

White Paper:

The Quality Layer You Can Defend to an Auditor

October 2026

The Quality Layer You Can Defend to an Auditor

Beaconcure · October 2026

The consequences of clinical data-quality issues are not always immediately evident. An error may go undetected until it results in a regulatory review finding, a submission delay, or a publication based on inaccurate information. At the regulatory submission stage, even minor delays can result in significant financial or strategic implications.

A regulatory submission can include hundreds of tables, listings, and figures (TLFs) that summarize the evidence regulators use to evaluate the safety and effectiveness of a new treatment. Ensuring the validity and integrity of these outputs is a critical component of the clinical development process.

For decades, double programming has been the cornerstone for clinical output validation. This process involves two programmers independently producing the same output, comparing the results, and adjudicating discrepancies. This approach is labor intensive, time consuming, and does not guarantee high quality results.

The situation: a hidden cost on every study

Double programming and manual QC quietly consume significant cost and time. Yet the spend never appears as a single line item that leadership can see or act on while being viewed as the “Gold Standard” for decades.

A 2025 scoping review of 262 publications puts the burden plainly: a typical Phase III study needs 200–500 TLFs, and double programming consumes **1.6–2.0x the original programming effort (based on CRO pricing data)** and extends timelines by **20–40%** ([Yan et al., medRxiv](#)). This multiplier effect is the result of the iterative refinement cycles required to compare results, identify the source of the discrepancy, and resolve the issue(s).

The financial implications can add up fast. Outsourced programming and validation of statistical tables is commonly priced at approximately \$1,000–\$2,000 per table. For a typical Phase III study with 200–300 tables, that is \$200,000–\$300,000. That cost does not include figures, listings, or dry runs.

Extrapolating across a clinical development program and the costs to develop and validate statistical outputs can run well into seven figures. ***Yet the fundamental limitation remains, double programming creates a false sense of security but does not guarantee error-free results.***

The problem: creating a false sense of security

Double programming doubles the work, increases costs and timelines, and does not guarantee all errors get caught.

Two programmers agreeing isn't proof you're right. It may just mean you made the same assumption twice.

Not all errors are created equal, and not all errors can be detected through double programming.

Some errors can originate within the programming specifications themselves, resulting in two independently developed programs that may produce identical incorrect results.

Even when the specifications are correct, two programmers can make the same mistake. If both misinterpret the Statistical Analysis Plan (SAP), the table shell, or derivation rules, their outputs may agree despite being incorrect. A 2019 American Statistical Association proceedings paper examined this phenomenon of "undetectable errors" demonstrating how the likelihood of shared errors increases when programmers share similar training, experience, and programming practices ([JSM 2019](#)).

The effectiveness of double programming is predicated on the independence of the validation process. This independence can be compromised if the second programmer copies code from the original programmer, which can be difficult to detect. A 2026 PharmaSUG China paper describes a tool specifically designed to detect code copied between production and QC teams ([Zeng](#)), highlighting the practical challenges of maintaining programming independence.

Ultimately, two programmers producing identical results establishes agreement - not necessarily correctness.

Identifying a discrepancy is not the same as understanding the root cause. When two independently programmed outputs differ, double programming identifies *what* is inconsistent but does not provide insight into *why* the inconsistency exists. Determining the root cause requires programmers to evaluate the underlying code, programming logs and specifications, calculations and derivations to isolate the source of the discrepancy.

This investigation can become an iterative process of comparing outputs, reviewing programming logic, modifying code, and regenerating outputs. Even a single discrepancy can result in multiple rounds of reprogramming before the underlying issue is identified and remediated.

Consequently, the cost of double programming extends beyond the effort required to generate statistical outputs. The iterative reconciliation process introduces considerable effort, consumes the time of valuable resources, and can result in unpredictable delays in delivering final TLFs.

An industry convention should not be mistaken for a regulatory requirement. For decades, double programming has been considered the “gold standard” approach for validating statistical outputs. The widespread adoption has created the perception that independent double programming is a regulatory requirement. Recent PharmaSUG publications highlight that double programming is not a regulatory requirement and that blanket application often yields diminishing returns ([PharmaSUG 2026 DS-111; Clinical Leader](#)).

This raises an important question: if double programming is not a regulatory requirement why does it continue to be viewed as the “gold standard” for validating statistical outputs?

Part of the answer lies in established standard operating procedures, organizational aversion to risk, and the perception that independently programmed outputs provide the strongest evidence of a valid QC. While double programming can be effective at identifying discrepancies, applying the same approach to every output assumes that all analyses carry the same risk and warrant equivalent validation effort. They do not.

