RJL SOLUTIONS

LESSONS LEARNED / PRACTITIONER NOTES

25 Years of Lessons Learned as a Practitioner

40 lessons in 8 themes. Practical actions, evidence checks and questions for anyone getting data and AI into production.

OWNERSHIP · MEANING · PEOPLE · DELIVERY
GROUNDING · EVALUATION

Richard White
RJL Solutions · rjlsol.com

EXPANDED EDITION · 18 OCTOBER 2026
OCTOBER 2026

Twenty-five years. 40 lessons.

I have spent twenty-five years inside the problem: building data, technology and AI capabilities at insurers, banks, the Federal Reserve Bank of New York and a global professional services firm. The platforms changed every few years. The reasons programs succeeded or stalled hardly changed at all.

These are the lessons that held up, and the first matters most. AI pays off when you redesign the work. Bolting it on doesn't. Each lesson comes with a practical action and the evidence I would ask a team to show. The principles apply to a twenty-person portfolio company as much as to a global bank. Controls, staffing and evidence should scale with the consequences, and with the capacity to run them.

Fred Brooks gave this a useful frame in No Silver Bullet. He separated the mechanics of building software from the difficulty of deciding what it must do. I apply that distinction to AI. Faster implementation still needs agreed business meaning, permitted actions and evidence of correct behavior. AI can help explore those decisions. Accountable owners have to validate them.

This expanded edition compares the practitioner lessons with 100 external AI implementation lessons. The actions and evidence checks turn the advice into questions a delivery team can answer. External research informs the guidance; it does not describe my client engagements or establish a universal success rate.

8 themes

01-05	**Ownership and value.** Decide who owns the result before deciding what to build.
06-10	**Data and meaning.** Moving data is not the same as making it usable.
11-15	**Architecture and platform.** Give every layer one job, and build on what already works.
16-20	**Governance, security and evidence.** Governance belongs inside delivery, not beside it.
21-25	**Delivery and adoption.** A capability is finished when people use it and your team can run it.
26-30	**People and the work.** People adopt what makes their own job better, and quietly drop what does not.
31-35	**AI design and grounding.** Good answers need the right evidence, context, tools and rules.
36-40	**Evaluation and learning.** Decide what good means, test it repeatedly, and learn from the first failing step.

Ownership and value.

Decide who owns the result before deciding what to build.

01 Redesign the work. Do not bolt AI onto it.

Add AI to a job that stays the same and you buy a small, incremental gain. The real results come from rethinking how the work gets done: which steps disappear, where people keep the judgment and who owns the outcome, with human review matched to the cost of an error. That usually means taking down barriers and habits the organization has always lived with. It's uncomfortable. It's also where the value is.

Put it into practice. Map the current workflow, remove unnecessary steps, and state which judgments stay with people.

Evidence to ask for. A before-and-after process map with an outcome owner and measures for quality, elapsed time and review effort.

Companion paper: Adoption by Design

Implementation evidence: McKinsey 2025

02 Name the owner before the technology.

A named owner without authority or available time is a routing label. The person accountable for a result needs the mandate to change the process and the cooperation of the teams that control its dependencies. Resolve that gap first; it decides more than the platform does.

Put it into practice. Give the owner decision rights, dedicated time and an escalation route across dependent teams.

Evidence to ask for. A named owner who can change the process and resolve a blocked dependency.

Companion paper: The Data and AI Operating Model

Implementation evidence: RAND

03 Measure the baseline before you build.

Set targets after you measure the starting point. Never before. Without a baseline you can't show success, and you can't stop a failing capability early on evidence. Where you can, compare against a group still working the old way, so the result reflects the change and not the season or the extra attention. Reuse counts and cost curves are design arguments. They don't prove savings.

Put it into practice. Record representative workload, completion time, defects, review effort and cost before setting improvement targets.

Evidence to ask for. A comparable baseline and, where feasible, a control group or phased comparison.

Companion paper: Adoption by Design

Implementation evidence: Booking.com

Ownership and value (continued).

Decide who owns the result before deciding what to build.

