Context Engineering for AI

A Strategic Framework for Building Reliable, Useful, and Governable AI Experiences

Purpose: To outline why Context Engineering is essential for enterprise AI success and provide a practical framework for turning AI experimentation into reliable business capability.

Executive Summary

Context Engineering is an engineering discipline that makes inference engines based on generative AI reliable and governable in enterprise settings. Many organizations have adopted the use of inference engines using large language models (LLMs). However, these often underperform because responses are not consistently grounded in trusted data, aligned to policy, or improved through feedback. This paper defines Context Engineering and outlines Axionix's approach to building AI experiences that are defensible to regulators, explainable to stakeholders, and safe to scale into real workflows.

GenAI programs that stall at pilots or fail in production are a major business problem. Typical failure modes such as hallucinations, stale answers, inconsistent outputs and trivial responses from "demo prompts" quickly become a significant business risk in regulated or customer-facing work, or may introduce policy violations, incorrect guidance, avoidable escalations, rework, and reputational damage that erodes adoption. Root causes are structural: fragmented knowledge, weak retrieval, poor information architecture, and ambiguous instructions that let models improvise where they should be constrained.

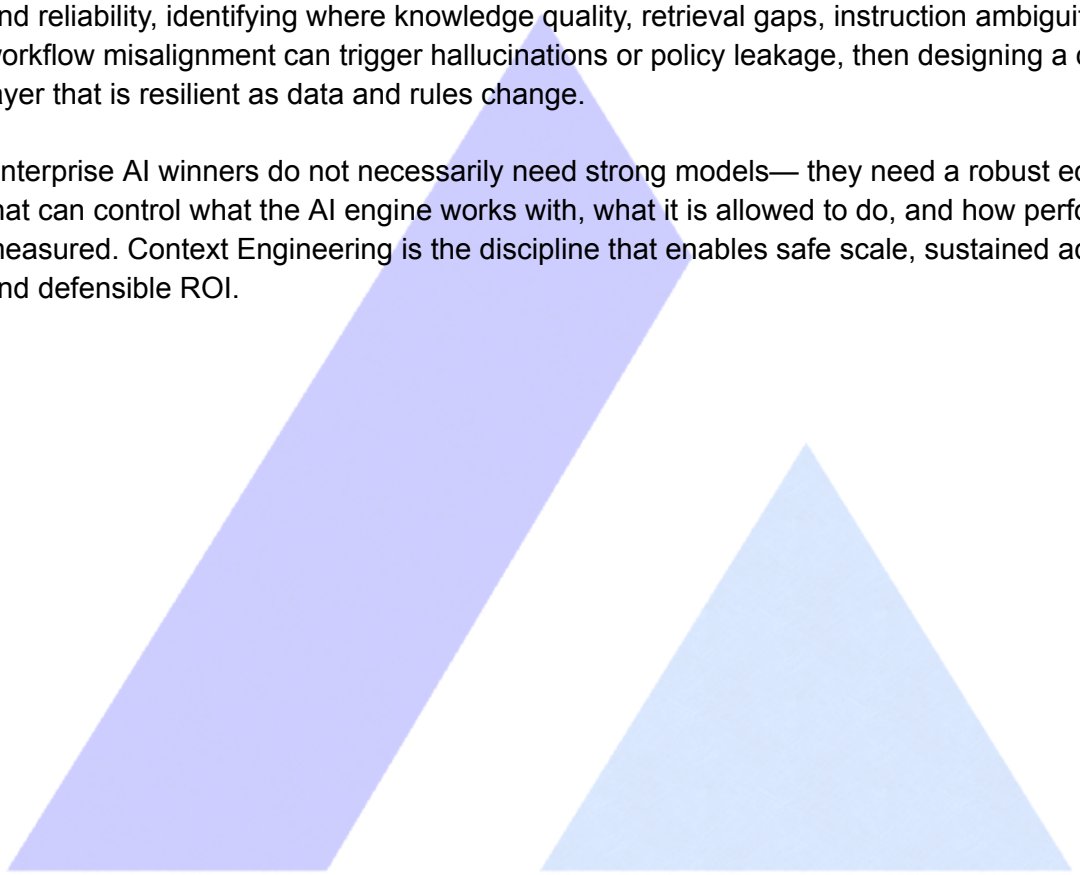
Context Engineering is the systematic process of **dynamically assembling the right evidence and instructions** for an LLM to produce accurate, relevant, and explainable outputs for a specific task. It operationalizes trusted knowledge sources, the pipelines that keep them current, retrieval mechanisms that select the right context per query, and structured prompts/policies that limit unsafe assumptions. Compared with repeated model tuning, context injection is typically more defensible and responsive—reducing cutoff/drift exposure, enabling rapid policy and content updates, and improving auditability through traceable context—while explicitly managing tradeoffs in latency, cost, context-window limits, and hallucination probability.

