RJL SOLUTIONS

EXECUTIVE PERSPECTIVE / ENTERPRISE AI

Why Enterprise AI Stalls After the Demo

What has to change before people can rely on the capability.

INFORMATION · AUTHORITY · OPERATING OWNERSHIP

Richard White
RJL Solutions · rjlsol.com

OCTOBER 2026

01 / THE GAP

The demo proves a narrower claim than the one you are about to fund.

Enterprise AI stalls when the conditions that made a demonstration work have no dependable equivalent in daily operations. The model can answer a question. The enterprise still hasn't settled which information it may use, what that information means, who owns the resulting work or how anyone will support the service.

This paper is for business, data and technology leaders deciding what happens after a promising pilot. My recommendation: fund one complete governed workflow. That means a bounded capability with real users, approved information, enforced authority, measurable outcomes and continuing ownership. Let its evidence decide whether you expand, reshape or stop.

What changes when the audience expands

A demonstration can use selected documents, familiar questions and a broadly privileged service account. The people presenting it may resolve ambiguity without noticing they've supplied knowledge the system doesn't have. Production exposes that ambiguity to users who expect consistent behavior. They can't inspect every hidden assumption.

IN THE DEMONSTRATION	WHAT EVERYDAY OPERATION REQUIRES
Someone selects the right documents	A maintained route to applicable and authorized sources
The presenter knows what a term means	Approved definitions, and a way to select the right context
A service account can reach everything	Authority bounded by the caller and the allowed operation
A plausible answer ends the exercise	A useful result, a review path, and an observable outcome
The builders handle problems informally	Named support, recovery, change and maintenance owners

Everyone queues in the same lane.

Most organizations queue in the AI solutions lane: tools, models and pilots. Meanwhile the lanes that lead to production usually sit open. Workflow redesign settles which steps go and who owns the result. Data fundamentals give the work owned definitions, certified products and access that holds. Skip them, and a pilot reaches an impressive demonstration and stops there.

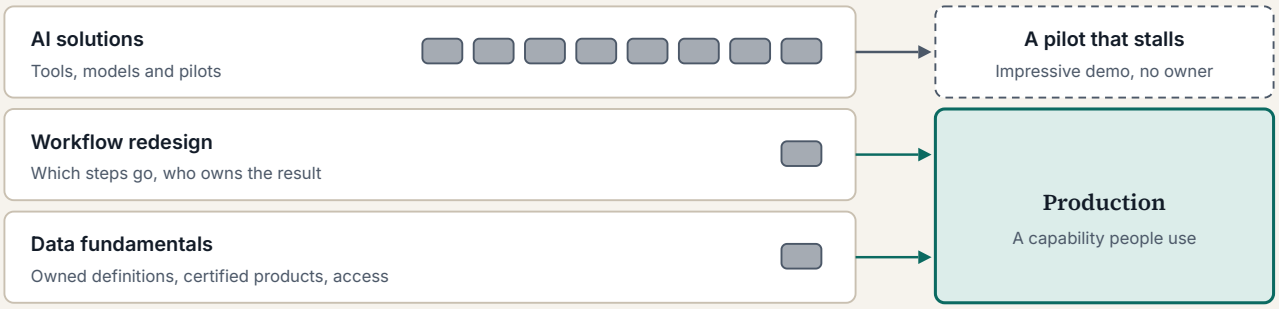


Exhibit: Everyone queues for AI solutions; the open lanes lead to production.

Follow one request beyond the screen

Picture an assistant helping a claims examiner assemble case context. The demo returns a status and a procedure. In daily use, though, the examiner needs the status of the right claim, a procedure that applies to that policy and date, and only information she's entitled to see. A confident answer can hide failure on any of those conditions.

This is an illustrative workflow. It isn't a client deployment or an achieved result, just a concrete way to examine the gap.

THE EXERCISE Ask the team to trace one request, from the user's purpose through its sources and controls to the answer and the business outcome. Wherever the explanation turns into an assumption, the release still has work to do.

02 / INFORMATION

Information needs agreed meaning and provenance.

An AI interface can make inconsistent information easier to consume. It can also hide the inconsistency behind a fluent explanation. A warehouse, catalog or search index helps organize information. The workflow still needs a business authority that decides what a result means and when a source applies.