Ultimately, our goal is to maximize confidence in the accuracy of statistical outputs and provide evidence of what constitutes a valid QC.

The implications: The cost of standing still

Maintaining the status quo is not a neutral choice. Clinical trials are becoming increasingly complex, generating exponentially more data, and face mounting development costs. Traditional validation methods are becoming more difficult to scale efficiently.

Several industry trends are amplifying this challenge:

- **More data.** According to research from Tufts CSDD and TransCelerate, Phase III protocols now collect approximately 5.9 million data points on average, representing an annual growth of roughly 11% since 2020. A substantial increase from fewer than 1 million in 2012 ([Tufts CSDD / TransCelerate, TIRS 2025](#)).
- **Greater analytical complexity.** Phase III protocols now average approximately 19 endpoints at database lock, an increase of roughly 30% over the past 15 years ([Tufts CSDD / TransCelerate, TIRS 2025](#)). Additional endpoints often introduce more complex reporting requirements, calculations and derivations. Thus, creating additional opportunities for errors in specifications, programming, and discrepancies that require investigation and remediation.
- **Escalating cost per study.** Traditional double programming requires significant effort to reproduce and adjudicate discrepancies. As the number and complexity of statistical tables increases, so does the associated programming and validation workload. Protocol amendments and SAP revisions can further compound these costs by requiring additional iterations.

- **Delays increase financial exposure.** The financial consequences of late-stage quality issues can far exceed the costs of validation. Tufts CSDD estimates that a day of development delay can represent approximately \$500,000 in unrealized sales and an additional \$40,000 in direct clinical trial costs ([Applied Clinical Trials](#)). The financial impact can vary significantly by development stage and therapeutic area. An error discovered late in the development process can result in a significant financial and strategic impact if it causes delays in time to market.

These trends reveal a fundamental challenge. As clinical trials become more data intensive and analytically complex, validation of statistical outputs that rely on double programming becomes increasingly costly and difficult to maintain. Validation methods need to evolve to meet the challenges of today's clinical trials.

Introducing Verify Intelligence: AI that shows its work

What if validating statistical outputs no longer relied upon double programming, and identifying discrepancies was accompanied by an explanation of its underlying cause?

Verify Intelligence introduces an independent, AI-enabled quality and validation layer designed to improve quality, efficiency, and transparency of statistical output validation.

Verify converts clinical trial data quality from an invisible cost into a visible, reducible one. It protects your submission from the errors conventional validation fails to catch, at a fraction of what double programming costs today.

Rather than requiring statistical programmers to independently recreate statistical tables, Verify Intelligence independently regenerates statistical tables directly from ADaM datasets, SAP, Mock Shells and Specs and systematically compares them against production outputs. Reviewers get a complete picture of where the two versions diverge, and why. Additionally Verify helps the programmer and statistician to get a clear snapshot of specs alignments between all documents.

Verify is an independent quality layer. It complements your existing programming and QC workflows rather than asking you to rebuild them. And it is AI that shows its work: every finding links to a specific, source-checkable cause, not a generic "AI-powered" verdict. Verify generates a full meta data report that allows SME to understand immediately what is the quality status of the tables.

How it works:

1. **Regenerate – Independently reproduce statistical tables.** Verify Intelligence creates its own version of each statistical table directly from the ADaM datasets, independent of the production code.

2. **Compare – Systematically identify discrepancies.** The independently generated statistical tables are compared to the production outputs at the individual datapoint level, automatically identifying inconsistencies.
3. **Explain – Investigate the underlying cause.** Rather than simply highlighting that two results differ, Verify Intelligence traces the root cause to the code, the assumptions, or the source data, and documents the findings.
4. **Remediate – Resolve issues within a traceable workflow.** Reviewers can assess findings, determine appropriate corrective action, and regenerate statistical tables as needed. The process maintains an audit trail of the investigation and resolution thereby providing evidence of a valid QC process.

Focused Capabilities, Meaningful Impact

Verify Intelligence currently supports several of the most commonly produced domains of statistical tables, including demographics, disposition, adverse events, and exposure/compliance.

This focused approach is by design. Rather than promising universal coverage, Verify Intelligence concentrates on the most frequently produced statistical tables where independent validation can be automated and where repetitive statistical programming and reconciliation consumes significant time and resources.

What it means for biometrics teams:

The benefits of Verify Intelligence extend well beyond the automation of statistical table validation. The platform has the potential to optimize how specialized biometrics resources are deployed and how organizations establish confidence in statistical results that can withstand regulatory scrutiny.

