AI STRATEGY & EXECUTION

Building with AI Is Easy. Winning with It Is Hard.

Andy Kannurpatti
Partner, Strategem AI
April 2026

Your team can build an AI-powered application this weekend. A capable engineer working with today's large language model APIs, agent frameworks, and no-code platforms can have a functional prototype running in hours. A non-technical founder with a clear problem statement and access to the right tools can have something demo-ready in days. This is real, and it represents one of the most significant shifts in the economics of software in a generation.

That is not the hard part anymore.

The hard part: understanding what to build, getting people to actually use it, and keeping it running reliably when the business depends on it. That part has not changed. It has, if anything, become harder to see. Because when building is easy, it is tempting to confuse a working demo with a working solution. Most enterprises today are making exactly that mistake, and they are paying for it quietly, in a currency that does not show up on the pilot's budget line: in the AI initiatives that never made it to production, in the systems that launched and went unused, in the pilots that generated impressive slide decks and no measurable change in how the business operates.

The Prototype Moment

The tools available today are genuinely remarkable. OpenAI, Anthropic, Google, and a rapidly expanding ecosystem of open-source models have made frontier AI capabilities accessible via API. Orchestration frameworks like LangChain, LlamaIndex, and emerging agentic platforms have simplified the construction of complex AI workflows to a level that would have been inconceivable three years ago. Retrieval-augmented generation has made it straightforward to ground an AI system in proprietary organizational data without fine-tuning. Vector databases, embedding models, and multi-modal capabilities are available as commodity cloud services.

The result is that the barrier to entry for building a working AI prototype has effectively collapsed. This matters enormously, and it changes the strategic question that executives should be asking.

The question is no longer: Can we build this?

The question is: What should we build? How will people actually use it? And do we have the operational capability to run it reliably at the scale and stakes our business requires?

Most organizations are still asking the first question. The companies that figure out the second three will capture most of the value.

The Three Hard Problems

System design: solving the right problem

The most dangerous failure mode in enterprise AI is not that the technology fails. It is that the technology succeeds at solving the wrong problem.

Building a functional AI system requires engineering skill. Deciding what that system should do — which decisions it should inform, which workflows it should change, which users it should serve, and how it should behave when the data is messy or the situation is ambiguous — requires something different. It requires genuine domain expertise: the kind of deep, accumulated understanding of how a business actually operates that cannot be extracted from a job description or reverse-engineered from a process map in a two-week discovery sprint.

AI systems that work technically but are poorly designed for the actual workflow of the people using them fail expensively and quietly. The organization writes off the initiative. The lesson drawn is often that "AI doesn't work here," when the correct lesson is that the wrong problem was solved. This is not an engineering failure. It is a strategy failure, and it is far more common than any AI vendor has an incentive to acknowledge.

User adoption: the behavior change problem

Most AI implementations fail not because the technology fails, but because human behavior does not change.

This is not a new observation. It is the same organizational reality that has bedeviled every major enterprise software implementation for forty years. The difference with AI is that the technology is so immediately impressive in a demo setting that it systematically leads organizations to underinvest in the change management required to make adoption real.

An AI system that generates accurate, useful outputs but that the people who were supposed to use it have quietly worked around is not a system. It is infrastructure spend with no return. The organizational change management required to move from a successful pilot to a genuinely adopted operational tool — including the training, the workflow redesign, the incentive alignment, the leadership commitment, and the iterative improvement cycle — is typically the largest cost in a successful AI implementation, and the most frequently underestimated.

The companies that have successfully deployed AI at scale will tell you the same thing: the technology was the easy part.

Scalability and reliability: operating in the real world

A prototype that works for ten users in a controlled demo environment is not the same as a system handling ten thousand users in production, with real data, real edge cases, real latency constraints, and real consequences for failure.

The engineering required to take an AI system from demo to production is a fundamentally different skill set from the engineering required to build the demo. It involves observability: knowing when the system is behaving unexpectedly before a user files a complaint. It involves cost architecture: a system with a $50 per month API cost in development can consume $500,000 per month at enterprise scale if the token economics were not designed into the architecture from the start. It involves graceful degradation: knowing how the system should behave when an upstream model is unavailable, when input data is missing, when a user query falls outside the designed operating envelope. It involves latency management, version control across model updates, data pipeline reliability, and the security and compliance posture required to operate in regulated industries.

These capabilities are not learned from documentation. They are learned from operating systems at scale, under real conditions, with real accountability when something goes wrong. The discipline is not unlike what industrial companies call operations management: the accumulated expertise of running complex systems reliably, efficiently, and safely over time. It is not glamorous work. It is the work that determines whether the business value materializes.