04 End every phase with a decision.

Proceed, reshape or stop. Making the three outcomes explicit lets a pilot produce a useful result even when the answer is not to scale it. A stopped pilot that taught you something is cheaper than one that drifts for months.

Put it into practice. Write the proceed, reshape and stop criteria before the phase starts; identify who makes the decision.

Evidence to ask for. A dated decision record tied to results, unresolved risks and the next funding commitment.

Companion paper: Why Enterprise AI Stalls After the Demo

Implementation evidence: NIST AI RMF

05 Business owns the risk. Technology owns the product. Both sign.

Make a technology team accountable for a business risk and you've built in failure. The people who can change the process, the policy or the customer promise aren't the ones answering for it. Split accountability by what each side controls. The business owner answers for the outcome, the meaning of the data and how much risk is acceptable. Technology often owns the data product itself: its pipelines, controls, quality and day-to-day running. A data risk report carries both names, each accountable for their part, with one escalation path when they disagree. Shared accountability works when every piece has exactly one name. It fails when everyone shares everything.

Put it into practice. Name one accountable person for each business, data and technical decision; record how disagreements are settled.

Evidence to ask for. A signed responsibility map covering outcome, meaning, risk acceptance and operation.

Companion paper: The Data and AI Operating Model

Implementation evidence: Uber ML

ASK YOUR TEAM Which steps will disappear, who can change the process, and what evidence would make us stop?

Data and meaning.

Moving data is not the same as making it usable.

06 A faster platform does not fix meaning.

I saw it again and again. A company moved to the cloud, stood up a modern platform and expected the value to follow. Usually it got a faster, more expensive version of the reporting problems it already had. Moving data was never the same as making it usable, trustworthy and governed. AI-generated code faces the same test. Who counts as a customer? Which date applies? What does an exception permit? Those still need decisions. Remove unnecessary rules where you can, and make the remaining ones explicit and tested.

Put it into practice. Identify disputed definitions and exception rules before migration or code generation; assign an owner to resolve each one.

Evidence to ask for. Approved business cases with normal, boundary and exception behavior, reconciled across sources and implementation versions.

Companion paper: About Richard White

Implementation evidence: Data Cascades · Brooks: No Silver Bullet

07 Meaning needs a business authority.

Engineering can encode a definition. It cannot supply the business mandate to decide it. Give each critical definition an owner who can settle a dispute, and keep the version and effective date when a definition changes an answer.

Put it into practice. Assign a business authority to each critical definition, including permitted differences between domains.

Evidence to ask for. Versioned definitions, effective dates and resolved disputes linked to consumer-facing measures.

Companion paper: The Four-Part System

Implementation evidence: Data Cascades

08 An identifier connects records; it does not settle what they mean.

A shared customer key is valuable. It does not establish that every domain defines customer, exposure or active status the same way. Govern the valid differences instead of pretending they do not exist.

Put it into practice. Test entity matching separately from the business definitions attached to the entity.

Evidence to ask for. Examples showing valid domain differences, false matches and how conflicting records are reconciled.

Companion paper: Data Products in Practice

Implementation evidence: Google ML rules

Data and meaning (continued).

Moving data is not the same as making it usable.

09 A product is a promise to its consumers.

Publishing a table establishes nothing about meaning, supported use or reliability. A data product names its owner, its contract, its freshness and what happens when it fails. The test is simple. A consumer can use it, the team can run it and an executive can explain its value and exposure. For the documents an assistant answers from, freshness is part of the promise; an out-of-date procedure can produce a confident, wrong answer. Record approved uses, provenance, content rights and retention, and pull superseded material out of indexes and caches.

Put it into practice. Set freshness, quality, approved-use, provenance and retention commitments for tables and documents alike.

Evidence to ask for. A source register, consumer contract and an exercised procedure for stale or withdrawn content.

Companion paper: Data Products in Practice

Implementation evidence: NIST GenAI

10 A badge is not evidence.

A contributor must not be able to certify a product by editing the word certified. Certification is a decision about a specific version and scope, backed by evidence that does not move. An unchanged badge never overrides contrary operating evidence.

