WORKING PAPER · 2026

Structural Risk Intelligence for Clinical Trial Software

A practical guide for development, quality, and regulatory teams

Abhishek Srivastava
abhishek@perform.digital
spiderbrain.ai

Engine validation: v5.1 / decision layer v5.0.7 · 2026-05-30
v5.5 re-validation: unified scorer 0.5.5 · 2026-06-18
Spiderbrain · spiderbrain.ai · contact@spiderbrain.ai

A practical guide for development, quality, and regulatory teams

Working paper · Spiderbrain v5.5 · 2026-06-18

Author: Abhishek Srivastava | abhishek@perform.digital | spiderbrain.ai

Abstract

Clinical trial software carries regulatory obligations that most commercial software does not. A defect in an Electronic Data Capture (EDC) system can compromise trial data integrity. A misconfigured safety processing component can delay a serious adverse event report. An undocumented dependency in a submission system can invalidate a marketing authorisation application.

This paper addresses a specific, measurable problem: before a software change is made, can we know exactly what it will break and how seriously? Across six clinical trial datasets drawn from ClinicalTrials.gov (oncology, cardiology, diabetes, mental health, neurology, and vaccines), Spiderbrain's risk scoring concentrates signal by 1.48x to 2.14x above the average component, while operating under data privacy constraints that are compatible with regulated environments.

We describe how this approach directly supports GCP and GAMP 5 requirements for change control, impact assessment, and traceability, and map specific capabilities to clinical development workflows.

Why Spiderbrain for Clinical Trial Software

Benefit	What it means for your team
Know what breaks before you change it	Every change comes with a scored, ranked list of what else is affected and how severely, produced automatically from the system's actual structure
Regulatory-ready audit trail	Every risk assessment is recorded in a tamper-evident, time-stamped log aligned with 21 CFR Part 11 audit trail requirements
No source code leaves your environment	Only a privacy-safe structural map is transmitted for scoring; source code, variable names, and data content never leave your organisation
Same answer every time	Scores are fully deterministic: the same system produces the same scores on every run, making them usable as the basis for documented regulatory decisions
Find hidden-critical components before inspection	The blind-spot index identifies components that have become structurally important without being documented as such, which is the type of gap most likely to cause an inspection finding
Works on systems of any scale	Validated on systems from 219 to 2,000 components; designed for the 5,000 to 20,000 component systems common in CTMS and EDC platforms
Makes AI coding tools safer in regulated environments	When an AI tool has Spiderbrain's risk map, it can warn before modifying a high-impact component and log that warning as part of the audit trail
Architecture diagrams from the actual system	Blueprint auto-generates a diagram of your system's structure from the live component map, not from a developer's memory of how it used to look
Detect drift from the validated state	Drift detection surfaces components that have moved structurally away from their historical position: the equivalent of an undocumented protocol deviation at the system level
Risk-based validation planning with objective scores	The keystone list provides a structurally-derived priority ranking for testing effort, defensible to a regulatory inspector and practically efficient

1. Introduction

Clinical trial software is among the most highly regulated software in existence. A defect in an EDC system can compromise data integrity and require a protocol deviation report. A

misconfigured safety signal processing component can delay a serious adverse event report and trigger regulatory action. An undocumented connection in a submission system can invalidate a marketing authorisation application.

Systems in this category operate under strict requirements:

- **21 CFR Part 11** (US FDA): electronic records, electronic signatures, and audit trail requirements.
- **EMA Annex 11** (EU): computerised systems in clinical trials, including change control and impact assessment.
- **GCP E6(R3)** (ICH): risk-based quality management, with explicit requirements for system validation documentation.
- **GAMP 5**: a risk-based approach to software validation, requiring categorisation by software type and commensurate impact assessment for every change.

A mature clinical trial management system can contain thousands of interconnected components built up over decades. In this environment, the question "what does this change break?" is not a convenience; it is a regulatory obligation.

AI tools have become embedded in modern development workflows. They carry a structural deficiency that is particularly damaging in regulated environments: they have no persistent memory of system structure. At the start of every session, the tool must re-examine the system from scratch. It cannot reliably tell you which component is load-bearing, trace the impact scope of a change, or flag that a shared configuration component is critically important until after a change has already caused harm.

Spiderbrain addresses this by building and maintaining a structural map of the system, scoring every component by severity, impact scope, and structural position, and making that map persistently available to development teams and AI tools. This paper examines why this approach is particularly valuable for clinical trial software and where it fits into regulated development and validation workflows.