Resolve the question before generating the answer

In the claims example, the person connected to a claim could be the insured person, the policyholder, a beneficiary, the employer or a representative. Link the wrong role to the case and you get a coherent, incorrect summary. An identifier and a relationship rule are business decisions. Someone accountable has to own them.

Metric questions carry similar dependencies. A count of open claims needs an agreed status definition, a reporting date, a record grain and a rule for reopened or transferred cases. Several definitions may be valid. The system has to pick the one that fits the user's purpose, and keep enough context to explain the choice.

For the first workflow, approve its essential definitions and test them against known examples. Put unresolved matches and conflicting definitions in front of reviewers. Giving the language model more context won't settle a disagreement about which definition the business accepts.

Maintain the knowledge the assistant consumes

A retrieved passage can be relevant and still be obsolete, restricted or wrong for the case. As the document gets parsed and indexed, keep its source, version, effective date and applicable product or jurisdiction. The answer should point to evidence the user may inspect.

The document owner also needs a working route for replacements, deletions and permission changes to reach derived indexes and caches. Suppose the source changes while the assistant keeps returning an old procedure. Retrieval quality measured during the pilot then tells you little about the current service.

Give dependencies a consumer contract

For each required data product or knowledge source, name the owner. Define the interface, meaning, quality, freshness, restrictions, and the behavior when the contract isn't met. Make the degraded state visible to the workflow. A status source too old for the task should trigger the agreed restriction or escalation.

Certify the dependencies this release needs, and keep that certification current through change. The practical question is simple. Can a consumer rely on the stated commitments today? A completed catalog entry or manifest helps only when the implementation, the tests and the support behavior honor it.

03 / AUTHORITY

Authority must survive the complete request path.

Production needs a defined relationship between the caller, the assistant and each permitted operation. Controls have to hold wherever information is read or an action happens, downstream services and caches included.

Bound the capability before exposing it

Identify both the caller and the agent acting for them. Define the approved user population, the information scope and the tool operations. A user allowed to read a case doesn't automatically authorize an agent to update it, issue a payment or contact a customer.

Use interfaces that expose the business operation the workflow needs. A constrained status lookup is easier to reason about, and to test, than an unrestricted query or update tool. When writes come into scope, deal explicitly with input validation, approval, duplicate protection and fixing incorrect actions.

Treat retrieved instructions as untrusted content

A document can contain text that tries to redirect the assistant. OWASP identifies this as an indirect prompt-injection path.[1] The retrieved content should supply evidence for the task. It should not grant authority, change tool permissions, or override operating policy.

Test hostile content. Enforce tool permissions independently of how the model interprets things. Controls must hold when a document, a prompt or a tool result asks the assistant to cross the boundary.

CONDITION TO TEST	EXPECTED BEHAVIOR, DEFINED BEFORE RELEASE
Access is revoked	Deny the affected request through sources and caches
A source or policy service fails	Restrict the affected path or escalate safely
Sources conflict, or identity is uncertain	Disclose the uncertainty and route it to review
A document requests an unauthorized action	Preserve the tool boundary and refuse the action

Retain evidence of what happened

Pick a real request and reconstruct it: the caller's authority, the source and component versions, the policy decisions, the tool calls, the answer and the observed outcome. Define retention and reviewer access without logging sensitive content you don't need.

Reconstruction supports incident investigation and the review of material changes. Put expected refusals and safe failures in the release evidence alongside the useful answers.

04 / OWNERSHIP

The capability needs an operating owner.

A pilot can deliver an interface while nobody has the authority to settle a definition dispute, change the business workflow or fund maintenance. Divided ownership leaves the complete capability with no accountable owner.

Establish decision rights for the first workflow

Name the actual decision holders. A product owner coordinates them, without inheriting the authority to approve access or accept business outcomes. Titles vary. The responsibilities have to be clear.

RESPONSIBILITY	THE DECISION THE NAMED OWNER MUST BE ABLE TO MAKE
Business process	Change the workflow and accept measured results
Product lifecycle	Set scope, prioritize changes, and retire the capability
Domain meaning and data	Approve definitions and maintain source commitments
Security and controls	Set permitted use, approve conditions, and stop an unsafe release
Service operation	Own support, recovery, operating limits and recurring cost

The first capability can set up these relationships without waiting for an enterprise reorganization. Give the delivery team access to the decision holders, an escalation route and an agreed decision time frame. Record disagreements as decisions to resolve. Don't bury them in technical backlog items.