Put it into practice. Tie certification to a specific version, approved uses and dated evidence; revoke or limit it when conditions change.

Evidence to ask for. A review record whose scope and expiry agree with current operating evidence.

Companion paper: Data Product Certification Standard

Implementation evidence: NIST AI RMF

ASK YOUR TEAM Which definition or source could change this answer, and who can settle the dispute?

04 / LESSONS 11-13

Architecture and platform.

Give every layer one job, and build on what already works.

11 Every layer has one job.

A catalog is not an enforcement point, a semantic model is not an access policy, and an API gateway is not a semantic layer. When a component quietly takes on a second job, nobody can say where the control actually lives.

Put it into practice. Assign retrieval, meaning, authorization, serving and monitoring to explicit components and contracts.

Evidence to ask for. One request traced across the actual enforcement points, including denied access.

Companion paper: The Reference Implementation

Implementation evidence: Dropbox Dash

12 Meaning is context, not authority.

Semantic and ontology layers help the system choose the right answer. They never decide what a caller may see. Access is decided at the data, every time, and architecture on a slide is not authorization.

Put it into practice. Enforce current user permissions at retrieval and action boundaries, including caches and derived content.

Evidence to ask for. Denied requests remain denied after access changes, across source, index and application paths.

Companion paper: Context Before Query

Implementation evidence: OWASP agency

13 Start with the capabilities you already run.

A new architecture shouldn’t assume a wholesale replacement of the current stack. Choose components by their responsibilities and exit conditions. Reuse services that work, and keep the contracts stable when a platform or model changes. Match the tool to the task. A fixed workflow or a simpler model is often enough, and easier to test, run and explain than an agent. Compare build and buy on the full lifecycle. The license is only part of it.

Put it into practice. Compare the existing process, simple rules, a fixed workflow and an agent against the same task.

Evidence to ask for. A decision record covering feasibility, quality, integration, lifecycle cost and an exit path.

Companion paper: The Federated Data Platform

Implementation evidence: Agent patterns

Architecture and platform (continued).

Give every layer one job, and build on what already works.

14 Count all of the cost.

Include source load, transfer, storage, serving compute, integration, support, synchronization and the people time spent reviewing and correcting output. Zero-copy is not zero-cost. A benefit claim needs an observation window, a baseline and the full operating cost, or it is a hope.

Put it into practice. Divide the full operating cost by accepted completed tasks, including failed attempts, human review and support.

Evidence to ask for. Cost per accepted outcome across representative workload and volume scenarios.

Companion paper: The Federated Data Platform

Implementation evidence: DORA 2026

15 A platform earns adoption by removing repeated work.

The promise is that a team can ship a supported product without rebuilding identity, deployment, monitoring and baseline policy from scratch. Build it with the first teams that need it, and turn a component into a shared service once a second team asks for it. A platform that only adds approval steps will be routed around.

Put it into practice. Prove a component with a real delivery team, then generalize it when another team has the same need.

Evidence to ask for. Reduced repeated setup and delivery work measured across actual consuming teams.

Companion paper: The Data and AI Operating Model

Implementation evidence: Uber ML

ASK YOUR TEAM Which repeated work does the platform remove, and what will one accepted outcome cost?

Governance, security and evidence.

Governance belongs inside delivery, not beside it.

16 Put governance inside delivery.

Programs failed when governance ran beside delivery as a separate review at the end. Build the controls, the tests and the evidence into the same pipeline that ships the work. Then the release record becomes a by-product instead of a project. Bring risk and control partners in at design, not at sign-off. And keep an inventory of every AI system in use, including the ones that arrive inside purchased software.

Put it into practice. Bring control partners into design and record every AI system, including embedded vendor capabilities.

Evidence to ask for. An inventory with owner, purpose, dependencies, risk tier, review date and release decision.

Companion paper: About Richard White

Implementation evidence: NIST AI RMF

17 Test the no, not just the yes.

Prove that bypass fails: missing context, a revoked right, out-of-scope data and a prohibited query should all be denied, and the denial recorded. Exercise source drift, stale output and a failed policy service before a reviewer does.

