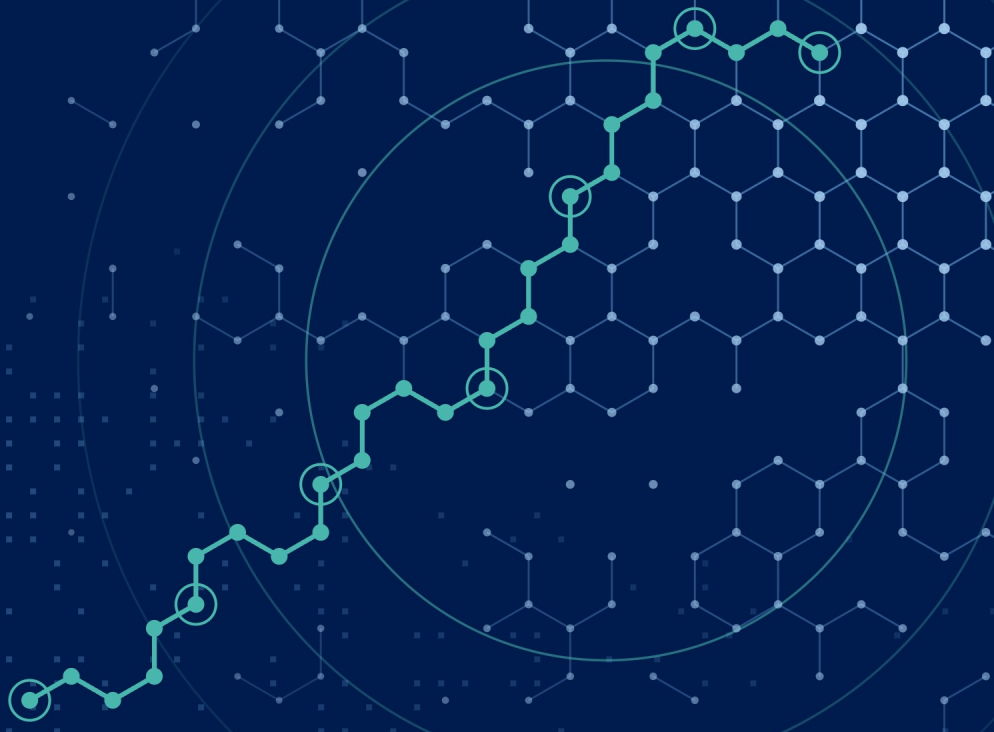
PROGRAMMING HEADS COUNCIL



THE PHC MANIFESTO

Programming Vision on AI

The Programming Heads Council Manifesto

A concise action agenda for statistical programming leaders

MANIFESTO STATEMENT

Statistical programming is moving towards an AI-enabled operating model in which programmers act as **data actors, evidence architects, accountable reviewers and workflow orchestrators**. We will embed AI into programming flows, strengthen data lifecycle expertise, and build professionals who use AI proactively while preserving scientific rigour, regulatory trust, accountability, and deep understanding of clinical trial data.

01 Define programmers as data actors

- Position programmers as **data actors and evidence architects** who understand data sources, transformations, limitations, derivations, downstream use and decision risk.
- **Move from output factory to data solution provider:** TFLs are one data solution, not the full identity of programming.
- **Own the data lifecycle:** study setup, data capture, standards, metadata, derivations, output review, interpretation support and regulatory/ad hoc response.
- Bring programming earlier into data capture, endpoint and analytical question discussions so downstream evidence is feasible, reliable and decision-relevant.

Make "can we do this?" SECONDARY TO

"should we do this, what evidence does it create, and what decision will it support?"

02 Train for effectiveness, not only efficiency

- **Teach effectiveness over efficiency:** AI may automate speed, while programmers create value through better questions, better data, better interpretation and better decisions.
- Use **study read-outs** where programmers explain the science, statistics, data limitations, assumptions, risks and consequences of findings.
- Build **manoeuvre-style learning** through simulated studies, study read-out rehearsals, dry-run reviews, scenario-based debugging, and mock authority questions.
- Preserve meaningful early-career exposure to real clinical trial data and decision contexts, even as AI automates implementation work.
- Use **rotations** through study setup, data management, statistics, medical writing and regulatory response to build end-to-end judgement.
- Make every programmer own a **development plan** covering AI literacy, data lifecycle mastery, critical thinking, stakeholder communication, and evidence interpretation.

03 Embed AI into programming workflows

- Integrate AI into the **statistical computing environment** and approved data science ecosystem, with secure access controls, audit trails, governed repositories, and validated packages.
- Use AI for specification review, code generation, company and regulatory standards checking, output comparison, anomaly detection, knowledge retrieval, and review documentation.
- Design workflows with provenance: source inputs, prompts or context where relevant, model or tool versions, generated outputs, review evidence, approvals, and change history.
- Use **producer-validator and exception-based patterns** where appropriate, while retaining human accountability for quality, scientific meaning and decision impact.
- Prioritise validation and quality review alongside generation, using **AI as a quality amplifier** that reduces repetitive checks and focuses human attention on plausibility, risk and exceptions.

04 Build the foundations for scale

- Make metadata, standards, machine-readable protocols, SAPs and specifications, derivation rules, controlled terminology and governed repositories **strategic assets**.
- Move away from fragmented document-centric flows where PDFs, spreadsheets, code, metadata and assumptions live in disconnected places.
- Create accessible, approved repositories for prior code, regulatory questions, ad hoc requests, standards, reusable patterns, and documented organisational knowledge.
- Build **AI-enabled programming platforms** with secure data access and traceability, and measure value through:

Quality	Cycle time	Rework reduction	Capacity	Adoption	Audit readiness	Stakeholder impact

05 Lead the change: proactive over reactive leadership

- Treat AI adoption as a **leadership agenda and operating-model transformation**, not a technology side project.
- As programming leaders, create urgency by showing how routine work will change and how programmers can own their development and professional relevance.
- Connect the programming community with statistics, data management, data science, medical writing, regulatory, PHUSE and other AI communities to align strategic and operational guidance.
- Make the **data actor and evidence architect roles** explicit in role descriptions, training, leadership expectations and career frameworks.

OUR PRINCIPLES

Operate by manifesto principles

Effectiveness	OVER	efficiency
Architect	OVER	executor
Data actor and evidence architect	OVER	output producer
Accountable reviewer	OVER	passive approver
Embedded, governed AI	OVER	side tool
Data readiness and standards	OVER	model chasing
Personal development ownership	OVER	waiting for training