---

2. The Structural Complexity of Clinical Trial Software

2.1 System Types in Clinical Development

Clinical development uses several distinct software categories, each with a characteristic structural profile:

Electronic Data Capture (EDC) systems manage case report form data from clinical sites. They are large multi-user applications with complex branching logic, real-time data

validation, regulatory audit trails, and integrations with laboratories and imaging systems. A change to a shared validation component can propagate to hundreds of study-specific forms.

Clinical Trial Management Systems (CTMS) coordinate the operational infrastructure of a trial: site management, payments, enrolment tracking, regulatory documents, and deviation tracking. Their component maps are wide rather than deep, with many modules sharing common services.

Electronic Common Technical Document (eCTD) platforms produce the structured regulatory submission that goes to the FDA or EMA. These systems handle document assembly, version control, region-specific rule checking, and lifecycle management across thousands of documents. Structural risk is concentrated in the assembly and validation pipeline, where a defect can silently corrupt a submission.

Safety and pharmacovigilance (PV) databases process individual case safety reports (ICSRs), aggregate signal detection, and produce periodic safety update reports. Their impact-scope risk is among the highest in clinical software: a shared case processing component that silently fails can delay expedited safety reporting.

Statistical Analysis Systems produce the analysis datasets and outputs that form the clinical study report. Reproducibility is a regulatory requirement; any non-determinism or undocumented dependency is a validation deficiency.

2.2 The Change Control Problem

GCP E6(R3) Section 5.8 requires that changes to validated systems be controlled and documented. GAMP 5 adds the requirement for impact assessment: before any change is implemented, the organisation must evaluate what it affects and what re-validation may be required.

In practice, this is done manually. A developer examines the changed component, consults documentation, and lists the components they believe are affected. This process is:

- **Incomplete:** undocumented connections and indirect relationships are routinely missed.
- **Inconsistent:** the quality of impact assessment depends on individual developer knowledge.
- **Slow:** in regulated environments, a manually-produced impact assessment can take days.
- **Unverifiable:** there is no deterministic link between the written impact assessment and the actual system structure at the time of the change.

Automated impact scoring closes each of these gaps.

2.3 The Traceability Gap in AI-Assisted Development

As development teams adopt AI tools, a new regulatory risk has emerged. AI tools that generate, modify, or review components in regulated systems must themselves be part of the validation framework. When an AI tool makes a change without understanding the structural role of the affected component, it can introduce defects in shared critical components while correctly solving the stated task.

The specific failure mode: the AI tool makes a locally correct change to a component, without knowing that the same component is also relied upon by many other parts of the system with incompatible assumptions. A structural map with impact-scope scoring would have flagged this before the edit.

---

3. How Spiderbrain Works in Regulated Environments

3.1 Data Privacy Model

Regulated clinical trial software almost always operates under confidentiality obligations: to the sponsor, to trial participants, and to the competitive position of the therapy under development.

Spiderbrain's architecture addresses this directly. Analysis runs on the development machine. What is transmitted to the cloud scoring service is a **privacy-safe structural map**:

Transmitted	Not transmitted
Component names and relative paths within the system	Source code or file contents
Structural connections between components	Commit messages or author identities
A cryptographic fingerprint of each component	Any data content or variable names
Privacy-safe timing signals from version history	Author email addresses

The structural intelligence is available to the development team without exposing regulated data. For environments where even component names are sensitive, a path-anonymisation option can be applied before transmission; the scoring works on structural relationships regardless.

Data at rest is stored in the EU, under GDPR and India DPDP frameworks.

3.2 Deterministic Scoring

All Spiderbrain scores are deterministic: given the same system map, the engine always produces the same scores. There is no random or probabilistic element. The same requirement applies to statistical analysis systems: a score that changes between runs without an underlying data change cannot be used as the basis for a documented impact assessment.

Every score is traceable to the system structure, with a clear path from raw connection counts through community grouping, reach calculation, and necessity estimation.

3.3 The Tamper-Evident Audit Log

Spiderbrain records every risk assessment in an append-only, hash-chained log. Each entry records the timestamp, the action, the component assessed, the risk scores before and after, and a cryptographic link to the previous entry. Tampering with any entry invalidates the chain and is detectable.

For regulated environments, this log serves as the technical audit trail for the tool's role in the development process: evidence that structural analysis was performed, and exactly what it found, at the time a change was assessed.