Put it into practice. Exercise revoked rights, stale knowledge, missing context and unavailable policy services before release.

Evidence to ask for. Recorded denials and safe failures with clear ownership of remediation.

Companion paper: The Federated Data Platform

Implementation evidence: OWASP injection

18 Give each agent its own identity and supervision.

Policies written for people do not automatically constrain an agent or its tools. Give each agent its own identity, defined permissions, an accountable owner and a monitored record of what it does. Poor classification and excessive permissions can turn a data error into a harmful action. Set explicit limits on steps, time and spend, and define when it must stop and ask.

Put it into practice. Give agents scoped identities, tool permissions, spend and step limits, with checkpoints for consequential actions.

Evidence to ask for. A stopped or escalated action when authority, budget or confidence is insufficient.

Companion paper: Security for AI agents

Implementation evidence: OWASP agency

Governance, security and evidence (continued).

Governance belongs inside delivery, not beside it.

19 A document must never grant a privilege.

Treat retrieved material as untrusted input. Separate trusted configuration from content, restrict tool authority and test prompt injection. Knowledge is not an instruction channel. Check what the model produces before another system acts on it.

Put it into practice. Treat retrieved text as data; validate schemas, parameters and policy before downstream execution.

Evidence to ask for. Injection tests and malformed outputs cannot gain authority or trigger prohibited actions.

Companion paper: Context Before Query

Implementation evidence: OWASP injection

20 If you cannot reconstruct one request, you cannot defend it.

Ask the team to trace one real request from the user's purpose through its sources and controls to the answer. Wherever the explanation becomes an assumption, the release still has work to do. Preserve the inputs, versions and observed outputs needed to reconstruct what happened, and confirm actions in the system of record, not in the model's account of them. Build that evidence once and map it to every regime that applies.

Put it into practice. Preserve the inputs, versions, source references and observed results needed to reconstruct a request.

Evidence to ask for. An auditor can inspect the request and verify the resulting action in the system of record.

Companion paper: AI Governance Evidence

Implementation evidence: Agent evaluations

ASK YOUR TEAM Where is authority enforced, and can we reconstruct both an allowed and a denied request?

Delivery and adoption.

A capability is finished when people use it and your team can run it.

21 A pilot proves the model can answer.

Production needs evidence that it meets agreed quality, control, usability and cost criteria, and a named owner for what happens next. A demonstration leaves ownership, data, access and operating evidence unresolved unless someone tests those questions explicitly. Model quality still matters. Evaluate retrieval, generation, tool use and outcomes separately, on your own cases instead of a vendor's benchmark, and check any AI grader against human reviewers. NIST's Generative AI Profile (AI 600-1) is a reference for what to measure. Then ask who will do their job differently on the first Monday after release. Have they agreed to?

Put it into practice. Test the local workflow with users, real constraints and agreed quality and risk criteria before expansion.

Evidence to ask for. End-to-end results cover retrieval, generation, tool execution, review effort and business outcomes.

Companion paper: Why Enterprise AI Stalls After the Demo

Implementation evidence: Clinic deployment study

22 Fund the running, not just the building.

Reuse depends on a product staying trustworthy after its original project ends. Budget for maintenance, evaluation, consumer support and exceptions alongside the build. Don't protect an unused product just because its project closed on time. Quality drifts quietly after launch, so keep evaluating in production, and record the corrections people make, not just their satisfaction scores.

Put it into practice. Assign ongoing quality and freshness checks, incident ownership and a process to turn corrections into regression cases.

Evidence to ask for. A production review detects degradation and produces a tracked change to the system or its scope.

Companion paper: The Connected Enterprise Data & AI Blueprint

Implementation evidence: AI postmortem

23 No average offsets a critical failure.

Do not turn readiness into a percentage. Nine strong results do not outweigh one failure in access, isolation, unauthorized action or required evidence. Restrict the scope, fix the cause and test again. Report results by user group as well, because errors can concentrate where an average cannot see them.

Put it into practice. Separate critical control gates from average scores and examine consequential errors by user group and context.

Evidence to ask for. Critical failures block release; cohort results and error consequences are visible to the decision-maker.

