outcomes.

Monitoring, Audits and Incidents

Establish periodic audits, monitoring of output quality (e.g. spot checks, failure trend analysis) and an incident/CAPA pathway for AI-related deviations.

Specifications & Context Management

Ensure clear analysis specifications and stable context management (standards, templates, libraries) to reduce variability in LLM outputs and improve predictability/ reproducibility.

Appendix E4. Ethics and Responsible AI Considerations

Scope/Proportionality. Ethical risks are lower for code generation than for AI-analysing patient data; nonetheless, responsible AI practices still apply, particularly for transparency, accountability and misuse prevention.

Values, Transparency, Accountability

- Use a lightweight process to elicit stakeholder values (fairness, transparency, accountability) and translate them into requirements for how GenAI is used in programming (logging, explainability of changes, reviewer accountability).
- Prefer interpretable approaches for high-impact outputs; if backbox assistance is used, ensure compensating controls (e.g. independent human verification, test coverage and traceable signoff).

Bias & Safety Context

While the model is not processing PHI/PII during code generation, bias can still enter via libraries, defaults or examples. Encourage reviewers to scrutinise default settings (e.g. imputation methods, outlier rules) and ensure methods remain scientifically justified.

Responsible Use Practices

Align team practices with responsible AI norms: clear intended use, human oversight, meaningful review, secure handling of prompts/outputs, and readiness for inspection (documentation, traceability).

SMEs: Harpreet Pruthi, Chadha Taranbirsingh, Daniel Boisvert, Matt Smith, Daniel Schramek, Jeffrey Meyers

Appendix F – Regulatory Alignment & External Frameworks

Appendix F1. Regulatory Alignment Overview

The regulatory guidelines and standards referenced in this section establish the external regulatory and governance context within which Generative AI (GenAI) can be responsibly applied to statistical programming in clinical trials. Collectively, these documents define the regulatory perimeter for AI use in drug development and clarify regulator expectations around safety, reliability, transparency and accountability.

These frameworks directly inform and justify the core design principles, including:

- Human-in-the-loop (HITL) workflows, ensuring clear accountability and expert oversight
- Static, reproducible outputs, aligned with inspection, audit and submission requirements
- Risk-based controls, proportionate to the context of use and potential regulatory impact
- Strong governance and traceability, supporting data integrity, validation and audit readiness.

Importantly, these regulatory readings also help distinguish acceptable uses of GenAI for operational acceleration (such as drafting code or documentation under standard QC processes) from higher-risk analytical uses that may influence the reliability of clinical trial results and therefore require additional controls, validation and regulatory scrutiny. By grounding this framework in established regulatory guidance, the paper provides statistical programming teams with a **defensible, inspection-ready approach** to leveraging GenAI – one that enables innovation and efficiency while remaining fully aligned with global regulatory expectations.

Appendix F2. Key Regulatory and Standards References

EMA/FDA Guiding Principles of Good AI Practice (Jan 2026)

Why It Matters

This joint EMA/FDA document provides a regulator-endorsed philosophy for AI use in drug development. Rather than prescribing technology, it sets expectations for how AI should be designed, deployed and overseen.

Relevance to statistical programming
- Reinforces human-centric AI, directly supporting the paper's insistence on human accountability for AI-generated code and outputs.
- Endorses a risk-based approach, allowing differentiation between:
  - Low-risk operational uses (e.g. drafting code, documentation) and
  - Higher-risk uses that could affect the reliability of trial results.
- Validates positioning GenAI as an augmentation tool, not an autonomous decision maker.

Why it belongs in the paper
It establishes that the paper's framework aligns with global regulatory intent, not just internal best practices.

EU AI Act (Including GPAI Obligations)

Why it matters
The EU AI Act is the first binding legal framework governing AI systems and directly affects the LLMs statistical programmers rely on, even if those tools are not built in house.