As organizations move from chatbots to embedded AI and autonomous agents, context complexity—and exposure—rises. Generic prompting is insufficient when AI must act on

enterprise knowledge, operate across tools, and comply with policy while producing repeatable outcomes. Without engineered context and evaluation, teams often discover gaps through incidents: compliance breaches, customer harm, brand damage, and costly remediation. Achieving success needs measurement: early indicators such as groundedness, accuracy, consistency, hallucination/error rates and escalation/editing burden; and outcome indicators such as reduced rework and support cost, stronger audit readiness, fewer disputes, faster cycle times, and ROI.

Axionix's **C.O.N.T.E.x.T.** framework matures context engineering from ad hoc experimentation to a governed operating discipline by aligning knowledge architecture, retrieval strategy, prompt/policy controls, workflow design, and continuous evaluation. The focus is risk reduction and reliability, identifying where knowledge quality, retrieval gaps, instruction ambiguity, or workflow misalignment can trigger hallucinations or policy leakage, then designing a context layer that is resilient as data and rules change.

Enterprise AI winners do not necessarily need strong models— they need a robust ecosystem that can control what the AI engine works with, what it is allowed to do, and how performance is measured. Context Engineering is the discipline that enables safe scale, sustained adoption, and defensible ROI.



1. The Problem: Why AI Efforts Underperform

According to a report from MIT, only 5% of AI initiatives undertaken by business organizations had a non-negative ROI. The report suggests four patterns that define the 'divide' between successful implementations and failed ones:

- Limited disruption: Only 2 of 8 major sectors show meaningful structural change
- Enterprise paradox: Big firms lead in pilot volume but lag in scale-up
- Investment bias: Budgets favor visible, top-line functions over high-ROI back office
- Implementation advantage: External partnerships see twice the success rate of internal build

The core barrier to scaling is not infrastructure, regulation, or talent. It is learning. Most GenAI systems:

1. *Do not retain feedback*
2. *Do not adapt to context*
3. *Do not improve over time*

The core learning barrier may be addressed by a suitable **context engineering approach** to enhance the capabilities of GenAI implementation. Context engineering leads to more accurate, meaningful and nuanced responses that help establish trust in the system and generate positive ROI for business.

Common symptoms of poor context design

- Responses that sound polished but are trivial on closer inspection
- Inaccurate and / or outdated responses
- Hallucinations or unsupported recommendations
- Inconsistent answers with similar questions
- Prompts that work in demos but fail in production
- AI experiences that do not match user intent, business rules, or workflow realities
- Low user trust and weak adoption

What leads to poor context design?

- Failure to implement a robust context retrieval strategy
- Failure to handle a diversity of knowledge sources
- Weak information architecture
- Ambiguous or conflicting prompt
- Overfocus on model choice instead of task context

Without Context Engineering, organizations may scale AI access without reliably scaling AI adoption.

2. What Is Context Engineering?

The Need for Context

LLMs are trained with publicly accessible content. The most common way to use a LLM is to use it as a Generative Pre-trained Transformer (GPT), where given a query or a task (usually referred to as a “prompt”), the LLM generates a meaningful output that is relevant to the query being made. Over the years LLMs have become better at generating an output from a prompt.