Companion paper: The 90-Day First Governed Capability

Implementation evidence: FTC Rite Aid case

Delivery and adoption (continued).

A capability is finished when people use it and your team can run it.

24 Start small, read-only and bounded.

Keep the first release read-only unless an action is essential. Add authority when ownership, approval, failure handling and recovery are ready to carry the consequences. Release in stages, and design the fallback before you need it. Elapsed time is not evidence of readiness.

Put it into practice. Roll out in bounded stages and prove the manual fallback, pause and rollback paths.

Evidence to ask for. An exercised recovery path and a documented decision before widening authority or exposure.

Companion paper: The Connected Enterprise Data & AI Blueprint

Implementation evidence: NIST AI RMF

25 Transfer is proven, not declared.

Ask your team to restrict one capability, roll back one release and revoke one user, using only the runbooks. Whatever needs a builder to step in is not yet transferable. Plan the incident response and the retirement while the builders are still there. The goal is a capability the operating team owns and can support.

Put it into practice. Have the operating team run revocation, rollback and incident recovery without a builder taking over.

Evidence to ask for. Completed operational drills, named responders and a retirement plan for data, access and dependencies.

Companion paper: The Reference Implementation

Implementation evidence: NIST AI RMF

ASK YOUR TEAM Can the operating team detect a bad result, restrict the capability and recover without the builders?

07 / LESSONS 26-28

People and the work.

People adopt what makes their own job better, and quietly drop what does not.

26 Count completed work, not how fast it feels.

The tool can make work feel faster without cutting the time to an accepted result. Measure finished work that passes review, with review and rework time included. Keep personal productivity apart from enterprise value. Saved minutes become value only when they change throughput, cost, quality or risk somewhere the business can see.

Put it into practice. Measure accepted work through review, correction and downstream handoff; identify where released capacity will be used.

Evidence to ask for. The business can distinguish time saved, capacity released and value actually realized.

Companion paper: Data Is Not Just Code

Implementation evidence: METR 2025

27 Managers make or break adoption.

Executives sponsor the change. Frontline managers decide whether it happens on Monday morning. Involve them in the design, give them the measures, and make running the new process part of how they lead the team. A manager who was never asked can quietly protect the old way.

Put it into practice. Give frontline managers the authority, measures and time to coach the redesigned process.

Evidence to ask for. Managers use the workflow and review adoption obstacles with their teams.

Companion paper: Adoption by Design

Implementation evidence: BCG workforce

28 Train for the job, not the tool.

People need to know when to trust the output, when to challenge it and when to escalate. Teach that with their own cases and in their own role, and protect time to practice. Training squeezed into a full workload produces people who tried it once.

Put it into practice. Practice role-specific tasks, common failures and escalation using representative cases.

Evidence to ask for. Users and reviewers can recognize an unsupported answer and follow the right escalation path.

Companion paper: Adoption by Design

Implementation evidence: McKinsey adoption practices

People and the work (continued).

People adopt what makes their own job better, and quietly drop what does not.

29 Say plainly what changes, and reward it.

Tell people early which tasks change, which stop and what their role becomes. Promise only what you can keep. Then check that measures and incentives reward the new process. People follow what they're measured on. The launch email doesn't move them.

Put it into practice. Explain task and role changes early, then align incentives, review responsibilities and skill development.

Evidence to ask for. Performance measures support the new workflow and employees understand their responsibilities.

Companion paper: Adoption by Design

Implementation evidence: BCG workforce

30 Watch the real work before you scale.

Sit with the people doing the work, not just the dashboard. Look for workarounds, personal tools, growing review queues and strain on the reviewers. A capability that needs more supervision than it saves will be abandoned, whether or not anyone announces it.

Put it into practice. Observe the work where it happens, including reviewers, handoffs and people who stop using the tool.

Evidence to ask for. Workarounds, review queues, accessibility barriers and concentrated workload are acted on.

Companion paper: Adoption by Design

Implementation evidence: Clinic deployment study

ASK YOUR TEAM Whose job gets easier, whose review queue gets longer, and where will the saved time become value?

