ARIC INSIGHT · VOLUME I · ISSUE 12 · PRACTICE IV · OPEN RECORD

AI Adoption for *Non-Technical* Organizations.

A working position from the firm on putting generative AI (the engine) and agentic AI (the deployment pattern that wraps it) to work inside organizations whose bench is not, and is not going to become, a software-engineering bench. Five named operating risks. A thirty-day assessment carried by unit function. An optimization program with an embedded engineer for operations and maintenance. A planning calendar borrowed from the public sector that holds against the cost trap the cloud generation already paid to learn.

EDITORIAL POSITION NOTE · AI, AGENTIC & SELC

FOREWORD · THE FIRM'S POSITION

The principals the firm sits with — chairmen and chief executives outside the technology sector, Group chief information officers carrying the operating weight of holdings whose primary discipline is not software, and the owners of the small and mid-sized firms that make up the bulk of any national economy — are now being asked, in writing, what their organization's AI posture actually is. The firm's working position is that the honest answer is not a tool licence and is not a pilot; it is **a thirty-day assessment carried by unit function**, a defined optimization program with operations-and-maintenance built in from day one, an *embedded engineer* who is paid to keep the system honest after deployment, and a cost discipline borrowed from the public-sector planning calendar that the firm reads as the only available defence against the credit-burn pattern the cloud generation already paid to learn. Editorial, not legal, tax or investment advice.

ISSUE
Vol I · Issue 12

DATED
27 May *MMXXVI*

VOICE
Firm · Reading Desk

CONTENTS

Contents & Reading Plan.

A six-section position paper plus a thirty-day field-log appendix from a real production environment. The headline 38.6% rework-verb rate, the additive-vs-corrective split, and the proficient-operator discipline are carried in Section VI.

Sections

Reading plan for three readers

The **principal of a non-technical operating company** reads Sections I, II, and VI first; Sections III through V on the second pass before any vendor conversation. The **Group chief information officer of a diversified holding** reads Sections IV and V first; Section VI as the cost-discipline annex. The **owner of a small or mid-sized business** reads Sections II, III, and VI first; the rest as the program scales.

> *A position paper carried in firm voice, plus a thirty-day field log from the firm's own working deployment.* **The numbers in Section VI are from a real production environment, not a vendor demo.**

SECTION I · THE DISTINCTION IN PLAIN ENGLISH

Generative AI is the *engine*. Agentic AI is the deployment pattern that *wraps* it.

An honest plain-English read on the two categories of artificial intelligence the principal now has to govern, why the distinction is not a marketing line, and what the operating implications are for an organization whose bench is not a software-engineering bench.

I.1 · *Generative AI — the underlying model class, used at the prompt surface.*

Generative AI, in the firm's working definition, is the underlying class of machine-learning model that produces text, code, structured data, images, and analysis on demand in response to an instruction. At the prompt surface every executive the firm reads for has now sat in front of, the operator types a question; the model composes an answer; the operator reads it, edits it, and uses it. The unit of value is a single completion. The model itself is stateless turn-to-turn — what feels like memory is the product surface around it: the chat history is logged and cached against the operator's account, so a returning user resumes the same thread with the prior turns replayed back into the model on every new message. *That is continuity for the user; it is not memory inside the model.*

The honest operating risks at the prompt surface are therefore not 'the model forgets.' They are the failure modes that show up as the session gets longer and the replayed context window gets denser: **logic-tree drift**, where the model's reasoning branches away from the operator's original intent because an early misinterpretation got pinned into the cached history and re-served on every subsequent turn; **accumulated misinterpretation**, where a single wrong premise stops being challenged because it now lives inside the context the model treats as fact; **token burn on rework**, where the operator pays in compute, time, and direct credit for re-asking, re-prompting, and rewinding around the drift; and **retrieval calls landing in the wrong data index**, where the model is wired to a knowledge base and surfaces a fluent answer from a corpus that was never the right one for the question. The audit trail is the chat transcript and is useful for review; it is not exhibitable as a system of record.

The honest read on generative AI used at the prompt surface is that it is a productivity tool in the same register a word processor or a spreadsheet was a productivity tool when each first arrived. It accelerates the operator who already knows what answer is correct; it does not, on its own, change the organization's operating posture. The principal does not need to govern a word processor at board level, and does not need to govern generative AI at the prompt surface at board level either, beyond a written acceptable-use policy, a confidentiality discipline on what gets pasted into a prompt window, and an operator-side habit of *starting a fresh session when the working thread has drifted* rather than spending more tokens trying to argue the drift back out.