Public, commercially available or open-sourced LLMs have the following limitations:

- ➔ The responses generated by the LLM are generic and not specific to the business or enterprise
- ➔ Outdated information, leading to the classic problem of *model cutoff date* and *model drift*

These limitations constrain the use of the LLMs in a business or enterprise context. These limitations may be mitigated in two ways:

- ➔ “Tune” a LLM with specific content relevant to the enterprise or business
  - ◆ This is an expensive process requiring skills and resources
  - ◆ The tuned LLM still has problems with model drift and model cutoff date, requiring frequent retraining to address
  - ◆ Not very suited for handling dynamic data from business operations that change rapidly
- ➔ Inject a context into the generative transformation process
  - ◆ Does not have the problems with drift or model cutoff dates
  - ◆ Can handle a diversity of data types, including rapidly changing business data

For most business organizations, context injection is a more practical approach. The challenge is to generate a context that would reliably work with the GPT machine to generate a relevant and useful output. This is addressed by **Context Engineering**. Context injection is at the core of the Retrieval Augmented Generation (RAG) process.

Injecting a context in a RAG flow has additional advantages:

- ➔ The RAG system can be made relatively insulated from model drift since the context can be easily updated as the underlying circumstances change.
- ➔ The context can be easily populated with dynamic data that is generated within the enterprise, ensuring that the RAG process provides a very uptodate information.

Context engineering implements suitable data architectures and algorithms to generate a context that is appropriate for a particular task.

Working definition

Context engineering refers to a systematic process of dynamically orchestrating and assembling a set of informational contextual components that are used by an autoregressive LLM (aka a GPT) to generate an output sequence.

The contextual components refer to:

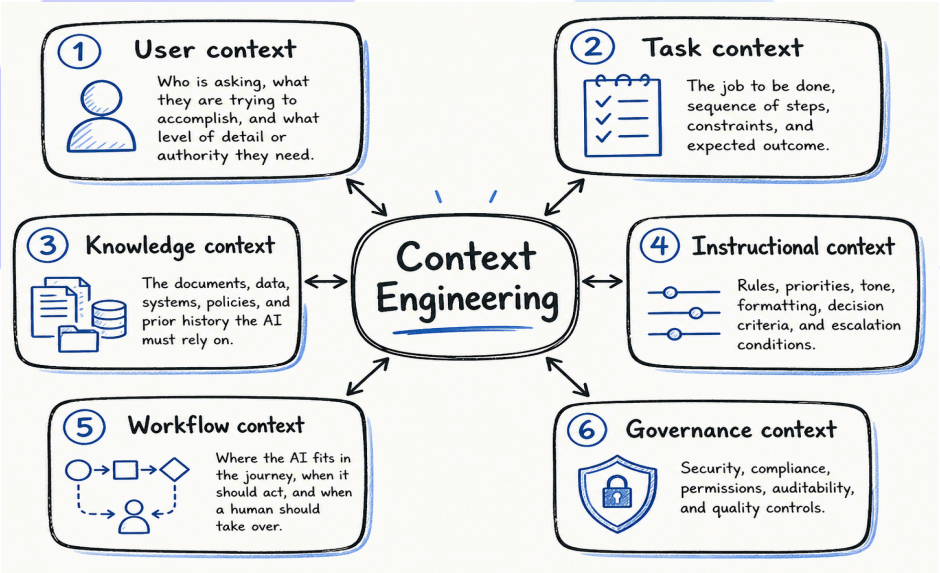
- ➔ The *underlying Knowledge Base* that holds the data used to generate the context
- ➔ The *methods (ELT pipelines) and tools* used to populate the knowledge base
- ➔ The *retrievers* used to extract contextual information that is relevant to a query
- ➔ *LLM Prompt engineering techniques* that elicit precise and relevant contextual information

A common practice is to implement specialized AI agents as *'**context harvesters**'* from the operating environment.

A simpler way to explain it

If prompt engineering is about phrasing the request, Context Engineering is about *harvesting* information that ensures a precise and relevant response.

Context Engineering spans multiple layers:



The engineering challenge for context *harvester*

All engineering activities involve a tradeoff. Engineering a context involves aligning the following competing considerations.