Relevance to statistical programming

- Introduces GPAI provider obligations, shifting part of compliance to tool vendors:
  - Technical documentation
  - Copyright-compliant policies
  - Training data transparency
- Uses a tiered risk model, reinforcing the paper's use case specific governance rather than blanket restrictions.
- Strengthens expectations around traceability, security and incident handling, which are essential for audit-ready programming workflows.

Why it belongs in the paper
It explains why tool selection matters and why not all GenAI tools are appropriate for regulated statistical programming.

FDA Draft Credibility Framework (Jan 2025)

Why it matters
This framework defines when AI outputs must be shown to be credible for regulatory decision-making and, when AI outputs do not.

Relevance to statistical programming

- Clarifies that operational drafting tasks (e.g. code generation later QC'd) are excluded, preventing unnecessary overvalidation.
- Provides a structured, defensible pathway for higher-impact use cases where AI output may influence regulatory conclusions.
- Introduces context of use (COU) as a central concept, which aligns with the paper's use case-driven structure.

Why it belongs in the paper
It supports a balanced approach: rigorous where needed, pragmatic where appropriate.

EMA Reflection Paper on AI (Sep 2024)

Why it matters
This is the most directly applicable regulatory guidance for AI use in statistical programming.

Relevance to statistical programming

- Explicitly states that AI used for data transformation, analysis or inference becomes part of statistical analysis.
- Requires such use to be prespecified, frozen and traceable before unblinding.
- Rejects learning-in-use models in confirmatory settings.
- Emphasises strong governance, validation, data integrity and human oversight.

Why it belongs in the paper
It directly justifies:
- Static outputs over dynamic AI execution
- Strict traceability and reproducibility
- Clear separation between exploratory efficiency and confirmatory evidence.

NIST AI Risk Management Framework (AI RMF) + GenAI Profile

Why it matters
NIST AI RMF provides a practical, operational risk framework that organisations can implement today.

Relevance to statistical programming
- Maps directly to GenAI risks encountered in programming:
  - Hallucinated logic
  - Data leakage
  - IP and licensing issues
  - Bias in interpretation
- Offers a unified way to manage risks across HITL, privacy, validation and governance using Govern–Map–Measure–Manage.
- The GenAI Profile explicitly addresses non determinism, a key concern for reproducibility.

Why it belongs in the paper
It serves as the structural backbone for the paper's risk considerations section.

ISO/IEC 42001 – AI Management System (AIMS)

Why it matters
ISO/IEC 42001 defines how organisations should systematically manage AI, much as ISO 9001 and ISO 27001 manage quality and security.

Relevance to statistical programming

- Reinforces that GenAI must operate within a managed system, not ad hoc experimentation.
- Supports:
  - Defined roles and responsibilities
  - Risk and impact assessments
  - Continuous monitoring and improvement.
  - Integrates naturally with existing GxP, QMS and ISMS frameworks.

Why it belongs in the paper
It positions GenAI enablement as a mature organisational capability rather than a one-off technology deployment.

IEEE 7000 – Ethics-by-Design

Why it matters
IEEE 7000 provides a structured method for embedding ethical considerations into system design.

Relevance to statistical programming
- While ethical risk is lower for code generation, it reinforces:
  - Transparency about AI use
  - Accountability for outputs
  - Traceability from requirements to deliverables.
- Helps concretely answer *"What does ethics mean for AI-assisted statistical programming?"*

Why it belongs in the paper
It ensures the framework addresses responsible AI expectations without overstating ethical risk.

FDA 21 CFR Part 11

Why it matters
21 CFR Part 11 defines the baseline requirements for electronic records and signatures in regulated environments.

Relevance to statistical programming
- Confirms that AI-assisted code used in submissions remains subject to:
  - Audit trails
  - Record retention
  - Controlled access
  - Human approval.
- Clarifies that GenAI typically sits outside the validated system, while outputs and execution must comply with Part 11.

Why it belongs in the paper
It anchors GenAI use in well-understood inspection language, reducing uncertainty and audit risk.