This is directly aligned with 21 CFR Part 11 §11.10(e): "Use of computer-generated, time-stamped audit trails to independently record the date and time of operator entries and actions that create, modify, or delete electronic records."

3.4 Risk Scores: What They Mean in Clinical Terms

Every component in the system map receives a set of scores. In plain terms:

Criticality rating: How much it would hurt if this component failed on its own. A shared data validation component rates high; a study-specific configuration file rates lower.

Impact scope: If this component fails, how much of the system fails with it. A shared configuration component may have a moderate criticality rating but a very high impact scope: it seems unremarkable until it fails, at which point it brings down many unrelated parts of the system. Impact scope is the pre-change impact assessment.

Combined risk: Criticality and impact scope in one number. This is the "how worried should we be?" score for regulatory documentation.

Keystones: The small number of components that everything else relies on. In a large clinical system there are typically 100 to 300 keystones. They also act as containment boundaries: damage that reaches a keystone tends to stop rather than cascade further,

because keystones are structurally separable. Identifying keystones allows the validation team to concentrate re-validation effort where it has the highest leverage.

Blind-spot index: Components that are structurally important but poorly examined or documented. In clinical software, a component that quietly became load-bearing while no one was watching is exactly the kind of gap that causes inspection findings. Blind-spot scoring makes this visible before it becomes a problem.

Drift: Components that have changed structurally in ways that no longer match their history. In a clinical system, structural drift is the equivalent of an undocumented protocol deviation: the system has moved away from its validated state.

---

4. Validation Results

4.1 Datasets

Six datasets were constructed from ClinicalTrials.gov trial metadata, each covering a distinct therapeutic area with 1,000 trial records. Connection counts reflect the density of relationships across real clinical development networks.

Therapeutic Area	Records	Connections
Oncology	1,000	4,836
Cardiology	1,000	3,788
Diabetes	1,000	3,232
Mental Health	1,000	4,641
Neurology	1,000	4,099
Vaccines	1,000	4,138

Connection density is highest in oncology (4.84 per record) and mental health (4.64 per record), reflecting the complex multi-arm, multi-intervention structures common in those therapeutic areas.

4.2 Signal Concentration (v5.1 Engine, 2026-05-30)

The primary metric is the **risk concentration ratio**: the mean risk score of the top 10% of records divided by the population average. A ratio of 2.0 means the highest-risk

components score twice as high as a typical component. This metric is independent of dataset size.

Therapeutic Area	Risk Concentration	Improvement over average
Diabetes	1.60x	+60%
Cardiology	1.59x	+59%
Oncology	1.57x	+57%
Mental Health	1.56x	+56%
Neurology	1.54x	+54%
Vaccines	1.48x	+48%

All six therapeutic areas show consistent risk concentration (1.48x to 1.60x), indicating that the scoring generalises across different clinical trial structures.

4.3 Updated Results on v5.5 Engine (2026-06-18)

The same six datasets were re-scored on the current v5.5 engine. Risk concentration improved substantially across all areas:

Therapeutic Area	v5.1	v5.5	Change
Oncology	1.57x	**1.78x**	+0.22
Cardiology	1.59x	**2.14x**	+0.55
Diabetes	1.60x	**2.07x**	+0.47
Mental Health	1.56x	**2.04x**	+0.48
Neurology	1.54x	**1.97x**	+0.42
Vaccines	1.48x	**1.55x**	+0.07

Mean risk concentration: 1.56x under v5.1, 1.91x under v5.5. Structural community grouping was successfully identified on all six datasets, finding 183 to 260 distinct component clusters per dataset.

4.4 Reliability Guarantees

The scoring system was validated against five guarantees on each dataset:

A. Identity safety: Renaming, editing, and deleting components in a test sample produced zero silent corruptions across 360 operations.

B. Collision safety: Where two components share the same structural fingerprint, deleting one did not cause the system to silently attach its risk assessment to the surviving component. Result: zero silent misattachments.

C. Score validity: All risk and reach scores computed correctly on real data. Result: zero invalid values across all six datasets.

D. Determinism: Scores are stable regardless of processing order, verified across three independent runs. Result: identical output on every run.

E. Partitioning invariant: Splitting the ranked output does not change the underlying signal concentration. Result: zero discrepancy after partition.

Guarantees A through E represent the minimum reliability framework for using Spiderbrain scores as the basis for regulatory documentation.

---

5. Clinical Development Applications

5.1 Pre-Change Impact Assessment (GCP Change Control)

The regulatory requirement: Before implementing a change to a validated system, the organisation must document which other components are affected and determine whether re-validation is required.