1. *Context Window* - This limits the size of the request that can be made to the LLM. While we would like to add as much context as we would like, we are constrained by the window size.

2. *Performance* - Performance is a key consideration. The performance of a GPT depends on the size of the incoming token count. The performance complexity of modern commercial GPT platforms is O(n), however, performance of many of the open source LLMs can be $O(n^2)$ making them very sensitive to the token count.

3. *Accuracy Metrics* - A large context helps to improve the accuracy of the generated output. Conversely, a shorter context would degrade the accuracy of the output and may be more open to hallucination.

4. *Cost* - Commercial LLMs usually charge by the number of tokens submitted in a query. Injecting a context increases the number of input tokens, and hence the cost per query. The marginal cost must be balanced against the benefits received due to a better response, which may eliminate further follow on queries.

Context Engineering is not just a prompt exercise. It is a design, architecture, engineering and operating model discipline.

3. Why Context Engineering Is Becoming Essential

A review of some of the most successful AI projects in 2025 shows why managing the context is one of the key elements of success:

- ➔ With its "AskCR" product, Consumer Reports vectorized over 90 years of structured and unstructured data into a vector database. It then engineered a context aware retriever that provided a "reasoned" answer to questions with zero hallucinations and high levels of accuracy. This product maintains the reputation of Consumer Reports that have been established over several decades.

- ➔ Starbucks DeepBrew recommends menu items for purchase on its online ordering platform based on previous purchasing habits. Additionally, and perhaps more importantly, it also shows additional options that are likely to appeal to that customer based on the current purchase trends. It considers a wide range of factors to generate a context for its AI query for such additional recommendations that contribute towards additional revenue.

The Key To AI Success

AI systems that focus on improving domain specific high value tasks based on factual, validated contextual data are likely to succeed and generate ROI. Even for such domain specific implementations, it is vital to control the contextual data that is used to generate a response.

- ➔ IBM Watson for Oncology, an investment of billions of dollars promising to revolutionize cancer treatment failed miserably. A key reason for this failure was that the system failed to include actual patient data in the context used for generating responses, sometimes relying only on hypothetical and theoretical content.

- ➔ There are several documented cases where a customer facing chatbot, fed with the wrong contextual information gave an erroneous response, leading to litigation, reputational damage and increased customer service costs. One of the more prominent ones was that with Air Canada wherein their customer service chatbot, fed with incorrect information, promised a refund that was not allowed in the company policy.

The Shift in Business Expectations

Business organizations have quickly moved from using AI chatbots as an intelligent query tool to more sophisticated applications where the AI systems are used to deliver nuanced opinions

based on available facts in the knowledgebase. Further, AI agents are increasingly used to initiate business workflows with or without a human in the loop. Successful implementation of such systems is essential to get ROI from AI based systems in a business environment.

As organizations move from isolated chats to embedded AI systems, the complexity of context grows quickly. Generic prompting is no longer sufficient when AI must:

- answer from enterprise knowledge
- support customers or employees in critical workflows
- work across tools and systems
- respect business policies
- produce repeatable outcomes
- integrate into operational processes

Leading Indicators

The leading indicators of success for a business AI project help to establish the success factors of the project, set expectations for the stakeholders and can establish if the implementation is on track before full business impact shows up.

Category	Indicators
User adoption	Percentage of target users actively using the AI; Repeat usage rate; Usage frequency per role or team; Feature adoption by workflow
Task coverage	Share of intended use cases where AI is actually being used; Percentage of workflows where AI is embedded vs optional; Number of business scenarios handled successfully
Output quality	Answer relevance; Groundedness to trusted sources; Hallucination/error rate; Accuracy against benchmark tasks; Response consistency for similar inputs
User trust and satisfaction	Satisfaction rating after interaction; Trust score; Perceived usefulness; Percentage of users who rely on the output without redoing the work manually
Handoff effectiveness	Percent of AI outputs requiring little or no editing; Handoff success rate to human teams; Exception/escalation appropriateness; Time saved per interaction in pilot stage

Lagging Indicators

These show whether AI is meeting the previously established success criteria and producing actual business value over time.