How These Regulations Impact on GenAI Tool Selection

Collectively, these regulations impose practical constraints on which GenAI tools are suitable for statistical programming.

Transparency and Traceability

GenAI tools must support:
- Prompt and output capture
- Model/version identification
- Integration with version control and approval workflows.

Tools that produce **ephemeral, untraceable outputs** are unsuitable.

Human Oversight and Controllability

Tools must:

- Enable human review and approval
- Avoid autonomous execution or submission
- Support clear accountability.

This rules out 'black-box' automation tools with limited governance controls.

Data Protection and IP Safety

Preferred tools:
- Are enterprise secured
- Do not reuse prompts or outputs for training without consent
- Provide clarity on training data and copyright handling.

Public or opaque tools pose **compliance and IP risks.**

Reproducibility and Stability

Regulations favour tools that:
- Support static outputs
- Allow context and version pinning
- Avoid hidden model drift in regulated workflows.

Fit Within Existing Validation and Quality Systems
GenAI tools must integrate with:
- GxP environments
- Part 11 compliant systems
- Established SOPs, QC, and audit processes.

These regulatory readings collectively justify the paper's central thesis:

GenAI can be strategically and safely leveraged in statistical programming, but only when embedded in human-led, risk-based, auditable and well-governed workflows.

SMEs: Thiago Figueiredo, Matt Smith, Harpreet Pruthi, Ramesh Ayyappath, Rashad Rasul, Tadeusz Lewandowski, Liming Jin, Shereen Khwajazada

Resources

[1] Gartner. (2026, 22 July). Why the AI Value Gap Is Really an AI Risk Gap. https://www.gartner.com/en/articles/ai-value.

[2] Johnson, P. (2025). Pondering AI. Technicalities, 45(5), 2–3. For more details, visit Pondering AI - Apple Podcasts.

[3] International Society for Pharmaceutical Engineering. (2022, March–April). AI maturity model for GxP application: A foundation for AI validation. Pharmaceutical Engineering.

[4] Willison, S. (2025, 3 February). Simon Willison's weblog. A computer can never be held accountable.

[5] U.S. Food and Drug Administration. (January 2026). *Guiding principles of good AI practice in drug development.* https://www.fda.gov/media/189581/download

[6] National Institute of Standards and Technology. (2023). Artificial Intelligence Risk Management Framework (AI RMF 1.0) (NIST Special Publication No. AI.100-1). U.S. Department of Commerce. https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-1.pdf

[7] llms.txt. (n.d.). https://llmstxt.org/

[8] Agents.md. (n.d.). https://agents.md/

[9] Model Context Protocol. (n.d.). What is the Model Context Protocol (MCP)? https://modelcontextprotocol.io/docs/getting-started/intro

[10] Anthropic. (n.d.). Skills documentation. https://code.claude.com/docs/en/skills

[11] OpenAI. (n.d.). Skills [Computer software]. https://github.com/openai/skills

[12] GitHub. (n.d.). About agent skills. https://docs.github.com/en/copilot/concepts/agents/about-agent-skills

[13] Yin, W., Zhang, S., Wu, Y., & Zhang, Z. (2024). Model context protocol (MCP): A standardized interface for context exchange in LLM agents (arXiv:2602.11988). arXiv.

[14] ai4b. (2025, 4 May). Data readiness assessment: Is your data prepared for AI success? DEV Community.

[15] UK Government Department for Business, Energy & Industrial Strategy. (n.d.). inspect_ai [Computer software]. https://github.com/UKGovernmentBEIS/inspect_ai

[16] European Medicines Agency. (2024, 9 September). *Reflection paper on the use of artificial intelligence (AI) in the medicinal product lifecycle.* https://www.ema.europa.eu/en/documents/scientific-guideline/reflection-paper-use-artificial-intelligence-ai-medicinal-product-lifecycle_en.pdf

[17] Geraci, J., Rao, P., Grandinetti, C., Qorri, B. et al. (2025). Current opportunities for the integration and use of artificial intelligence and machine learning in clinical trials: Good clinical practice perspectives. *Journal of the Society for Clinical Data Management,* 5(2). https://doi.org/10.47912/jscdm.426