How Spiderbrain helps: Before any change, the developer or AI tool requests the impact scope of the target component. The result is a scored, ranked list of what is affected. The impact assessment can be generated directly from this output: "This change to the data entry validation component (criticality 8.4, impact scope 7.1, combined risk 7.9) affects 23 downstream components, of which 4 have combined risk above 6.0."

This replaces a manual review with a deterministic, reproducible output. The log entry for the assessment is tamper-evident, timestamped, and verifiable.

Where it matters most: Shared validation components in EDC systems, common configuration in CTMS, base processing components in safety case databases, statistical macros shared across multiple analysis programs.

5.2 Blind-Spot Audit Before a Regulatory Submission

The problem: A submission or product dossier depends on software that has been in use for years. Some components that were once peripheral have become structurally critical as the system evolved. These are not documented as critical; they simply became critical.

How Spiderbrain helps: The blind-spot index surfaces components with high combined risk and low examination or documentation history. In a pre-submission audit, this list becomes a targeted review queue: the validation team examines each high blind-spot component, confirms or updates its documentation, and closes the gap before the submission.

This is not hypothetical; the validation gap most often identified in inspection findings is not in the originally-documented critical components. It is in components that were missed because no one knew they mattered.

5.3 Drift Detection Between Validated and Current State

The regulatory requirement: GAMP 5 requires that the validated state of a system be maintained, and that any deviation from that state be detected, evaluated, and documented.

How Spiderbrain helps: After any development period, the drift index surfaces components that have moved away from their historical structural position (not just components that changed, but components whose change has structural consequence). This is the distinction between a change log and a risk-prioritised deviation list.

For example: a utility used by two components and now used by fifteen has changed structurally even if its own content is unchanged. Its impact scope will have increased. Drift detection flags this for re-evaluation.

Where it matters most: Any gap between validation cycles in a system under active development. Long-running Phase III trials often accumulate changes across three to five year timelines. Drift detection makes the accumulated structural risk visible at any point.

5.4 Keystone Identification for Risk-Based Validation Planning

The GAMP 5 principle: Risk-based validation concentrates testing effort on the components of highest risk. The challenge is determining which components are highest risk; the GAMP 5 guidance points to criticality but does not prescribe how to measure it.

How Spiderbrain helps: The keystone list provides a structurally-derived answer. In a clinical system with 5,000 components, there will typically be 100 to 300 keystones. These are the hubs that, if they fail, carry downstream consequences. They are also containment boundaries: validating keystones correctly limits cascade risk.

A risk-based validation plan built on the keystone list is defensible to a regulatory inspector (because it is derived from deterministic structural analysis) and practically efficient (because it concentrates effort where it has the highest leverage).

5.5 AI-Assisted Development with Structural Safeguards

The problem: AI tools used in regulated development must understand structural risk before making changes. An AI tool that modifies a high-criticality shared component without knowing its impact scope is operating without appropriate safeguard.

How Spiderbrain helps: With Spiderbrain connected, the AI tool's working context for every change includes the structural position of the target component: its risk scores, its keystone status, and the number and severity of its dependencies. The AI tool's proposed change is structurally reviewed before it is presented to the developer.

The impact-scope alert is shown before any change is applied, requiring developer approval. For regulated systems, this approval event is recorded in the tamper-evident audit chain.

5.6 Architecture Documentation for Regulatory Review

The problem: Regulatory submissions and Annex 11 impact assessments sometimes require a description of system architecture. These descriptions are typically written from memory or from diagrams that do not reflect the current state of the system.

How Spiderbrain helps: Blueprint auto-generates an architecture diagram from the live structural map. The resulting diagram reflects the actual system structure at a specific point in time. For regulatory documentation, a Blueprint exported after each structural assessment is a point-in-time architecture record, directly usable in validation documentation.

5.7 Shared Risk Visibility for Multi-Vendor Trials

The problem: Large clinical trials typically involve multiple vendors: the EDC vendor, the CTMS vendor, the central laboratory data integrators, and the imaging CRO. Each holds a fragment of the system. When something breaks at an integration point, no single vendor has visibility into the full picture.

How Spiderbrain helps: Cortex gives a cross-vendor team one shared structural map: a single deterministic view of the integration landscape, with impact scores across the full dependency surface. When the EDC vendor changes an interface component, the CTMS vendor's shared map shows immediately which of their integrations are affected and how severely.