I.2 · *Agentic AI — the deployment pattern that wraps a generative model with tools, state, goals, and audit.*

Agentic AI is not a different model class. **It is the deployment pattern that takes the same underlying generative model and wraps it with four things the prompt surface does not have**: tools the model is allowed to call against the organization's systems — a database query, a calendar write, an email send, an invoice match, a registry lookup, a payment instruction; a written goal the system pursues across many steps; state held across the working session, so the entity identified on Tuesday is the same entity surfaced again on Friday with the new context attached; and a rendered audit trail, for every step taken, that the organization can stand behind.

The operating implication is that an agentic system is not a tool the operator drives; it is a working capacity that takes action on the organization's behalf, inside the organization's systems, against the organization's data, on the organization's authority. *That is a different governance posture and a different cost posture from the prompt surface, even though the underlying generative model may be identical.* The principal does need to govern the agentic deployment at board level. The Group chief information officer does need to architect it deliberately. The small-business owner does need to choose, with eyes open, which operating functions get an agentic capacity and which do not.

I.3 · *Why the distinction matters for a non-technical organization.*

The firm reads the principal's current question — *what is our AI posture* — as one that conflates the two categories. A licence for a generative AI tool used at the prompt surface is not an agentic deployment. A pilot of an agentic capacity is not a deployment. The operating bench of a non-technical organization does not, on its own, carry the discipline to keep the two categories cleanly separated, and the vendor decks the principal is reading do not help — the same brand name will sell the operator a chat box and an agent on the same invoice, and the principal will be asked to underwrite both as if they were the same thing. They are not.

> **Generative AI is the engine. Agentic AI is the deployment pattern that wraps that engine with tools, state, a goal, and an audit trail so that it takes action on the organization's behalf.** *The underlying model may be identical; the governance posture, the cost posture, and the failure modes are not.*

II. *The five operating risks, named.*

Five named failure modes the principal of a non-technical organization should be reading for, in writing, before any agentic capacity is allowed near the operating bench. The risks are not theoretical. Each has been observed inside operating deployments the firm has reviewed. Naming them on the open record is the discipline that lets the principal read for them on day one rather than discover them on day ninety.

- **Hallucinations.** The system produces a fluent-sounding answer with no source basis. A customer name that does not exist in the file. An invoice figure that does not reconcile to the ledger. A regulatory citation that does not appear in any code. *The discipline:* source-span attribution on every substantive output, and a written rule that any output without a citable source is not used.
- **Ghost edits.** The system makes a change to the organization's data, system, or document of record that no operator authorised and no audit trail surfaces. A field overwritten. A status flipped. A line item added. *The discipline:* every write the system performs is logged, attributable to the agent identity, and reviewable by an operator who can roll it back.
- **Silent failures.** The system fails to complete the goal but renders a finished-looking output anyway. The reconciliation that did not balance, presented as if it had. The customer-service ticket marked resolved when no resolution was sent. *The discipline:* functional verification — a defined post-condition check on every goal the agent claims to have completed, and an exception path that escalates when the post-condition is not met.
- **Intent gap.** The operator asked the system for one thing and the system pursued another. The operator asked for a draft; the system sent. The operator asked for a read; the system filed. *The discipline:* the human-in-the-loop boundary, written in the engagement, that draws an explicit line between what the agent may propose and what the agent may execute without a confirming human signal.
- **Rapid burn rate.** The system runs in a loop the operator did not intend, consumes compute credit at a rate the operator did not size for, and produces an invoice the principal did not authorise. *The discipline:* the credit ceiling, the per-session budget, the per-goal cap, and the monitoring that surfaces a burn-rate anomaly before the invoice does.

The risks compound on each other. A hallucination that triggers a ghost edit produces a silent failure the operator does not catch; the intent gap that drives the agent into the wrong workflow produces a rapid burn against a goal nobody wanted completed. The firm reads any deployment that does not have a written posture on all five risks, on the day the engagement letter is signed, as **a deployment carrying compounding exposure the principal has not yet been priced for**.

> *Hallucinations. Ghost edits. Silent failures. Intent gap. Rapid burn rate.* **Five named failure modes.** *The firm reads any agentic deployment without a written posture on all five — on the day the engagement letter is signed — as a deployment carrying compounding exposure the principal has not yet been priced for.*

III. *The thirty-day assessment, carried against the actual org chart.*

An assessment carried against the organization's actual org chart — function by function — not against a vendor's reference architecture. The vendor's reference architecture does not match the principal's operating bench; the principal's operating bench is what the program will be sized against.