[18] Laurent, A. (2026, February 14). A GAMP 5 guide to *AI/ML validation in GxP environments.* IntuitionLabs. https://intuitionlabs.ai/articles/gamp-5-ai-ml-validation-gxp

[19] Lemaire, W., Nadolny, P., Andrus, J., Schaffer, C., & Cameron, S. (2025). SCDM Executive Committee Perspective: The Future of Clinical Trials with AI and Decentralized, Hybrid and Data-driven Approaches. *Journal of the Society for Clinical Data Management*, 5(3). https://doi.org/10.47912/jscdm.448

[20] U.S. Food and Drug Administration. (2025). *Considerations for the use of artificial intelligence to support regulatory decision-making for drug and biological products: Draft guidance for industry and other interested parties.* https://www.fda.gov/regulatory-information/search-fda-guidance-documents/considerations-use-artificial-intelligence-support-regulatory-decision-making-drug-and-biological

[21] U.S. Food and Drug Administration. (2025, 1 December). FDA expands artificial intelligence capabilities with agentic AI deployment [Press release]. https://www.fda.gov/news-events/press-announcements/fda-expands-artificial-intelligence-capabilities-agentic-ai-deployment

[22] Anderson, A. (2023, 6 September). Responsible AI use with Human-in-the-Loop (HITL) [Image]. LinkedIn. https://www.linkedin.com/pulse/responsible-ai-use-human-in-the-loop-hitl-anderson-anthony/

[23] Gartner. (2024, 12 February). *How to calculate business value and cost for generative AI use cases.* https://www.gartner.com/en/documents/5188263

[24] Deloitte. (2026, 19 January). *AI tokens: How to navigate AI's new spend dynamics.* https://www.deloitte.com/us/en/insights/topics/emerging-technologies/ai-tokens-how-to-navigate-spend-dynamics.html

[25] McKinsey & Company. (2026, 24 April). *From promise to impact: How companies can measure—and realize—the full value of AI.* https://www.mckinsey.com/capabilities/quantumblack/our-insights/from-promise-to-impact-how-companies-can-measure-and-realize-the-full-value-of-ai

[26] KPMG. (2025). *Quantifying the GenAI opportunity.* https://kpmg.com/us/en/articles/2025/quantifying-the-genai-opportunity.html

Acronym List

ADaM – Analysis Data Model: A standard for datasets used in statistical analysis, particularly in clinical trials.

ADRG – Analysis Data Reviewer's Guide: A submission document that explains analysis datasets, key derivations, and related review considerations.

AI – Artificial Intelligence: The simulation of human intelligence by machines, especially computer systems.

AIMS – AI Management System: A structured management system for governing the responsible design, deployment, monitoring and improvement of AI systems.

API – Application Programming Interface: A defined interface that allows software systems or services to communicate and exchange data.

CAPA – Corrective and Preventive Action: Procedures for addressing and preventing the recurrence of problems.

CDAs – Confidential Disclosure Agreements: Agreements that protect confidential information shared between parties.

CDISC – Clinical Data Interchange Standards Consortium: An organisation that develops data standards for clinical research.

CI/CD – Continuous Integration/Continuous Deployment: Practices for automating the integration and deployment of code.

CIA – Confidentiality, Integrity and Availability: Core information security principles used to assess and protect data and systems.

CLI – Command Line Interface: A text-based interface used to interact with software or systems through commands.

COU – Context of Use: The circumstances under which a model, tool or output is used.

CRO – Contract Research Organisation: An organisation that provides outsourced research services to pharmaceutical, biotechnology or medical device companies.

CSR – Clinical Study Report: A comprehensive report describing the design, conduct, analysis and results of a clinical study.

CSV – Computer System Validation: Evidence-based confirmation that a computerised system performs as intended in its regulated context.