---

6. Limitations and Honest Assessment

The following limitations are relevant to clinical trial deployment:

Internet connection required. Spiderbrain analyses locally but scores via EU cloud servers. A fully air-gapped development environment cannot use the cloud scoring service. For teams requiring air-gapped operation, this is a blocking limitation that requires discussion under the Enterprise plan.

Statistical programming languages have partial coverage. SAS (dominant in statistical programming) and R receive a lighter structural analysis that may miss some dependencies. Teams with SAS-heavy analysis pipelines will see less complete structural maps for those components.

Re-scoring is a manual step. The structural map is only as current as the last explicit rebuild. In a fast-moving development cycle, the risk scores shown may be days behind the current system state. The Pioneer founding tier provides live automatic updates; standard plans require an explicit rebuild trigger.

Desktop application only. The desktop application runs on Windows and macOS. Linux-based development environments are not currently supported.

Closed scoring engine. The scoring engine cannot be self-hosted. For sponsors with strict data residency requirements who cannot accept the EU cloud scoring service, on-premise options are discussed under Enterprise but are not a standard offering.

AI-assisted team merge is in beta. For teams using AI-assisted conflict resolution in the shared structural map, this feature should not be used as a primary mechanism in validated workflows until it exits beta status.

---

7. Compatibility with Regulatory Frameworks

Regulatory Requirement	Framework	How Spiderbrain Addresses It
Change control with documented impact assessment	GCP E6(R3) §5.8, GAMP 5	Impact scope scoring produces a deterministic, reproducible impact assessment for every change
Audit trail for electronic records and AI tool usage	21 CFR Part 11 §11.10(e)	Tamper-evident, hash-chained log of every risk assessment and decision
Risk-based validation planning with criticality classification	GAMP 5 Chapter 4	Keystone list and criticality rating provide a structurally-derived priority ranking
Identification of undocumented or poorly-examined critical components	GAMP 5 retrospective validation	Blind-spot index surfaces high-risk components with low documentation history
Detection of deviation from validated state	GAMP 5 Chapter 7	Drift detection identifies structural changes from the validated baseline
Architecture documentation for submission or inspection	Annex 11 §4.3	Blueprint generates architecture diagrams from the live structural map
Reproducibility and determinism	Statistical analysis requirements	Fully deterministic scoring with verified determinism guarantee
Source data confidentiality	ICH E6(R3), sponsor contractual obligations	Privacy-safe structural map: no source content or data leaves the organisation

8. How Spiderbrain Compares to Existing Approaches

8.1 Manual Impact Assessment

Manual impact assessment is the current regulatory default. It is human-dependent, non-reproducible, and does not scale beyond small systems. It produces no structural evidence of its own completeness: an assessment that misses a dependency cannot be distinguished from one that finds it, until the missed dependency causes a failure.

Automated structural scoring is deterministic and auditable. The assessment covers every connection in the system map, not just the ones the developer remembers.

8.2 Code Quality and Security Scanning Tools

Code quality and security scanning tools identify component-level defects: potential failure points, security vulnerabilities, and non-compliant patterns. They do not produce structural maps or impact scores. They answer "is this component correct?" not "what does this component depend on and what depends on it?"

Structural scoring and code quality scanning are complementary. A high-criticality component identified by structural scoring is also a high-priority target for code quality review.

8.3 Document Search and AI Query Tools

Search-based tools find components that are thematically similar to a query. They do not follow real structural connections, do not compute impact scope, and do not identify keystones. Their results can vary between queries for the same input.

Spiderbrain does not compete with search tools; it is structurally complementary. A clinical team may benefit from both: search to find relevant components, structural scoring to assess the risk of changing them.

---

9. Discussion: Where Clinical Trials Benefit Most

The therapeutic areas with the highest connection density (oncology at 4.84 connections per record, mental health at 4.64 per record) show risk concentration ratios of 1.57x and 1.56x under the v5.1 scoring, rising to 1.78x and 2.04x under v5.5. This confirms that the scoring generalises well across complex structures; high-density networks do not flatten the risk signal.

The practical implication is that the systems most likely to carry hidden structural risk (the shared safety processing pipelines in oncology, the multi-arm adaptive trial systems) are also the ones where Spiderbrain's scoring is most reliable as a triage mechanism.

For EDC systems specifically, the diabetes dataset (1.60x v5.1, 2.07x v5.5) demonstrates the scoring on a structure typical of form-based data capture: moderate connection density with a small number of structurally dominant components. This matches the known architecture of form validation systems, where a small number of shared validation rules are called by a large number of study-specific forms.