The assessment is time-boxed at thirty calendar days from the day the engagement letter is signed, with a written deliverable returned to the principal and the Group chief information officer on day thirty. The deliverable is not a strategy document. It is a function-by-function read of the organization's current operating bench against three questions: which goals are presently completed by a human operator that could be completed, in whole or in part, by an agentic capacity; what the dependency map looks like; and what the five-risk register looks like, function by function, against the agentic capacities the firm is recommending.

The unit functions the assessment is carried against — the firm's standing register, calibrated for a non-technical operating company, adjusted on the first morning of the engagement to match the principal's actual chart:

- **Office of the principal** — calendar, correspondence, briefing materials, board-pack preparation, decision log. High agentic suitability for drafting and triage; load-bearing intent-gap and ghost-edit discipline required because the signing authority is the principal.
- **Finance and treasury** — invoice intake, expense reconciliation, payment-instruction preparation, cash-position reporting, audit-pack assembly. High agentic suitability for reconciliation and reporting; absolute discipline on the human-in-the-loop boundary at the payment-execution line.
- **Operations** — purchase orders, supplier onboarding, inventory and asset registers, scheduled-maintenance dispatch, exception reporting. Agentic suitability rises with the maturity of the underlying system of record.
- **Sales and account management** — pipeline hygiene, meeting preparation, follow-up drafting, contract intake, renewal-window monitoring. High agentic suitability for drafting and monitoring; intent-gap discipline at the contract-send line.
- **Human resources** — applicant intake, scheduling, onboarding-checklist execution, policy lookups, anniversary and review monitoring. Any deployment touching hiring or termination decisions requires counsel sign-off on the decision boundary.
- **Customer service** — ticket triage, first-response drafting, knowledge-base lookup, escalation routing, resolution verification. Silent-failure discipline at the resolution-verification line.
- **Legal and compliance** — contract intake, clause comparison, regulatory-change tracking, filing-calendar maintenance, privilege-aware document handling. Any deployment touching the legal function must be architected with counsel of record, not alongside counsel of record.
- **Information technology** — ticket triage, access-request handling, patch-window scheduling, log-monitoring summarisation. The function is also the one carrying the operations-and-maintenance discipline for every other function's deployment.

The day-thirty deliverable returns, in writing: the function-by-function read; the recommended sequence of optimization streams (which functions go first, which go last, which do not go at all); the five-risk register for each recommended stream; the operations-and-maintenance posture; and the cost envelope for the next ninety days and the next twelve months. **The deliverable commits neither side beyond the assessment band itself.** The principal reads it, marks it up, and decides whether the program runs.

> *The assessment that survives the boardroom is the one mapped to the organization's* **actual** *org chart, function by function, not to a vendor's reference architecture. Each function gets a read. Each read carries the five-risk register against it.*

IV. *Workflow streams, automation, verification, smoke test, embedded engineer, refresh roadmap.*

The optimization program is the discipline that converts the thirty-day assessment into **an operating capacity the principal can underwrite.** The shape is borrowed from the Secure Enterprise Lifecycle the firm carries inside Practice IV, adjusted for a non-technical operating bench.

Workflow streams — chosen, not invented. A workflow stream is a defined, end-to-end operating sequence the principal already runs by hand. The intake of a new supplier from request to first payable. The intake of a new customer from inbound enquiry to first invoice. The monthly close from trial balance to board pack. The optimization program chooses, from the assessment, the streams that carry the highest ratio of *(recoverable operator hours)* to *(deployment risk under the five-risk register)*. The streams are chosen; they are not invented.

Automation — the agentic capacity, scoped to the stream. The agent is scoped, prompted, and tooled to complete the defined goal on the defined stream, with the defined human-in-the-loop boundary, against the defined system of record. The scope is written down. The tools the agent may call are written down. The post-conditions for goal completion are written down. The escalation paths are written down.

Functional verification, smoke test, go-live. Functional verification is the test, run before the stream goes live, that confirms the agent completes the defined goal against a defined fixture set under the defined post-conditions. The smoke test is the narrower, repeatable check the operator runs every morning before the stream is opened to the day's traffic — a five-minute exercise the operator can do without engineering support, the result of which is logged. The go-live posture commits the principal to a written rollback plan, a written exception path, and **a written ceiling on the agent's daily authority**. Authority grows by a written change order, not by a quiet drift.

