

The ABCs We Must Learn, by Hand

By Rahul Jindal

Almost every large company now has AI. Almost none of them have the return. McKinsey's 2025 state of AI survey found that about 88% of companies use AI in at least one function, yet only 39% report any earnings impact, and most of those say it is under 5%. BCG put it more bluntly the same year: only 5% of companies are getting value from AI at scale, and 60% are getting nothing material at all. The MIT NANDA team gave the cycle its headline number, 95% of enterprise generative AI efforts showing no measurable bottom line within six months.

Read those numbers carefully and a pattern shows up. The failure is almost never the model. The MIT team named the real cause the learning gap: tools that do not connect to how the business runs. The company bought the intelligence and skipped the two things that make intelligence useful. We learned our A. We forgot our B and our C.

So here are the ABCs every operator now has to learn. AI is the intelligence. BI is Business Intelligence, the structured truth we spent twenty years building. CI is Context Intelligence, the new layer that teaches a machine how your specific business actually works. Transformation needs all three. Most programs are running on one.

There is a fourth thing the letters do not capture, the one in the picture above. Three blocks, and a hand keeping them upright. The blocks are the ABCs. The hand is what holds them the moment the agent stops answering questions and starts taking actions. We will get to the hand near the end, because it is the part that decides whether all of this is safe to switch on.

A is for AI. It is the part you rent.

The strange thing about this moment is that the hardest part is now the easy part. Frontier models are extraordinary, and they are available to your competitor on the same terms they are available to you. Intelligence has become a utility. You pay by the token and it shows up.

That is exactly why AI on its own moves so few numbers. A general model knows everything about the world and nothing about your world. It has never seen your definition of an active customer, your close process, your escalation rules, or who is allowed to approve a refund. Ask it a question that depends on any of those and it will answer with total confidence and no idea whether it is right.

There is a second trap here that most leaders have not internalized yet. A smarter model does not fix a missing context problem. It makes it worse. A weaker model gives you an answer that is obviously off, so you catch it. A stronger model gives you an answer that is fluent, well structured, and wrong in a way no one notices until it has shipped. Capability without context does not reduce risk. It disguises it.

B is for BI. It is the part you already built.

Business Intelligence is the discipline most companies have been investing in for two decades. Data warehouses, pipelines, dashboards, and the semantic layer that says what a metric means. The whole point of BI was to take messy operational data and turn it into one trusted version of revenue, churn, and headcount that a human could read on a chart.

That work is not wasted. It is the foundation. When a model writes a query against your raw tables with no semantic layer, it guesses at what your columns mean and produces plausible nonsense. Snowflake published the cleanest version of this with its own tooling: a strong general model scored 51% on business questions against raw schemas, and around 90% once a semantic model told it what the terms meant. Treat the exact figure as vendor friendly, but the direction is real and every serious lab reports the same shape. Definitions move accuracy more than model size does.

Here is the limit. BI was built for a human sitting in front of a dashboard. The human supplied the missing context from memory. They knew that the finance definition of bookings differs from the sales one, that the West region data is two days behind, that you never trust the draft version of the pricing table. None of that judgment is written down. It lives in the analyst's head. BI gives an agent the metrics. It does not give the agent the mind of the person who knew how to use them.

C is for CI. It is the part almost no one has built.

Context Intelligence is the layer that turns the knowledge in people's heads into something a machine can use. The phrase going around the industry is context engineering, and the people who named it were precise about what they meant. Tobi Lutke, the Shopify CEO who popularized the term, called it the art of providing all the context needed for the task to be plausibly solvable by the model. Andrej Karpathy described it as filling the context window with just the right information for the next step. Anthropic frames context as a finite resource with an attention budget, where every extra token of noise costs you a little accuracy, so the goal is the smallest set of high signal information that gets the job done.

In an enterprise, that high signal information comes in three kinds.

Knowledge: what things mean.