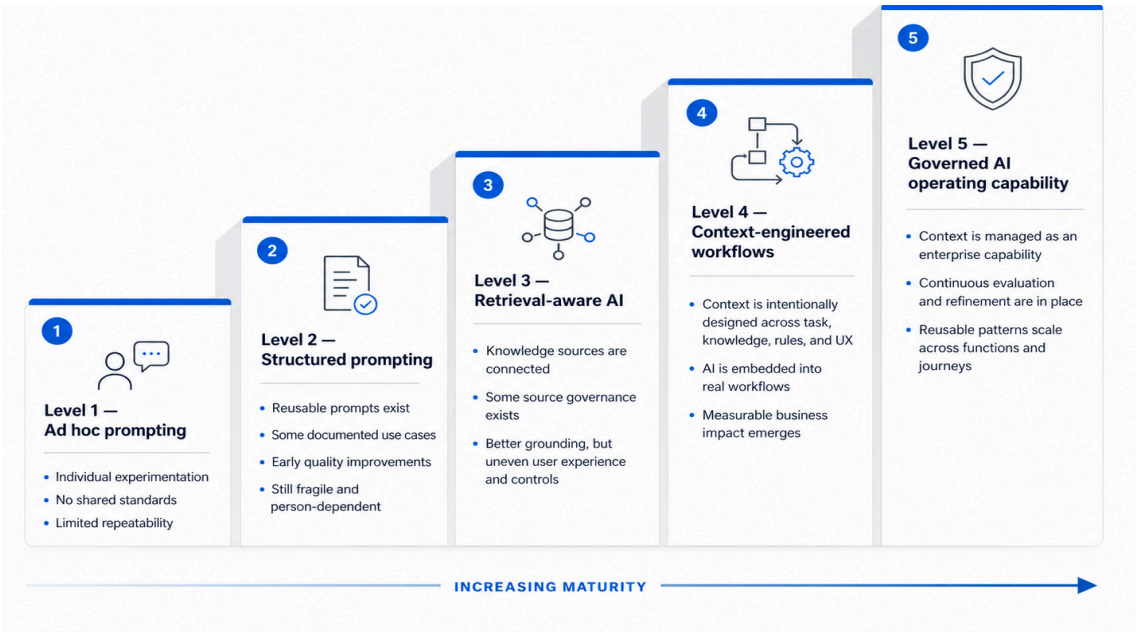
Category	Indicators
Productivity improvement	Reduction in task completion time; Increase in throughput per employee/team; Reduction in manual effort; Faster onboarding or reduced training time
Cost impact	Reduction in operating cost; Lower support cost per case; Reduced contractor/vendor dependence; Infrastructure cost vs value delivered
Quality improvement	Reduction in error rates; Fewer rework cycles; Fewer missed steps in workflows; Improved compliance adherence
Customer impact	CSAT improvement; NPS improvement; First-contact resolution improvement; Reduced average handle time; Lower churn or complaint rate
Revenue impact	Conversion rate improvement; Higher average order value; Better lead qualification; Improved sales cycle speed; Uplift in upsell/cross-sell
Business process performance	Cycle time reduction; SLA attainment improvement; Fewer bottlenecks; Improved forecast accuracy or planning quality
Risk reduction	Fewer policy violations; Lower fraud/loss incidents; Reduced knowledge loss dependency on specific employees; Improved audit outcomes
ROI	Total value realized vs implementation cost; Payback period; Benefit realized per use case; Value per active user or per transaction

AXIONIX point of view

The winners in enterprise AI will not simply have access to strong models. They will have stronger context systems. We firmly believe that a weaker model with superior context-engineering will produce better results than the strongest model with poorly engineered context.

4. AI Context Maturity Model

The AI Context Maturity Model provides a simple way to assess how far an organization has progressed from ad hoc experimentation to a governed, scalable AI operating capability. It highlights that sustainable AI value does not come from model access alone, but from how well context is designed across knowledge, instructions, workflows, controls, and continuous improvement. This model helps leaders identify current gaps, align priorities, and define the next steps needed to turn isolated AI efforts into reliable business capability.