DLP – Data Loss Prevention: Strategies or tools to prevent unauthorised data sharing or leakage.

EMA – European Medicines Agency: The regulatory agency for medicines in the European Union.

EU – European Union: A political and economic union of European member states.

FDA – Food and Drug Administration: The US agency responsible for regulating food, drugs and medical devices.

GAMP – Good Automated Manufacturing Practice: Guidelines for manufacturers and users of automated systems in pharmaceutical industries.

GenAI – Generative Artificial Intelligence: A subfield of AI that uses large machine learning models to generate data such as text, images or code.

GPAI – General-Purpose AI: AI systems or models designed for broad use across multiple tasks and contexts.

GxP – Good Practice: Regulations and guidelines governing quality in fields such as pharmaceutical, clinical and manufacturing.

HITL – Human-in-the-Loop: A system or workflow where humans provide oversight, review or intervention in AI-assisted processes.

IDE – Integrated Development Environment: A software environment that supports coding, testing, debugging and development workflows.

IEEE – Institute of Electrical and Electronics Engineers: A professional association for electronic engineering and electrical engineering.

InfoSec – Information Security: The discipline of protecting information and systems from unauthorised access, use, disclosure, disruption, modification or destruction.

IP – Intellectual Property: Legal rights associated with creations, inventions, software, data or other proprietary assets.

IQ/OQ/PQ – Installation Qualification/Operational Qualification/ Performance Qualification: Validation activities used to confirm that systems are installed, operate and perform as intended.

ISMS – Information Security Management System: A structured framework for managing information security risks, controls, and continuous improvement.

ISO – International Organization for Standardization: An international standard-setting body.

ISPE – International Society for Pharmaceutical Engineering: A professional association focused on pharmaceutical engineering, manufacturing and quality practices.

LLM – Large Language Model: An AI model trained on large text datasets to generate and process human-like language.

MCP – Model Context Protocol: A standardised interface for context exchange in LLM agent systems.

ML – Machine Learning: A branch of AI in which systems learn patterns from data to make predictions or support decisions.

MLOps – Machine Learning Operations: Practices for deploying, monitoring, managing and maintaining machine learning systems in production environments.

MSA – Master Service Agreement: A contract that defines general terms and conditions for services provided between parties.

NIST – National Institute of Standards and Technology: A US federal agency that develops technology, metrics and standards.

OSS – Open-Source Software: Software with source code that is made available for use, review, modification and distribution under defined licence terms.

PHI – Protected Health Information: Any information about health status, provision of healthcare, or payment for healthcare that can be linked to an individual.

PII – Personally Identifiable Information: Information that can identify an individual.

QA – Quality Assurance: Systematic processes to ensure quality requirements are fulfilled.

QC – Quality Control: Procedures to ensure the quality of products or outputs.

QMS – Quality Management System: A formalised system of policies, processes, procedures and responsibilities for achieving quality objectives.

RAG – Retrieval Augmented Generation: An AI technique that retrieves external information to supplement model outputs.

SAP – Statistical Analysis Plan: A document detailing the statistical methods and analyses to be used in a clinical trial.

SCE – Statistical Computing Environment: An environment or system used for statistical programming and analysis.

SDLC – Software Development Life Cycle: The process of planning, creating, testing and deploying software.

SDRG – Study Data Reviewer's Guide: A submission document that explains study data structure, standards implementation, and review considerations.

SDTM – Study Data Tabulation Model: A clinical data standard for organising and formatting data for submission to regulatory agencies.

SME – Subject Matter Expert: An individual with deep expertise in a specific domain, process or technical area.

SOP – Standard Operating Procedure: Detailed written instructions to achieve uniformity in performing specific functions.

SOPLite – Simplified Standard Operating Procedure: A concise, implementable standard operating procedure.

SOW – Statement of Work: A document describing the scope, deliverables, timelines and responsibilities for a specific engagement or project.

TFL – Tables, Figures and Listings: Common outputs produced during statistical analysis and reporting in clinical trials.