- **For statistical programmers: Less repetitive programming, more analytical problem-solving.** Stop rebuilding standard tables from scratch and stop hunting for why two files disagree. They review findings that already come with a cause.
- **For biostatisticians: Greater visibility to accuracy of results.** Not every discrepancy represents a programming error. By providing supporting evidence for each discrepancy, Verify Intelligence helps biostatisticians distinguish between cosmetic errors from issues requiring more in-depth statistical involvement.
- **For quality and operational leadership: Greater transparency and evidence.** Traditional QC processes generate documentation that demonstrates validation activities were performed and discrepancies were resolved. Verify Intelligence strengthens that documentation by linking identified findings with supporting evidence and resolution decisions.

Verify Intelligence, Beaconcure's newest module, is in production at a Top 10 global pharmaceutical company with three other Top 10 companies beginning pilots in the new year. Developed in close

collaboration with biometrics professionals who use the Verify platform, the solution addresses the evolving challenges of validating statistical tables in today's rapidly evolving clinical development landscape.

Beaconcure's approach is grounded in deep domain expertise, combining decades of experience in biostatistics and statistical programming. Verify is built by biostatisticians and clinical programmers: the people who used to sign off on TLF packages. With seven years in business, more than 500 studies validated by Verify, and trusted by Top 10 biopharmaceutical companies.

The payoff: lower risk, lower cost, better use of experts

Verify Intelligence significantly reduces, and eventually, may even eliminate human manual double-programming QC. It replaces it with independent, deterministic, and AI-assisted validation that catches what conventional review structurally misses. Every finding is explainable and traceable back to source.

	Double programming today	With Verify Intelligence
Effort	1.6–2.0x the original programming	Independent regeneration replaces the second manual build on in-scope tables
Shared mistakes	Pass undetected when both programmers error the same way	Independent check that does not inherit the production programmer's code or coding choices
Discrepancies	Flagged, then traced by hand across files	Flagged automatically, with the root cause explained
Fixes	Re-program, re-run, re-compare	Remediate and recalculate in the same session
Audit trail	Scattered across programs, logs, and emails	One documented, traceable record per finding
SAP changes	No fast way to see the blast radius	Affected tables re-checked in one workflow
Quality of ADaM	ADaM issues typically surface only downstream, when TLF outputs disagree or during late review	ADaM data issues are flagged as soon as tables are generated, before they propagate into the TLF package

The Organizational Impact: Greater Confidence, Lower Costs, and Improved Scalability

1. **Risk becomes visible and manageable.** The value starts with risk avoided: inspection findings, resubmissions, and late-stage discrepancies that never reach a regulator. An invisible, potentially catastrophic risk becomes a visible, managed, and reducible one, at a fraction of today's QC cost.

2. **Quality becomes more integrated and efficient.** Discrepancies are flagged automatically and explained, with no more hunting across files to see why two versions don't match.
3. **Specialized expertise is redirected toward higher-value activities.** Programmers and statisticians spend less time reconciling files and more time on the science.

The Business Case For CROs: Margin Improvement, Scalability, and Competitive Differentiation Verify doesn't ask sponsors to change how they work. It makes the QC you already own faster, less expensive and easier to defend.

Redefining Confidence in Statistical Output Validation

For decades, the industry has accepted double programming as the method for establishing confidence in the accuracy of statistical outputs. While this approach has served an important purpose, it is resource-intensive, difficult to scale, and not infallible.

Verify Intelligence represents a shift from labor intensive, duplicative programming toward automated, scalable, and evidence-driven validation of statistical tables.

The promise of Verify Intelligence: less time producing and validating results, more confidence in their accuracy, and a validation process built to withstand regulatory scrutiny.

See it on your own tables. Talk to the Beaconcure team about running Verify Intelligence against a completed study and comparing what it finds with your existing QC.

Sources

- Yan, Zhang, Tian. [Evidence Behind the Automation of Clinical Trial Statistical Programming: A Scoping Review \(2020–2025\)](#). medRxiv preprint, not peer reviewed.
- [Dual Programming Validation for Analysis: undetectable error](#). JSM Proceedings 2019, American Statistical Association.
- Gupta. [PharmaSUG 2026 Paper DS-111](#) and [PharmaSUG 2025 Paper MM-227](#).
- [Is Double Programming Really Required For Validation?](#) Clinical Leader, March 2025.
- Zeng. [DupChecker: detecting code duplication in double programming](#). PharmaSUG China 2026.
- Getz et al. [TransCelerate–Tufts CSDD study on clinical trial data volume](#). Therapeutic Innovation & Regulatory Science, December 2025.
- Getz. [How Much Does a Day of Delay in a Clinical Trial Really Cost?](#) Applied Clinical Trials, June 2024.