

# C<sup>2</sup>: The Agentic Multiplier

By Rahul Jindal

---

## The pilot graveyard

Walk into almost any Fortune 500 today and you will find the same artefact: a working AI demo, recorded in a glossy three-minute video, sitting six months past its intended scale date. The chatbot answered policy questions on stage. The agent processed the sample case end-to-end. The CHRO clapped. And then nothing.

MIT's Project NANDA, RAND, Gartner and S&P Global have all published variants of the same number from different angles: somewhere between 80% and 95% of enterprise AI initiatives are delivering zero measurable ROI, or have been quietly shelved. Boards have started asking why. The answers being offered are mostly the wrong answers.

The most common explanation, the one you will hear in every vendor pitch and every analyst report, is that the content is not ready. Clean the wiki. Hire technical writers. Run a documentation transformation. Once the knowledge base is in shape, the theory goes, the agents will work. It is wrong, but it is wrong in a specific and useful way: it points at half of the problem and treats it as if it were the whole.

The other half is context. The agent that answers policy questions correctly on stage is the same agent that, in production, applies an India-only rule to a German employee, or gives an IC the manager view of a workflow, or refuses to act on a delegation that any human in the room would have honoured. The content was right. The situation was wrong. And nobody had built the system that connects the two.

The piece nobody is treating as one problem is the multiplication sign between them.

## The two dimensions

Enterprise agent readiness has exactly two axes. The first is Content Maturity: the quality, completeness, structure and machine-readability of the knowledge an agent draws from. The policies, procedures, decision trees, escalation matrices, workflow definitions,

approval thresholds. The "what" of an answer. The second is Context Maturity: the enterprise's ability to surface and connect the situational signals that govern how that content should be applied. Who is asking. Where they sit in the org. Where they are in a process. What the unwritten rules of this particular organization actually are. The "who, when, why, where" of an answer.

Both are necessary. Neither is sufficient. The interesting claim, the one the rest of this paper is built on, is that the relationship between them is multiplicative, not additive. A high-content, zero-context organization does not get half-credit. It gets a confident wrong answer machine, which is operationally worse than no agent at all because it converts a known limitation into a hidden liability. A high-context, zero-content organization gets the inverse problem: beautiful routing, polite UX, and answers that range from generic to wrong.

$C^2 = \text{Content} \times \text{Context}$ . The squared notation is the point. Zero on either axis produces zero useful agents.

***“Agent readiness is multiplicative, not additive. Zero on either axis is zero output.”***

The mistake most enterprises make is to treat these as sequential projects. Year one: content cleanup. Year two: identity integration. Year three: scale the agent. By the time the identity layer arrives, the content has decayed back to where it started, because nothing in the content investment was set up to surface the contextual rules that were never written down in the first place. The act of adding context is what reveals the content gaps. The act of structuring content is what forces someone to articulate the contextual rules. The dimensions co-evolve. Whoever sequences them loses.

## **The 2×2, and why most enterprises are stuck**

Plot content on one axis and context on the other and you get four quadrants. Each has a name, a population, and a characteristic failure mode.

**Pilot Graveyard.** Low content, low context. Where most of the Fortune 500 actually sits today, no matter what their AI dashboards say. There is a chatbot. The chatbot has a logo. The chatbot is a hallucination machine with branding. The pilots die between demo and deploy because both axes need work and nobody is treating them as one problem. The most common pathology in this quadrant is the executive who confuses "we have a working RAG demo on a cleaned-up subset of our policies" with "we are ready to scale."

**Confident Wrong Answers.** High content, low context. The most dangerous quadrant, and the one the documentation-first transformations end up in. The policy library is excellent. The agent retrieves accurately. The retrieval is then applied to the wrong person, the wrong entity, the wrong workflow state. The agent is now a compliance liability. The CHRO who pushed for the content cleanup discovers that doing the content work alone made the underlying problem worse, because the agent now sounds more authoritative when it gets it wrong. Heavily-regulated industries land here often, because their compliance instinct pushed content investment first and treated identity and state as "phase two."