The neurology dataset (1.54x v5.1, 1.97x v5.5) shows the lowest concentration ratio in the clinical set. This likely reflects the more fragmented structure of neurology trial protocols (adaptive designs, multiple endpoints, biomarker integration), which distributes criticality

more evenly across components. Even at this level, the signal is substantially above random (1.00x) and provides a useful triage ranking.

---

10. Conclusion

Regulated clinical trial software is an environment where the consequences of structural gaps are not just technical; they are regulatory, safety-related, and potentially patient-harming. The questions that Spiderbrain answers structurally and deterministically (which components are load-bearing, what breaks when one changes, which components have drifted from their validated state, which critical components have gone unexamined) are precisely the questions that current regulatory frameworks require development teams to answer, and that current tools leave to manual judgment.

Across six ClinicalTrials.gov datasets spanning oncology, cardiology, diabetes, mental health, neurology, and vaccines, Spiderbrain's risk scoring concentrates signal by 1.48x to 2.14x above the population average, with deterministic and reproducible results validated against five structural guarantees. The privacy-safe architecture is compatible with clinical trial confidentiality requirements, and the tamper-evident log provides an audit trail for AI tool usage that aligns with 21 CFR Part 11 and ICH GCP requirements.

The most immediate applications in clinical development are: pre-change impact assessment for GCP change control, blind-spot auditing before regulatory submissions, drift detection between validation cycles, keystone identification for risk-based validation planning, and AI-assisted development with structural safeguards. For multi-vendor trial infrastructures, Cortex provides a shared deterministic map across the full integration landscape.

The limitations are real: an internet connection is required for scoring, SAS and R are not fully supported at the structural depth needed for statistical programming environments, and the engine cannot be self-hosted in air-gapped environments without the Enterprise on-premise option. Teams should evaluate these constraints against their specific regulatory and operational context before deployment.

Where the constraints are acceptable, Spiderbrain provides something no current regulatory toolkit offers: a deterministic, auditable answer to the question that every clinical trial software change must answer before it proceeds: "what exactly does this break, and how badly?"

---

11. What Changed in v5.5

The v5.1 validation results in §4 were produced on 2026-05-30. The v5.5 engine, validated on 2026-06-18, extends the capability in several practical areas:

Complete risk profile in the cloud. In v5.1, some scoring layers were computed locally but not retained in cloud storage. In v5.5, the complete risk profile is preserved through upload, so AI tools connected to the system receive a full picture.

Structural community grouping. v5.5 adds automatic grouping of components into structural communities: clusters of closely related components. This is adopted on all six clinical datasets, finding 183 to 260 distinct communities per dataset. Community grouping helps teams understand system architecture at a level above individual components.

Extended audit log. The tamper-evident log now covers changes in risk score level across rebuilds, not just individual decisions. A team can replay exactly how a component's combined risk changed between two assessments.

Architecture diagram (Blueprint). v5.5 ships the Blueprint canvas with automatic architecture derivation from the structural map, directly usable in regulatory documentation.

Join the v5.5 Clinical Trials Programme

We are running a structured programme with clinical development organisations who want to evaluate v5.5 against their own systems under realistic working conditions.

What participation involves:

- A structural risk map built on your system (your source stays on your machine)
- An impact-scope audit of your last three production changes, with a structured report
- A blind-spot review of components likely to be in scope for your next regulatory submission
- A debrief to discuss findings and applicability to your validation framework

Participation is at no cost during the programme period. Findings are shared with your team; no system data is used outside your engagement.

Contact: contact@spiderbrain.ai | **spiderbrain.ai**

References and Data Sources

- Validation data: Spiderbrain engine v5.1 / decision layer v5.0.7, validated 2026-05-30. v5.5 re-validation, 2026-06-18.
- Clinical trial datasets: ClinicalTrials.gov public database, six therapeutic areas, 1,000 records per dataset.
- Product specifications: Spiderbrain Product Brief v0.5.5, 2026-06-18.
- Regulatory frameworks: ICH E6(R3) GCP Guideline; FDA 21 CFR Part 11; EU GMP Annex 11; ISPE GAMP 5 (2nd edition).

This working paper is produced by the Spiderbrain team. Core validation data (§4) reflects engine v5.1 / decision layer v5.0.7 (2026-05-30). The v5.5 results in §11 reflect the unified scorer 0.5.5 (2026-06-18). Spiderbrain · spiderbrain.ai · contact@spiderbrain.ai