The entities, definitions, metrics, and relationships of your business. What counts as a case. What counts as resolved. How a customer in the CRM maps to the same customer in billing and support. This is BI's territory, extended so a machine can resolve it rather than a human.

Expertise: how work gets done.

The playbooks. How you actually run the monthly close, an investigation, an accommodation request. The triggers, the edge cases, the order of operations your best people follow without thinking. Today this is tribal knowledge. It walks out the door when they do.

Norms: what is allowed.

The policies, permissions, approval paths, and compliance limits that decide what an agent may and may not do. In a dashboard a wrong permission is an annoyance. For an agent that can act, it is an incident.

“Intelligence is the part you rent. Context is the part you own. The moat is what you own.”

Why does this layer matter so much in practice? Because attention is finite and long context decays. Stanford researchers documented the lost in the middle effect, where a

model uses information at the start and end of its context far better than anything in the middle, sometimes performing worse than if you had given it no documents at all. Chroma's 2025 work named the broader version context rot: as you stuff more in, reliability falls, even on simple tasks. The answer is not a bigger context window. It is a system that selects the right context for the question and leaves the rest out. That system is what Context Intelligence is.

Signal and noise: what to believe, what to discount

A warning, because this space is loud. Almost every data vendor now sells a context layer, and they do not mean the same thing by it. For some it is a semantic layer for metrics. For others it is a governance and lineage shell, or agent memory, or a data catalog, or a piece of plumbing exposed through a protocol. One phrase is being stretched across at least five different products. When a vendor tells you they are the context layer, your first question is which of those five they actually built.

The signal underneath the noise is solid, and it does not come from the marketing. It comes from the labs and the research. The need for context is real, it is measurable, and it is the difference between a demo and production. The productized context layer is a contested category where ten companies are racing to own the word. Hold both thoughts at once. Build the capability. Stay skeptical of anyone who says they can sell it to you whole.

The three letters multiply, they do not add

The reason most programs stall is that they treat these as a shopping list. Buy AI, maybe refresh BI, skip CI. But the three do not add up. They multiply. A great model on top of clean metrics with no business context is still confidently wrong. Rich context with no trusted metrics underneath is grounded in garbage. And perfect data with no model on top is a dashboard nobody reads.

Transformation is AI times BI times CI. If any one of them is near zero, the product is near zero, no matter how large the other two are. That single idea explains the 95% better than any other theory I have seen. Most of those failed programs scored high on A, medium on B, and zero on C.

The hand that holds them: the harness

There is a moment the ABCs do not cover. It arrives the instant the agent stops answering and starts acting.

Up to that point, the three letters are about knowing. AI supplies the reasoning, BI supplies the numbers, CI supplies the meaning. Put them together and the agent knows enough to give a good answer. But knowing is not doing. The day the agent files the case, sends the email, or approves the refund, a new question shows up that none of the three letters answer. What happens when it is wrong and no one is watching?

That is the job of the hand. In the picture at the top of this piece, three blocks are stacked and a small figure keeps a hand on them. The blocks are the ABCs. The hand is everything that keeps them from toppling once the agent runs on its own: the check that catches a confident wrong answer before it ships, the limit on what the agent is allowed to touch, the feedback loop that turns each mistake into a rule so the same mistake cannot happen twice, and the log that lets a human see what the agent actually did.

The industry has started to name this layer. After prompt engineering and context engineering comes harness engineering. Birgitta Böckeler of Thoughtworks put the definition in its simplest form: an agent is a model plus a harness, and the harness is everything around the model that raises the odds it gets the answer right the first time and catches the misses before a human ever sees them. She splits the hand into two jobs. Guides that steer the agent before it acts, and sensors that catch what slips through after.

Notice the through line. A prompt shapes one instruction. Context shapes one decision. The hand governs the thousand decisions the agent makes while you are asleep. That is why it cannot be an afterthought for anything that acts on its own.