**Brilliant Routing, Empty Warehouse.** Low content, high context. Less common, but real. Identity-heavy stacks (HRIS-led, IDP-led, customer-data-platform-led organizations) sometimes find themselves here. The agent knows exactly who is asking, where they are in their workflow, who they report to, what they decided last quarter. It just has nothing useful to tell them, because the underlying knowledge is tribal or stale. The plumbing is the most polished part of the system. The product is the worst.

**Genuine Agentic Capability.** High content, high context. Where the actual ambition lives. Where the "agents will displace 60% of routine work" numbers start to become real instead of aspirational. The agent answers, acts, escalates and learns. A handful of frontier teams sit here. Mostly inside AI-native companies and inside aggressive transformation programs that treated content and context as one co-evolving system.

The diagonal matters. Most strategy decks draw an arrow from the bottom-left to the top-right, as if the journey is one continuous line. It is not. The Confident Wrong Answers quadrant is a real trap, and it is the quadrant most expensive to climb out of, because the org has already declared victory on content and built its political narrative around that

declaration. The Brilliant Routing trap is cheaper to escape, because there is nothing to undo, only something to add.

*“The most dangerous quadrant is high-content, low-context. The agent gets more confident as it gets more wrong.”*

## The Content Maturity Ladder

Content Maturity has five levels. They are not optional. You cannot skip one. You can, however, declare you are at a level you are not, and most enterprises do.

**L1 — Tribal.** Operational knowledge lives in senior people's heads, in Slack threads, in a folder of one-off emails that nobody else has access to. Nothing is canonical. If your three most senior ops leaders left tomorrow, the institutional memory would leave with them. The honest first move at L1 is an audit: what actually exists, and where. Agents cannot operate here at all, and the temptation to skip to a documentation project before the audit is what produces the bloated, contradictory wikis that define L2.

**L2 — Documented.** Written down. Confluence, SharePoint, an intranet, sometimes a vendor knowledge base. Unstructured, duplicated, stale, contradictory across pages. The document for the policy still exists, but the most recent change is documented somewhere else, and three pages disagree on the threshold. Most large enterprises sit at L2 and mistake it for readiness. The agent you build on top of L2 produces confident wrong answers. RAG demos look great. Production is dangerous. The trap is believing that more documentation effort fixes the issue. The issue is structure, not volume.

**L3 — Structured.** Atomic, tagged, version-controlled units of content. Policies are separated from procedures. Decision logic is extracted from paragraphs and made explicit. There is one canonical answer for each question, and you can find it in under thirty seconds. RAG starts working at L3. This is the level where agents become actually useful, on a bounded set of well-structured processes. The trap at L3 is declaring victory. Structured content is a real achievement. It is also still narrative; it does not let the agent do anything beyond answer.

**L4 — Machine-actionable.** Content is not just readable. It is executable. Decision trees live in structured formats. Approval matrices are tables. The workflow has defined inputs, outputs, branching, and a contract the agent can call. Agents graduate from answering questions to doing work. This is where the "agents will displace 60% of routine work" numbers start to be defensible. The hard part of L4 is not the engineering. The hard part is forcing process owners to articulate the rules they have never written down, in language unambiguous enough to be executed. The political friction is what makes L4 the longest phase of the journey for most enterprises.

**L5 — Self-maintaining.** Content has owners, review cycles, usage telemetry, automated staleness detection. The corpus improves through use, not despite it. Every agent error is logged against the content unit that produced it. Owners triage. Corrections feed back into structured knowledge. The system gets sharper monthly. L5 is the only level that resists decay; without it, an L4 corpus rots to L2 within eighteen months because content systems do not naturally maintain themselves and nobody is paid to do it.

A short self-test: pick the policy that changed most recently in your organization. How many days from the change being decided to the canonical content reflecting it correctly across every page that depends on it? At L2 the answer is months. At L3, weeks. At L4, hours. At L5, the change is the publish event, and the dependent workflows update automatically.