Questions to ask your own team

- Are your AI outputs consistently grounded in trusted sources?
- Do you know which context variables most affect quality?
- Are business rules and escalation logic encoded or assumed?
- Is the user experience designed around AI behavior, or vice versa?
- Can you explain why the AI responded the way it did?

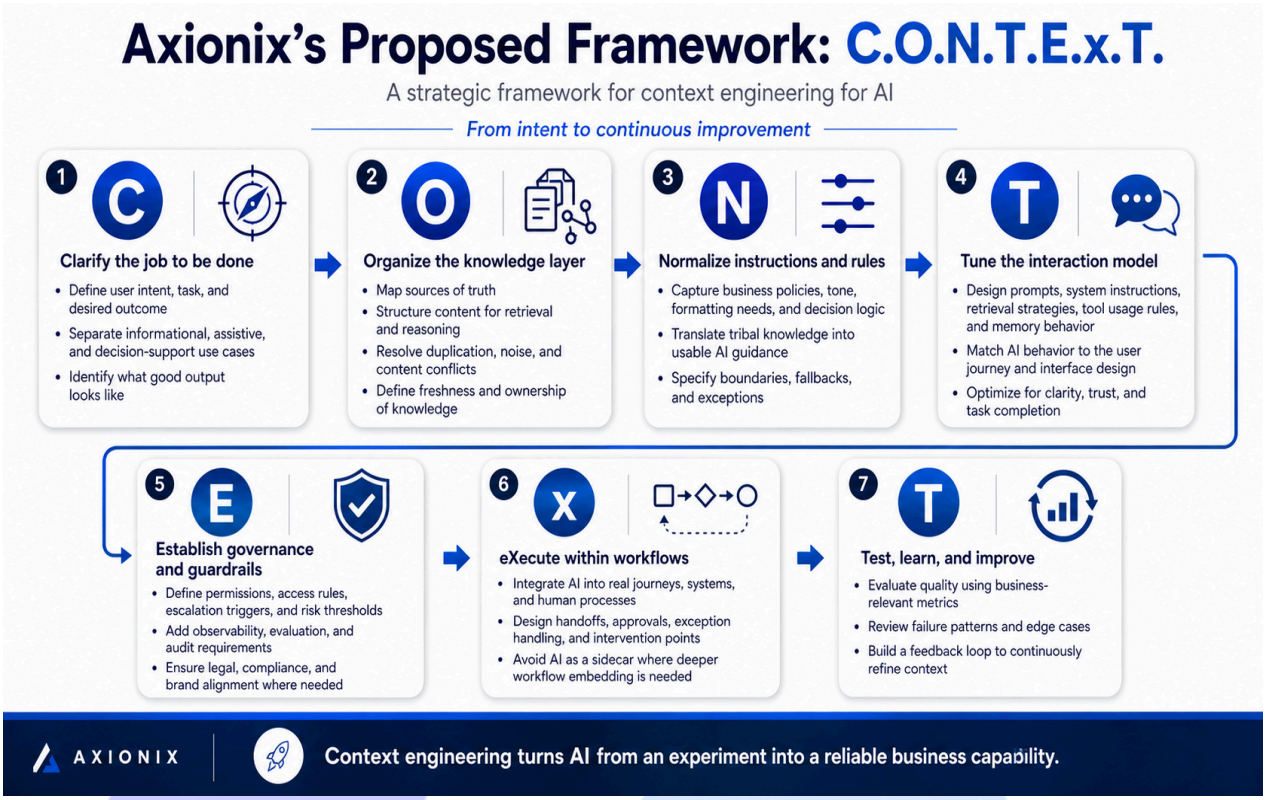
Risks of Ignoring Context Engineering

Organizations that skip Context Engineering may experience:

- Higher hallucination and error rates
- Low trust from employees and customers
- Increased compliance and brand risk
- Poor adoption despite high AI investment
- Pilot fatigue without production value
- Expensive rework after failed deployments