Share infrastructure and retain domain authority

Shared teams should provide reusable identity, deployment, monitoring, evaluation and catalog services, wherever those services meet the workflow's requirements. Domain teams should maintain the business meaning and source commitments they're qualified to own.

This avoids forcing every business unit to build its own controls, while keeping business decisions close to the people accountable for the work. NIST's AI Risk Management Framework, a voluntary framework, explicitly addresses clear roles, executive responsibility and ongoing review.[2] The operating model makes those responsibilities concrete for this release.

Transfer responsibility through demonstrated operation

Bring internal engineers and service owners into the implementation, the tests and the support from the start. Before transfer, have the receiving team demonstrate four things: a routine change, an access revocation, an incident investigation and a rollback.

THE TEST OF READINESS	The receiving team runs the runbook without the builders. Fund source upkeep, evaluation and exception handling alongside the initial build, so the continuing obligation has an owner and a budget.

05 / VALUE

Value must survive everyday work.

A pilot can win enthusiastic feedback without improving the business process once review, corrections and exceptions are counted. The outcome that matters is completed useful work at acceptable cost and risk. Measure it in the workflow. Response quality and the number of people who tried the assistant won't tell you.

Establish the comparison before release

Record how the work gets done today, including information gathering, waiting, review, corrections and escalation. Choose an eligible population and a baseline period that represent the intended scope. When comparing results, account for material differences in case mix or workload.

For the claims assistant, measure the effort of assembling usable case context, including the examiner's verification and correction time. Track usage among eligible cases, incomplete tasks, unsupported answers, unresolved identity matches and inappropriate disclosures. Define the target, and the outcomes you won't accept, before the team sees any results.

Time saved is available capacity. It becomes a cash saving only when staffing, overtime or purchased capacity changes. Keep that distinction visible in the investment case, and call any estimate what it is: an assumption to validate.

Count the cost of the complete task

Include model and compute use, retrieval, integration, human review, support and recurring information upkeep. Divide the relevant operating cost by completed useful tasks over the same period. Define what completion means, so retries and abandoned requests can't flatter the result.

An assistant may cut the time to draft an answer while adding time to verify it. A cheaper model may throw more exceptions. Compare the complete workflow and its service objectives before deciding whether a change improves the economics.

Count what ownership costs, then decide on evidence.

The business case usually approves what's visible at the start: build, licenses, vendor fees, integration and pilot compute. Ownership pays a longer list every year, in six groups of two recurring costs.