AI design and grounding.

Good answers need the right evidence, context, tools and rules.

31 The right answer starts with the right evidence.

If retrieval picks the wrong entity, period or policy, a stronger model still starts from the wrong facts. Test the evidence before rewriting the prompt. Keep headings, dates and source identity when splitting documents, and compare lexical search, semantic search and reranking on your own questions. A passage that sounds relevant isn't enough. It has to support this answer, for this caller.

Put it into practice. Use known-answer questions to test evidence coverage, source relevance, citation support and permission filtering.

Evidence to ask for. The required passages are retrieved for ordinary, ambiguous and no-answer cases, with measured retrieval errors.

Implementation evidence: Retrieval experiments · Dropbox Dash

32 Context is a budget. Spend it on what matters.

More context isn't automatically better context. Give the model the task, the governing instructions, the evidence it needs and a few representative examples. Keep irrelevant documents and redundant tool output out of the working window. When the instruction set keeps growing, find the repeated failure. Adding another exception won't fix it. Measure whether the extra context improves the task before paying for it on every call.

Put it into practice. Compare compact, sufficient context with larger inputs on the same tasks and inspect truncation and conflicting instructions.

Evidence to ask for. Quality, cost and latency are measured together; necessary evidence survives the context limit.

Implementation evidence: Context engineering

33 Build tools the agent can use correctly.

An API built for an engineer is not automatically a good tool for an agent. Names, parameters, limits and errors have to make the intended operation clear. Return the records needed for the decision, not an entire database. Overlapping tools create another choice the model can get wrong. Test whether the agent selects the right tool, supplies valid arguments and handles failure before adding more tools.

Put it into practice. Provide clear schemas, examples, bounded results and actionable errors; test invalid arguments and tool outages.

Evidence to ask for. Traces show correct selection, parameters and recovery without unnecessary calls or excessive returned data.

Implementation evidence: Agent tools

AI design and grounding (continued).

Good answers need the right evidence, context, tools and rules.

34 A fixed rule should remain a fixed rule.

Let AI help interpret a situation, draft an explanation or suggest a next step. Keep mandatory policy, calculation and authorization decisions in explicit controls that can be tested. If a condition is non-negotiable, asking the model to remember it is weaker than enforcing it. Record the rule version and make a conflict visible; do not let a persuasive explanation override the business decision.

Put it into practice. Separate generated judgments from mandatory rules, validate rule applicability and route unresolved conflicts to the accountable person.

Evidence to ask for. A fluent answer cannot override a prohibited action or an applicable mandatory rule.

Implementation evidence: Uber redlining · OWASP agency

35 Leave a usable memory, not a growing transcript.

Long-running work needs a record of the objective, decisions, completed steps, unresolved issues and sources. Preserve that state outside the conversation and load only what the next step needs. A summary that drops a restriction can be more dangerous than a context limit. Test the handoff after an interruption, and treat stored memory as governed content that can become stale or contain an injected instruction.

Put it into practice. Checkpoint the working state with source references, restrictions and versioned decisions; test restart and handoff.

Evidence to ask for. A resumed task keeps the original constraints and continues from verified progress without repeating consequential actions.

Implementation evidence: Long-running agents · OWASP injection

ASK YOUR TEAM Did the system find the right evidence and use its context, tools, rules and memory correctly?

Evaluation and learning.

Decide what good means, test it repeatedly, and learn from the first failing step.

36 Agree what good means before automating the score.

A generic helpfulness score won't tell you whether the answer is complete, supported or usable. Start with real cases, and have the people who own the work label them. Include ambiguity, and cases where the right response is to decline or escalate. Settle expert disagreements before scaling an evaluator. A specification can be shorter than the code. It still has to preserve the required behavior. Turn approved decisions into acceptance cases, and update the specification and tests when a defect exposes a missing rule.

Put it into practice. Define separate rubrics for correctness, support, completeness, action safety and usability, with explicit failure examples.

Evidence to ask for. A reviewed reference set and resolved labeling disagreements support the release criteria.