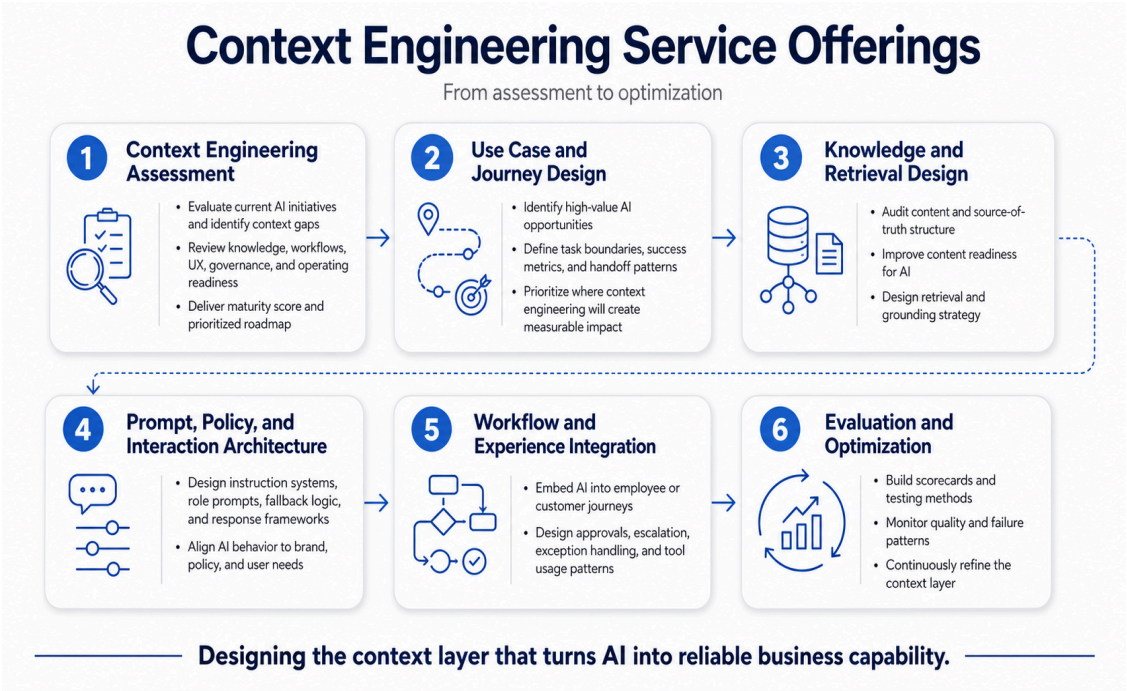
5. C.O.N.T.E.x.T. - Axionix's Proposed Context Engineering Framework

Poor context is one of the most expensive invisible problems in enterprise AI. Hence, at Axionix, we came up with a framework to rapidly mature context engineering.



6. How Axionix Can Help

Axionix helps organizations design the context layer that turns AI from an experiment into a dependable experience.