The rule that makes the hand compound mirrors the one you use on context. A wrong fact is a context bug, and you fix the definition underneath it. A wrong action is a harness bug, and you fix the hand. Do not reach for a bigger model in either case. Repair the layer that failed, and it stays fixed for every agent that comes after. The two kinds of bug get mended in different places. Both stop coming back.

This is where the title earns its second meaning. The hand is built by hand. No vendor can sell you the map of which of your agent's actions need a human gate, which need a hard stop, and which need only a quiet log. You write those down the same way you wrote your definitions, one honest rule at a time. Rented intelligence sits on top. The context and the hand underneath are both yours, and both are the moat.

So the arithmetic gains one term, but only for agents that act. AI times BI times CI gives you an answer. The hand is what makes that answer safe to release into the business. For an agent that only informs a person, that person can be the hand. For an agent that acts on its own, the hand is the difference between help and an incident.

Where to start

Do not boil the ocean. Pick one process you run often and cannot afford to get wrong, and one that is forgiving enough that a mistake while you learn is not a disaster. Then mine what you already have for it: the logs, the macros, the procedures your best people wrote, the definitions your data team already agreed. That is most of a first draft of the context, before anyone writes a single new page.

Now set the loop that makes it compound. Let the AI read your systems and propose the definitions and the rules. Have a senior person approve, edit, or reject each one, because a wrong definition promoted without a human is how the whole thing loses trust.

Autonomous discovery, governed promotion. And when the agent gets something wrong in production, do not reach for a bigger model. Find the missing definition or the stale rule, fix it once, and it stays fixed for every agent after. Done right, the tenth agent is sharper than the first, because the layer underneath learned something each time.

Key takeaways

If you are trying to move a leadership team off the model conversation and onto the one that matters, these are the lines that tend to land.

- Intelligence is rented. Context is owned. The moat is what you own.
- A smarter model does not fix bad context. It just lies more convincingly.
- Every wrong answer from an agent is a context bug, not a model bug.

- We do not have a model gap. We have a trust gap.
- Stop prompt engineering. Start context engineering, then harness engineering. Prompts are disposable. Context compounds. The hand is what lets it act.
- The ABCs are what the agent knows. The hand is what stops it acting on what it got wrong. Build both, or do not let it act.
- A wrong fact is a context bug. A wrong action is a harness bug. Neither is fixed with a bigger model.
- A definition without an owner is a future incident.
- The playbook in your best person's head is the most valuable and least governed asset in the company. Write it down or it walks out the door.
- Transformation is AI times BI times CI. Score a zero on any one letter and the product is zero.

The work for the next decade

The companies that win the next decade will not be the ones with the best model. Everyone will rent the same model. They will be the ones whose context compounds, where each agent makes the next one smarter, and whose hand is steady enough to let those agents act. That work is unglamorous. It is writing down what your business actually means, who is allowed to do what, and how the work really gets done, in a form a machine can read. Then it is building the hand that holds all of it upright. Both are done by hand. Both are yours.

We spent a decade learning the A. The B has been sitting in our data teams the whole time. The C is the letter that decides whether any of it pays off. The hand is what you build the day you let it act. Learn all three, in order, keep a hand on them, and the 95% number stops being a verdict on AI. It becomes a description of everyone who only learned one letter.

Companion papers

Three longer-form working papers take the pieces of this essay further. Also collected under Papers.

- **Building the Context Layer Inside People Operations** →The foundation for HR AI that works in production.
- **The Global Process Owner in an Agentic World** →From doing the process to certifying the agents that run it.
- **How Content and Knowledge Teams Evolve** →From writing documents to owning the context agents run on.

Take it with you

Email this as a LinkedIn pack

Get a feed-ready LinkedIn post (under the 3,000-character cap), a long-form LinkedIn article version, the hero image, and an editable document version of the full essay, delivered to your inbox. Ready to post.

Email this

Shared privately. Please do not redistribute.