The Commodity Trap

When everyone can build a prototype, the prototype itself is worth nothing. This is the paradox of AI democratization: the same forces that have made AI accessible to everyone have made AI capability, by itself, competitively irrelevant.

Competitive advantage no longer comes from having built something AI-enabled. It comes from having built the right thing, connected it to the right workflows, gotten the right people to actually use it, and sustained it in production at a cost and reliability level the business can depend on.

The evidence of this is already visible in enterprise AI spending patterns. Analysts estimate that the majority of corporate AI pilots do not reach production. The initiatives that do reach production disproportionately share a common characteristic: they were not just built competently; they were designed with genuine domain knowledge, deployed with deliberate change management, and operated with the engineering discipline of a production system from the beginning.

Boards and customers are becoming better at distinguishing the organizations that have done this from those operating impressive demo environments. The window for trading on demo quality is closing.

Where the Moat Is

The organizations that will create sustained competitive advantage from AI are those that have, or can build, a specific three-part combination.

The first is **domain expertise**: an accumulated, granular understanding of how a specific industry or function actually operates. The real decision flows, the data that matters and the data that looks like it matters but does not, the edge cases that a well-designed system must handle and a poorly-designed one will not, the regulatory and organizational constraints that determine what can be deployed and what cannot. This knowledge is not in a model. It cannot be purchased from a vendor. It is built over years of operating in a domain.

The second is **operational engineering discipline**: the skills and habits of running complex systems reliably at scale. This is the site reliability engineering mindset applied to AI systems: not just building something that works, but designing it to fail gracefully, monitoring it continuously, and improving it iteratively. The companies that have this in manufacturing, logistics, financial services, and healthcare have been building it for decades. Applying it to AI systems is a translation exercise, a difficult one, but a more tractable problem than acquiring domain expertise from scratch.

The third is **AI fluency**: not just the ability to use AI tools, but the judgment to know which tools are appropriate for which problems, how to evaluate their outputs, how to compose them into systems that are more reliable and useful than their individual components, and how to evolve those systems as the underlying models change every twelve to eighteen months. The tools will continue to change rapidly. The judgment about how to deploy them well is the durable capability.

Each of these three, taken alone, is increasingly common. Domain experts are everywhere. Organizations with strong operational engineering capability exist across industries. AI technical skill is proliferating rapidly. The combination of all three, working together and genuinely integrated, remains rare, and it is structurally difficult to replicate quickly.

That combination is the moat.

The Executive Decision

The decision facing executives right now is not whether to invest in AI. That question has been answered. The question is whether the investment is being made against the right problem.

Stop measuring AI progress by the number of pilots launched. Start measuring by the number of operational AI systems running reliably in production, used by the people they were designed for, with measurable impact on the decisions or workflows they were built to improve.

Ask a harder version of the question your team is answering. Not: *Can we build this?* But: *Do we understand the problem deeply enough to design this well? Do we have the organizational capability to get people to use it? Do we have the engineering discipline to run it reliably at the scale our business requires?*

If the answer to any of those is uncertain, the pilot budget is not the gap. The capability is.

A Different Kind of Partner

The organizations we work with at Strategem AI are typically not struggling to find people who can build AI applications. They are struggling to find partners who understand their domain deeply enough to help them design the right application, who have the operational engineering discipline to take it from pilot to production, and who have the AI fluency to make it work and keep it working as the technology evolves.

That combination of domain knowledge, operational discipline, and AI fluency is what Strategem AI was built to provide. If you are navigating the gap between an impressive AI pilot and a system that genuinely changes how your business operates, we would welcome the conversation.

strategem.ai

Andy Kannurpatti is a Partner at Strategem AI, where he works with executive teams at the intersection of strategic operations, capital discipline, and AI-enabled transformation.

References & Notes

[1] McKinsey Global Survey on AI, 2024/2025: Consistent finding that a significant portion of enterprise AI pilots do not scale to production.

[2] Gartner Hype Cycle for Artificial Intelligence: Enterprise AI positioned in or approaching the "trough of disillusionment" phase for scaled deployment, distinct from prototype/pilot maturity.

[3] Industry cost benchmarks for LLM API usage at scale: Token costs that appear trivial in development environments routinely generate material cloud spend surprises at production scale without explicit cost architecture.

[4] LangChain, LlamaIndex, and equivalent frameworks: open-source orchestration tooling that has materially reduced the engineering effort required to build RAG-based and agentic AI applications since 2023.