Operations and maintenance — the embedded engineer. The O&M posture is the part of the program most often skipped by vendor-led deployments and most often the point at which the deployment fails. The firm's working position is that every agentic stream the principal runs in production carries *an embedded engineer* — a named individual with operating responsibility for the stream, monitoring the audit trail, reading the burn rate, watching the exception register, and authorised to pause the stream the moment any of the five risks crosses a written threshold. The engineer is embedded inside the principal's organization, not loaned from the vendor and not loaned from the firm. The firm's role is to recruit the engineer to a written job description, train against the program's operating annex, certify at a defined cadence, and provide on-call backstop. **The reporting line is to the Group chief information officer; the budget line is the principal's.**

Transition-and-refresh roadmap. Agentic capacities do not stand still. The underlying models, the tools, and the regulatory surface each change on a quarterly cadence. The roadmap commits the principal to a written posture on each: which streams get a quarterly refresh, which get a half-yearly refresh, which get an annual refresh; the model-version pinning posture on each stream; the rollback discipline when a refresh produces a regression; and the planned transition window when a stream is migrated from one underlying capacity to another. **The roadmap is the discipline that prevents the program from becoming a one-time deployment whose drift the principal does not see until the audit surfaces it.**

> *Workflow streams chosen, not invented. Automation scoped to the stream. Verification before go-live. Smoke test the operator runs every morning.* **An embedded engineer with operating responsibility for the stream, on the principal's budget line.** *A transition-and-refresh roadmap that holds beyond year one.*

V. *PPBE — planning, programming, budgeting, execution — and the cloud-migration depth trap.*

The firm carries a planning discipline borrowed from the public-sector budget calendar — **Planning, Programming, Budgeting, Execution** — and calibrated for the small and mid-sized commercial bench. Four windows that produce a cost posture the principal can underwrite without surrendering the agility the AI program is bought for.

- **Planning** — the multi-year posture. What the program is for, which streams it is allowed to touch, which it is not, and what success looks like over the planning horizon. Reviewed annually; revised on a written change order, not on a quiet drift.
- **Programming** — the twelve-month sequence. Which streams are in this year's program, which are deferred, which are in next year's planning window. The bridge between the planning posture and the budget the principal signs.
- **Budgeting** — the line-item cost envelope for the twelve months. Embedded-engineer salary, credit-consumption ceiling per stream, vendor-licence cost, refresh-window cost, on-call backstop. The budget carries a *credit ceiling* that the program does not exceed without a written authorisation from the principal.
- **Execution** — the monthly cadence. What was spent, what was produced, which streams ran inside their five-risk register, which streams crossed a threshold, what change orders are proposed for next month. The execution report goes to the Group chief information officer; the principal reads it on a quarterly cadence.

The cloud-migration depth trap, honestly characterised. The cloud generation, between 2010 and 2014, paid an expensive lesson the firm has carried on the desk ever since. The migration motion compressed quickly; the cost discipline lagged the deployment by a full cycle; the boards that read the cost trap early gained a structural advantage that compounded for the rest of the cycle, and the boards that did not paid the lesson at full price. The agentic cycle is now repeating the pattern in a new register — and a non-technical organization, with a non-technical operating bench, is the part of the market most exposed to it.

The firm's working position is that an agentic program for a non-technical organization should **not, by default, move the organization's operating data into a vendor's cloud at the depth the cloud-migration generation taught the market to do**. The streams the program runs against can, in most cases, be operated with the agent *reaching into* the organization's existing systems of record under a defined access posture, rather than with the organization's data *migrating into* the vendor's environment to be operated on. The discipline is portability of data, modular contract terms, defined exit motions, and a written posture on the data-egress cost the principal would carry if the vendor relationship ended. *The principal who reads the trap early is the principal whose program is still affordable in year three.*

The standing offer to the three readers. For the principal of a non-technical operating company, the firm's standing first conversation is the thirty-day assessment band — fixed-fee, time-boxed, with the written deliverable described in Section III. For the Group chief information officer of a diversified holding, the same band scoped against the holding's operating companies in a defined sequence, with the embedded-engineer recruitment and certification posture written into the deliverable. For the owner of a small or mid-sized business, a compressed version of the same band — sized to the smaller operating bench, calibrated to the smaller revenue line — that returns the same five-risk register, the same workflow-stream selection, and the same PPBE posture. *The deliverable commits neither side beyond the band itself.*

> *Planning. Programming. Budgeting. Execution.* **The principal who reads the cloud-migration trap early — and chooses portability over depth on the day the contract is signed — is the principal whose AI program is still affordable in year three.**

VI. *Field Findings — month one in a real production environment.*

The firm carries one further read the principal should see, because it comes from the firm's own working deployment of the agentic capacity this paper is positioned on. **The numbers are from a real production environment, not a vendor demo**, and they sharpen Section II's rapid-burn-rate risk into a discipline the principal can actually budget against.