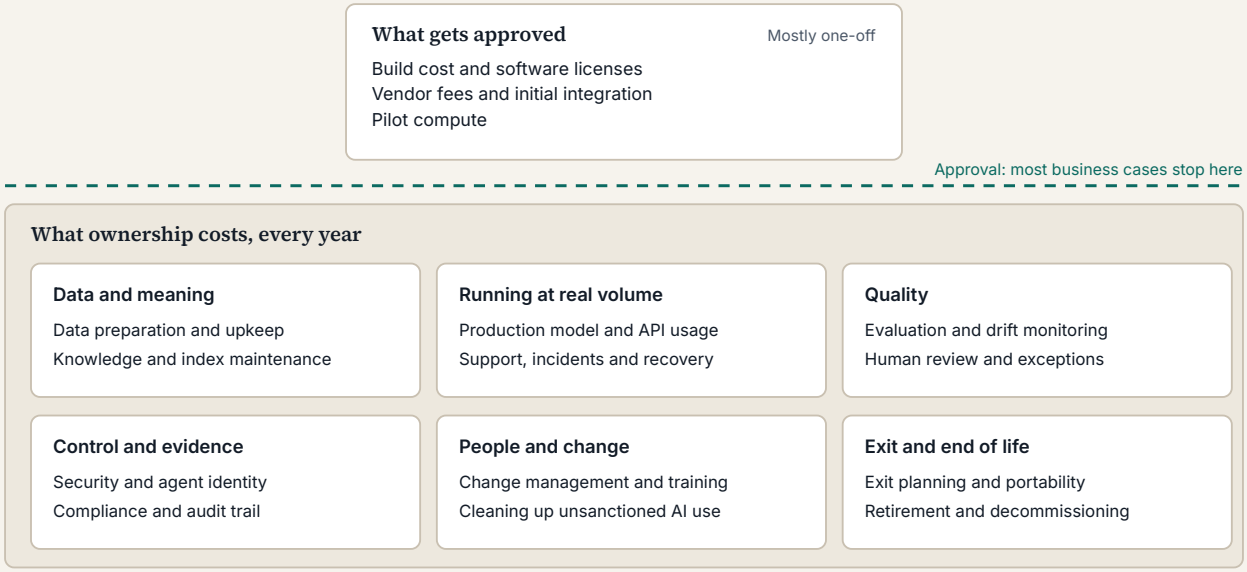


Exhibit: The business case approves the tip; ownership pays the rest, every year.

Make expansion depend on release evidence

Agree acceptance thresholds for useful results and for critical controls. The test set should include representative work, known answers, expected refusals, hostile content and degraded dependencies. Record component versions so you can identify the evaluated capability and change it deliberately.

Test the operating behavior as well as the answer. Establish who handles an incident, who can restrict a tool or a user population, and how the service recovers. A material change to a source, definition, model, prompt or tool should trigger an assessment of what needs re-evaluating.

THREE OUTCOMES

Expand when business results, control evidence and operating capacity support the next scope. Reshape when the workflow or the assumptions need to change. Stop when the value, or a workable path to it, no longer justifies the investment. Make those three outcomes explicit and a pilot can produce a useful result even when the answer is not to scale it.

06 / WHAT TO DO

Start with one complete governed workflow.

Choose a valuable task with a business owner, identifiable users, reachable dependencies and a testable boundary. Deliver its information, controls and operating responsibilities together. Then reuse the proven contracts and services in the next capability.

01 **Frame the deciding constraints.** Map the workflow and agree the outcome measure. Identify sources, unresolved definitions, permitted actions and decision holders. Pick a useful release that tests the deciding constraints. Before promising a release, resolve missing information or authority, or reshape the scope.

02 **Prove the complete path.** Use approved real data and representative users. Trace requests through sources, definitions, permissions and tools. Test failures and refusals early enough to change the design. Compare completed work with the baseline, review and exceptions included.

03 **Build and hand over the service.** Complete release evidence, support, monitoring, rollback and dependency commitments. Have internal owners demonstrate operation before you extend scope. A 90-day first-capability pattern can organize this work. The schedule you can actually hit depends on access, decisions and integration constraints.

04 **Make the next investment decision.** Review a recent request trace, the outcome baseline and the gaps that matter. Give each gap an owner and closure evidence. Decide to proceed, reshape or stop, with explicit conditions for expansion.

RJL works in four phases: Frame (two to three weeks), Prove (four to six), Build (eight to sixteen weeks per release), then Run & Transfer until internal ownership is demonstrated. Use the **Connected Enterprise Data & AI Blueprint** for the wider architecture, and the **10-Question AI Production-Readiness Diagnostic** to structure the evidence discussion. Both are at rjlsol.com, where the first sixty-minute working session is free.

Bring the workflow and its open decisions to a working session: **rjlsol.com/contact** or **rwhite@rjlsol.com**.

Further reading

[1] OWASP Cheat Sheet Series, LLM Prompt Injection Prevention Cheat Sheet, accessed October 2026. See remote and indirect injection, and controls across the application.

[2] NIST AI Risk Management Framework 1.0, January 2023 (under revision in 2026). See GOVERN 1.5, 2.1 and 2.3 for ongoing review, roles and executive responsibility.

The release recommendations in this paper are RJL Solutions' operating guidance. They don't establish certification, and they don't replace your own approval process.

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.

GUIDE **The Connected Enterprise Data & AI Blueprint**
The architecture and operating decisions behind this paper, in 24 pages.

WORKSHEET **The 10-Question AI Production-Readiness Diagnostic**
Four pages to complete with your business owner, technical lead and control owner.

RJL SOLUTIONS

WEB rjlsol.com

EMAIL rwhite@rjlsol.com

ENGAGEMENTS Scoped and priced in a written proposal; first session free

This paper describes no client engagement and no client result. The claims workflow is illustrative. Before production adoption, any implementation has to validate platform maturity, regional availability, licensing, regulatory obligations, performance and operational ownership.