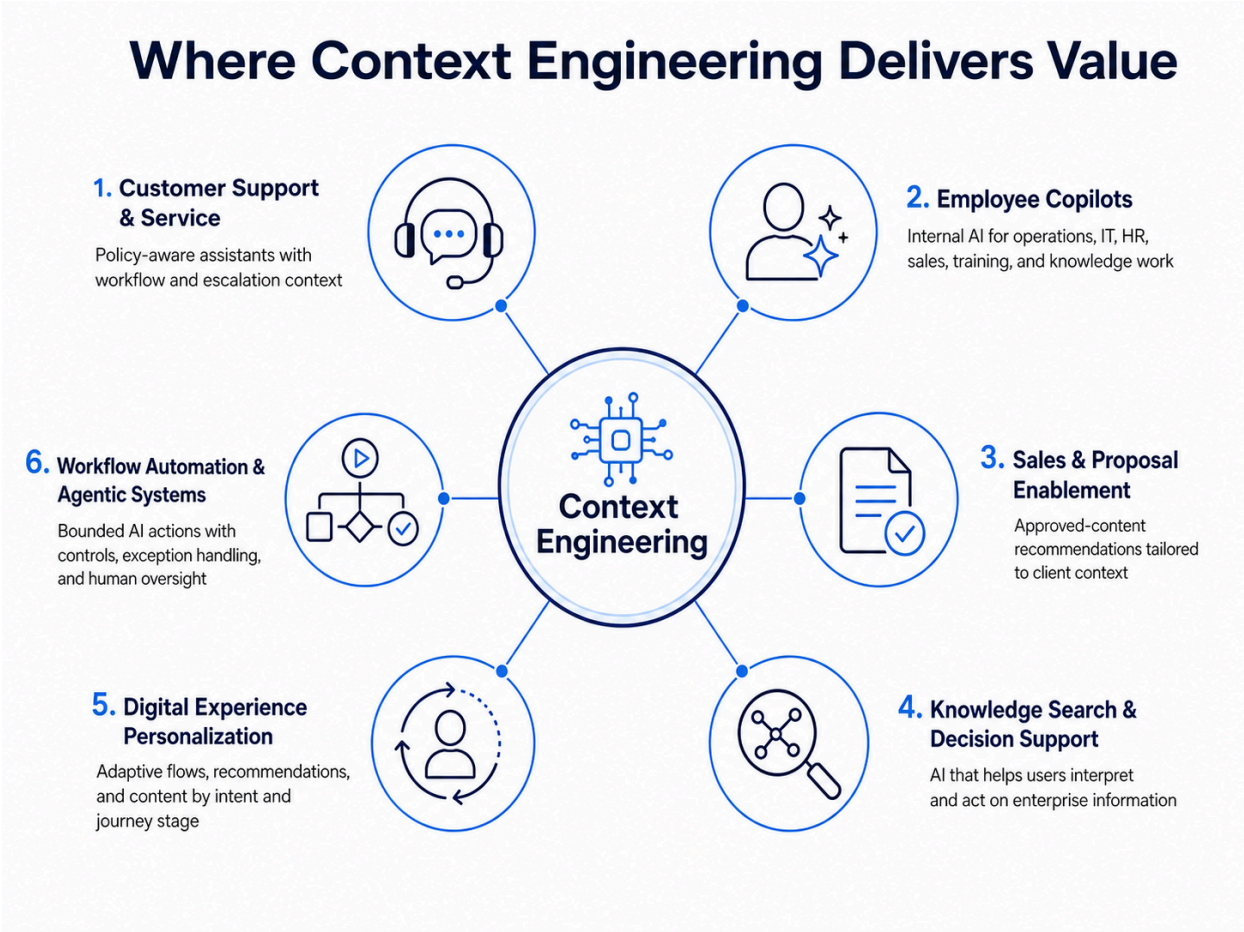
Tangible Outputs of a Context Engineering Engagement

What Clients Walk Away With

A Context Engineering engagement should not end with recommendations alone. It should produce a practical set of strategic, design, governance, and optimization assets that help organizations move from AI experimentation to reliable operating capability.

Assessment and Prioritization ● Context maturity assessment ● Use-case prioritization matrix	**Architecture and Experience Design** ● AI journey maps ● Knowledge source architecture ● Context design blueprint
Behavioral and Governance Controls ● Prompt and policy library ● Guardrail and escalation framework	**Measurement and Continuous Improvement** ● Evaluation scorecard ● Optimization backlog

Context Engineering becomes most valuable in environments where AI must do more than generate plausible responses. Its impact is strongest where systems must operate with the right knowledge, within clear policy and workflow boundaries, and in ways that build trust with users while producing measurable business outcomes. In practice, this makes Context Engineering especially relevant across a set of high-value enterprise domains.



7. Conclusion

As AI Systems become more mature and deeply embedded in business systems, they will inevitably move up the context engineering maturity model. Effective context engineering shall be a key differentiator for success.

The C.O.N.T.E.x.T. framework from Axionix provides a solid foundation and roadmap to build a strong foundation to build AI solutions that deliver business value. It helps to converge strategy, design, knowledge architecture, workflow thinking, and governance to build a successful AI system.

For organizations seeking dependable AI outcomes, Context Engineering is not optional. It is the operating discipline that makes AI work in the real world.