VI.1 · *What a thirty-day window looks like inside a real agentic deployment.*

Across a thirty-day window — **23 April through 22 May 2026** — the firm logged **902 commits across 63 distinct project tasks and 71 plan-to-build transitions**. The rework-verb rate on those commits — the share carrying the verbs *Update, Adjust, Refine, Improve, Fix, Polish* — measured **38.6%**. For reference, a third-party citation the firm compiled independently (Google AI Mode's reading of generally-cited figures — *not a vendor-published or peer-reviewed industry standard*) placed typical human-in-the-loop review at roughly **20–30%** and the upper bound of refinement-heavy enterprise prompt-engineering sessions at up to ~90%. The iteration-marker rate (*round N, vN, retry, re-render*) was 1.5%, which reads as in-place editing rather than versioned rounds. The headline 38.6% is real. It is also the wrong number to budget against, because it conflates two categories of cycle that should not be counted the same way.

VI.2 · *The honest split — additive iteration versus corrective rework.*

The firm reads the 38.6% as two distinct categories. **Additive iteration** — the operator asked for an addition, a new direction, a tone change, a scope expansion; the first version was a valid execution of the spec at the time, and the second is a real product decision rather than a fix. **Corrective rework** — the operator asked for X, the system shipped not-X (missed a component, ignored a constraint, used the wrong framework, deviated from a named reference), and the cycle exists because the system did not follow the spec on the first attempt, not because the spec changed.

Three corrective patterns concentrated the cost in the thirty-day window: **wrong-context retrieval**, where the system opened the wrong sibling file in a dense directory of similarly-named scripts and reasoned from it before confirming which one was in scope; **instruction non-compliance on framework-mirroring tasks**, where the operator named an explicit reference ("mirror this", "use the same structure as", "include the picture examples from the prior one") and the deliverable shipped without the named components; and **fabricated causal diagnostics**, where the system claimed a production-state fact — a scheduler firing, a job registration — without first reading the actual state, and then proposed a follow-on hypothesis to explain a prediction that did not hold. The firm's read on the last is that one such incident in the window would, on its own, have cost on the order of **~USD 200 in chase-work against a problem that did not exist** — had the operator not been technical enough to push back on the unverified claim on the turn it was offered.

VI.3 · *Where the cost lands — and the human accountability factor that contains it.*

The cost discipline of effort-based agentic pricing is that **every corrective cycle is paid twice** — once for the first attempt and once for the fix. In the firm's own field log for the thirty-day window, the invoice landed *above the top of the small-business spend bracket described as a "high-intensity startup" profile in the third-party citations the principal compiled independently* — many multiples above the entry-level plan baseline. The firm carries the like-for-like caveat the retrospective itself carries: the workspace running production cron jobs, an inbox-integration backplane, a Postgres-backed console, multiple deployed artifacts, and high-frequency document-generation workloads is *not* a like-for-like peer to a standard-developer profile; the comparison flags an order of magnitude, not a settled overrun number. The structural read survives the caveat. The operator least equipped to spot the deviation on turn one is the operator who carries the largest share of the corrective bill, because the corrective half of the work is paid in compute, in time, and in direct credit each time the prior turn is replayed back into the session's context window on the next turn.

The corollary, and the discipline this paper closes on, is that **the human accountability factor in any agentic program is the proficient operator** — the operator who can read the system's output critically on the turn it is produced, name the deviation against the spec on that same turn, and stop the session before the prior turn is replayed back into the session's context window and the rework compounds. The thirty-day assessment of Section III, the embedded engineer of Section IV, the credit ceiling of Section V, and the proficient-operator discipline named here are the **four checks** the firm reads any non-technical organization's AI program against. A program that carries fewer than four is a program shifting the cost of agent error onto the operator least able to carry it. *The proficient operator is not a luxury bolted onto the program after deployment — the proficient operator is the load-bearing check that makes the other three disciplines hold.*

> *902 commits. 63 tasks. A* **38.6%** *rework-verb rate, split between additive iteration and corrective rework that should not be counted the same way.* **The human accountability factor in any agentic program is the proficient operator** *— and the program that does not place one in the human-in-the-loop position has shifted the cost of agent error onto the operator least equipped to spot it on turn one.*