Implementation evidence: Airbnb evaluations

37 An AI grader needs its own evaluation.

Using AI to judge AI can reduce review effort. It can also automate the same mistake at scale. Compare the grader with human decisions on cases that were not used to tune it, inspect disagreements and repeat the check after model or prompt changes. Keep expert review for consequential decisions. Agreement on easy answers is not evidence that the grader can detect the failure that matters.

Put it into practice. Calibrate each evaluator against held-out expert judgments, including bad examples and high-consequence errors.

Evidence to ask for. Documented disagreement analysis shows which errors the grader catches, misses or wrongly flags.

Implementation evidence: Dropbox labeling

38 One successful run is not reliability.

The same request can succeed once and fail on the next attempt. Repeat consequential tasks, and vary the inputs, user groups and operating conditions. Test whether the system consistently reaches an acceptable outcome. Finding one successful example proves little. Report uncertainty and the consequences of error. A narrow release can be the right call when evidence is limited. A high average is no permission to ignore a critical failure.

Put it into practice. Run repeated trials and test realistic noise, tool failures and cohort differences against explicit release gates.

Evidence to ask for. The decision includes success variation, sample limitations and any conditions where the system must escalate.

Implementation evidence: Agent evaluations

Evaluation and learning (continued).

Decide what good means, test it repeatedly, and learn from the first failing step.

39 Find the failing step before changing the model.

An answer can degrade because the source changed, retrieval missed the evidence, a tool failed or the serving path changed. Inspect the trace and compare the first failing step with a known-good case. Change one factor at a time where practical. Check the outcome in the underlying system as well as the wording of the response. A new model will not reliably repair a broken data or execution path.

Put it into practice. Use component checks and end-to-end traces to isolate source, retrieval, generation, execution and infrastructure regressions.

Evidence to ask for. A reproducible failing case becomes a regression test and verifies the fix in the production path.

Implementation evidence: AI postmortem

40 Prove it on what comes next.

A model can appear accurate because its test data contains information that would not exist when the decision is made. For a forecast, test on later periods and preserve the inputs available at the time. For any AI capability, retest when the model, source, workflow or user population changes. Evidence is dated. Revisit the assumptions that justified the release instead of treating the original approval as permanent.

Put it into practice. Use time-appropriate holdouts, check for information leakage and schedule reassessment after material changes.

Evidence to ask for. Later-period results and a change-triggered review confirm that quality, cost and risk remain acceptable.

Implementation evidence: Google ML rules · METR follow-up

ASK YOUR TEAM Would this release still pass on unseen cases, repeated attempts and the next period of real work?

10 / WHAT THEY SHARE

Decisions, not technology.

None of these lessons depends on a particular vendor, model or regulator. They depend on decisions: who owns the result, what counts as evidence and when to stop. Regulated firms write those decisions down because a supervisor will ask. Fast-moving firms should write them down too, because an investor, a customer or a future acquirer will ask the same questions.

If one of these lessons describes the initiative on your desk, bring it to a working session. The first sixty minutes are free.

START HERE Pick one lesson, record the current condition, and assign an owner and a review date. Bring the evidence back before widening the scope.

SOURCES / RESEARCH AND IMPLEMENTATION EVIDENCE

Read the evidence in context.

Survey associations, interviews, experiments and first-party implementation accounts answer different questions. Results depend on the task, people, tools and conditions studied. Standards and security guidance inform controls; they do not prove financial returns. Implementation sources reviewed 17 October 2026; Brooks added 18 October 2026.

No Silver Bullet: Essence and Accidents of Software Engineering
1986-09; accessed 2026-10-18. Foundational essay; sections 1-2 and 5.2. AI application is the author's interpretation, not an AI productivity study.

Effective harnesses for long-running agents
2025-11-26. Vendor engineering experiments and guidance.

The state of AI in 2025: Agents, innovation, and transformation
2025-11-05. Organizational survey.

The Root Causes of Failure for Artificial Intelligence Projects and How They Can Succeed
2024. Interview research.

150 successful Machine Learning models: 6 lessons learned at Booking.com
2019. Implementation research; successful applications validated through randomized experiments.

AI Risk Management Framework Core
2023 framework; living resource. Consensus risk-management guidance.