## **The Context Maturity Ladder**

Context Maturity also has five levels. Most enterprises have never tried to build past L2. This is the unbuilt continent of enterprise AI.

**L1 — Anonymous.** The agent treats every user identically. First-generation chatbot territory. Useful for the most generic questions only. Anything that depends on who is asking breaks immediately. Calling an L1 system "agentic" is a category error. It is a search box with a personality.

**L2 — Identity-aware.** The agent knows basic attributes: name, role, department, location, business unit, entity. It can filter retrieval by geography, refuse requests outside the user's permissions, personalize the obvious. Most enterprise AI pilots live at L2 and

treat the SSO integration as the finish line. It is the floor of context maturity, not the ceiling. The trap is believing identity is the same as context. It is the smallest part of context.

**L3 — State-aware.** The agent knows where the user is in a workflow. In-flight processes, completed steps, pending approvals, blocking dependencies. The agent can pick up a conversation mid-process without the user restating the situation. Workflow state is what turns a chatbot into a participant. L3 is the level where you stop asking users to re-explain themselves and start actually saving them time. The integration debt to get here is real; the most common failure mode is bolting state on top of a stateless architecture instead of redesigning around it.

**L4 — Relationship-aware.** The agent understands org hierarchy, reporting lines, delegation authority, team composition. A manager asking "my team's leave balance" is a different question from an IC asking "my leave balance," and the agent knows the difference without being told. Skip-levels, acting roles, dotted-line relationships, delegated approvals. L4 is where agents start handling manager workflows correctly and routing approvals to the actual decider rather than the nominal one in the org chart. The trap is stopping at the formal org graph; the informal one is where the real decisions happen.

**L5 — Judgment-aware.** The agent has access to historical decision patterns, exception precedents, the unwritten norms of how this organization actually behaves. The "we tend to be strict on this" vs "we tend to be lenient on that." It handles edge cases the way a 10-year veteran would: reads precedent, weighs unwritten norms, explains its call. L5 is what separates a sophisticated lookup tool from a genuine agentic system. It is also the level almost no enterprise has attempted to build. The reason: it forces explicit articulation of organizational norms that are currently held implicitly, and that articulation is politically uncomfortable. The enterprises that do the work get an agent that can be trusted with judgment. The ones that do not get an agent that needs a human to ratify every non-trivial call.

*“L4 and L5 of the context ladder are the unbuilt continent of enterprise AI.”*

## **Why you cannot sequence them**

The instinct in every transformation program is to sequence. Fix content first, then add context. Get the knowledge base clean, then wire in identity, then add state. The instinct is wrong, and the reason it is wrong is the most important point in this paper.

Adding context is what reveals the content gaps. Until the agent knows the user is a manager, you do not know your policies are silent on the manager case. Until the agent knows the user is in Germany, you do not know your content has merged India-specific rules into the global page. The contextual lens is the diagnostic that surfaces what is missing from the content corpus. Build content in isolation and you will build for the wrong cases.

Structuring content is what forces articulation of the contextual rules. The act of extracting a decision tree from a 40-page policy forces someone to write down the rules that were previously held implicitly in the head of the senior person who used to answer the question. Those rules, once written, are the contextual conditions: "if the request is from a manager with five or more reports, route to skip-level approval." The content work is what produces the context schema. Skip the content work and the context system has nothing to act on.

The two dimensions are coupled. The reason most enterprise transformation programs fail is that they are organized as sequential projects with separate owners: a knowledge management workstream and a systems integration workstream. Neither workstream is set up to surface what it learns to the other. The artefact each produces is right by its own lights and useless when combined. The real unlock is to treat both as a single modeling problem: build a machine-readable model of how the organization actually operates, including the messy parts nobody wants to write down. That model has both content units and contextual conditions in it, and the work of constructing it forces the two to develop together.

The framing shift is from documentation project to modeling discipline. A documentation project ends. A modeling discipline is an operating capability.

## **The three-phase path**

Given that most enterprises sit at Content L2 / Context L1-L2, the path forward has three phases. They are not optional and they should not be reordered.