Scaling Machine Learning at Uber with Michelangelo
2018-11-02. First-party implementation account.

Everyone wants to do the model work, not the data work: Data Cascades in High-Stakes AI
2021. Interview research; 53 practitioners.

Rules of Machine Learning
Living page. Engineering guidance from deployment experience.

Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile
2024-07. Risk-management guidance; not an outcome study.

Sources (continued).

Survey associations, interviews, experiments and first-party implementation accounts answer different questions. Results depend on the task, people, tools and conditions studied. Standards and security guidance inform controls; they do not prove financial returns. Implementation sources reviewed 17 October 2026; Brooks added 18 October 2026.

Building Dash: How RAG and AI agents help us meet the needs of businesses
2025-04-24. First-party implementation account.

LLM06:2025 Excessive Agency
2025 edition. Security guidance.

Building effective agents
2024-12-19; subsequently updated. Vendor engineering guidance from customer and internal experience.

Balancing AI tensions: Moving from AI adoption to effective SDLC use
2026-03-10. Survey and qualitative research interpretation.

LLM01:2025 Prompt Injection
2025 edition. Security guidance.

Demystifying evals for AI agents
2026-01-09. Vendor engineering guidance and implementation examples.

A Human-Centered Evaluation of a Deep Learning System Deployed in Clinics for the Detection of Diabetic Retinopathy
2020. Observational implementation research; eleven clinics in Thailand.

A postmortem of three recent issues
2025-09-17. First-party production incident account.

Rite Aid Banned from Using AI Facial Recognition After FTC Says Retailer Deployed Technology without Reasonable Safeguards
2023-12. Regulator complaint and proposed settlement account; allegations identified as such.

Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity
2025-07-10. Randomized controlled trial; 16 developers, 246 tasks.

Sources (continued).

Survey associations, interviews, experiments and first-party implementation accounts answer different questions. Results depend on the task, people, tools and conditions studied. Standards and security guidance inform controls; they do not prove financial returns. Implementation sources reviewed 17 October 2026; Brooks added 18 October 2026.

AI Transformation Is a Workforce Transformation
2026-02-04. Survey and practitioner guidance.

The state of AI: How organizations are rewiring to capture value
2025-03-12. Organizational survey.

Introducing Contextual Retrieval
2024-09-19. Vendor experiments; results depend on datasets and configurations.

Effective context engineering for AI agents
2025-09-29. Vendor engineering guidance.

Writing effective tools for agents-with agents
2025-09-11. Vendor engineering experiments and guidance.

Scaling AI in Legal: Building Uber's Redlining Agent
2026-10-08. First-party implementation account.

Eval-driven development: Lessons from evaluating GenAI at scale
Publication date not shown in retrieved page. First-party implementation guidance.

Using LLMs to amplify human labeling and improve Dash search relevance
2026-02-26. First-party implementation account.

We are Changing our Developer Productivity Experiment Design
2026-02-24. Study update and methodological limitations.

ABOUT

Richard White

I was global firm chief data officer at McKinsey & Company, owning data, AI and platforms end to end, and before that chief data officer at New York Life, where I built the chief data office and reported data risk to risk committees at multiple levels. Earlier I led data at the Federal Reserve Bank of New York and enterprise data governance at Citigroup. I founded RJL Solutions in 2024. My book, *The Day After the Demo*, is free at rjlsol.com/book.

RJL Solutions connects strategy, architecture, engineering, governance and operation through Frame, Prove, Build, and Run & Transfer. Each phase is separately scoped and priced in a written proposal before you commit. The first sixty-minute working session is free.

START **Why Enterprise AI Stalls After the Demo**
The short one to forward to a colleague.

ORGANIZE **The Data and AI Operating Model**
Ownership, teams, decision rights and funding.

PRACTICE **Adoption by Design**
The people and workflow changes that make a capability useful.

RJL SOLUTIONS

WEB rjlsol.com

EMAIL rwhite@rjlsol.com

Practitioner guidance, expanded with implementation research. External studies are identified in the source notes; no client engagement or client result is described.