**Phase 1 — First useful agents.** Target: Content L3, Context L2, in parallel. Pick the top twenty processes by volume, not all five hundred. Structure their content end-to-end. Wire identity, entity and geography in parallel. The unlock is agents that are 85%+ correct on a bounded set of high-volume processes, with real users at real volume. The hard part is political: forcing the rest of the organization to live with "only these twenty processes are supported" while the muscle is being built. Most programs fail at this step by trying to cover everything and ending up with nothing that works.

**Phase 2 — Action, not just answer.** Target: Content L4, Context L3. Make content executable. Connect the agent to workflow state. Agents move from answering questions to doing work. This is the hardest engineering phase. The integration debt is real and most of the cost is in connecting agents to systems of record without rebuilding the systems of record. The political work continues: process owners must articulate decision rules with enough precision to be executed. Some of those rules will turn out to be wrong, or inconsistent, or quietly racist, and the articulation is what surfaces it. The act of structuring is the act of auditing.

**Phase 3 — Adaptive system.** Target: Content L5, Context L4-L5. Self-maintaining content. Relationship and judgment awareness. This is the eighteen-to- thirty-six-month horizon. This is where the "agents will displace 60% of routine work" numbers actually live, and this is the level almost no enterprise has reached. The hard part is institutional: content ownership has to become part of the operating cadence, telemetry review has to become a ritual, precedent capture has to become an operating discipline. Phase 3 is not a project. It is a capability. The organizations that build it will compound on it for a decade. The ones that treat it as a one-time investment will watch their L4 corpus decay back to L2 inside two years.

The phases are sequenced but the axes inside each phase are parallel. That is the entire trick. You move both ladders one rung at a time, in lockstep, on a bounded scope. The enterprises that try to push content from L2 to L5 before touching context are the same enterprises whose AI pilots are dying in slow motion right now.

## **The diagnostic**

The C<sup>2</sup> diagnostic is twelve questions, six per axis. Roughly four minutes. It places your organization on the 2×2, shows your level on each ladder, identifies the binding constraint (the axis that is currently your floor), and recommends the phase that comes next. It is a starting point, not an audit; a shared vocabulary that a transformation team and a board can use to agree on where the organization actually sits rather than where it prefers to think it sits. Most leadership teams find their formal answer (what the AI dashboard claims) and their lived answer (what the diagnostic surfaces) are at least one level apart on both axes.

The diagnostic does not solve the problem. It does the smaller, more useful thing of making the problem legible. Most enterprise AI programs fail not because the right move is unknown but because nobody at the top of the organization has agreed on the diagnosis. C<sup>2</sup> is built to be the artefact a CHRO, a CIO and a CEO can look at together and arrive at a shared read of the same building.

**Take the C<sup>2</sup> diagnostic** →

## **From documentation project to modeling discipline**

Every framework in this body of work converges on the same point. The Margin Thesis says agentification only displaces budget lines when content and context are mature enough to replace human work, not just assist it. Organizational Metabolism says the binding constraint on AI absorption is metabolic rate, not capability. Human Ether says the productivity gap between theoretical capacity and real output is the largest hidden lever in the modern organization. C<sup>2</sup> is the operational layer underneath all of them. It is the readiness math that decides whether the rest is theoretical.

The strategic stake is who builds this capability. Treat content and context as one co-evolving model and you get an organization that can absorb every new model generation on a cadence of months rather than years. Treat them as separate IT plumbing projects and you get an AI portfolio that looks busier every quarter and produces less every year. The first organizations to pass Phase 2 will compound on it. The ones still sequencing their

content cleanup will be the case studies in the next decade's books on enterprise transformation.

The shift required is not technical. It is conceptual. Stop treating agentification as a documentation project on one side and a systems integration project on the other. Start treating it as the work of building a machine-readable model of how the enterprise actually operates. Including, especially, the parts that have never been written down.

That model is what an agent runs on. C<sup>2</sup> is how you measure whether it exists.

---

Shared privately. Please do not redistribute.