

Building trusted AI in the enterprise

Anthropic's guide to starting, scaling,
and succeeding based on real-world
examples and best practices

The integration of Generative AI (GenAI) has moved from buzzword to bottom line, with early adopters reporting impressive productivity gains and meaningful revenue growth across key business functions. From customer service teams cutting response times in half to marketing teams generating months of content in days, today's proven enterprise success stories show that implementing GenAI isn't just about improving operations—it's about reshaping competitive advantage for years to come.

The business impact is already clear

Early adopters of large language models (LLMs) are already seeing remarkable results:

- Customer support teams are responding **20-35%** faster to inquiries
- Engineering teams are reducing coding time by **15%**
- Content creators are working **30-50%** faster
- Back office operations are **20-50%** more efficient¹

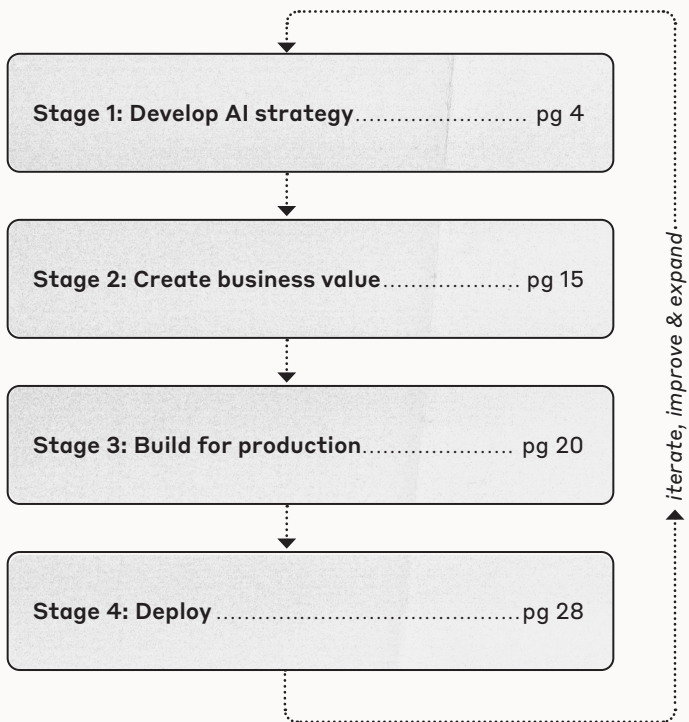
Perhaps the most striking, top performers are attributing over **10%** of their earnings to Generative AI implementations.²

Drawing from thousands of enterprise deployments of Claude, Anthropic's AI assistant, we've found the winning formula is methodical: identify high-impact use cases, build strong foundations, and scale what works.

This guide shares learnings and real-world examples from organizations that have successfully navigated this transformation. Here's the framework we'll be following:

1. Bain & Company [Technology Report 2024](#)

2. McKinsey [State of AI report 2024](#)



Stage 1:

Develop your AI strategy

A successful enterprise AI strategy requires a three-dimensional approach encompassing people, processes, and technology. Each dimension requires specific attention to ensure comprehensive implementation and sustainable adoption.

PEOPLE

Executive alignment and sponsorship

Get executive buy-in by articulating a strategic vision that ties AI initiatives to business outcomes. Clearly articulate the role Generative AI will play in organizational productivity, growth, and success. Plan for ongoing leadership engagement beyond the initial approval and provide realistic guidance around timelines and impact.

Governance and oversight

Governance shouldn't be an afterthought. Establish an AI review board, define ethical guidelines, and create transparent processes for model evaluation and incident response. The goal is to build trust while maintaining momentum.

PROCESS

Pilot selection and design

Avoid the temptation, or external pressure, to tackle the biggest opportunities first. An ideal pilot strikes a balance: significant

enough to demonstrate real value, yet contained enough to deliver results quickly. Consider use cases with enthusiastic business sponsors, clear data availability, and manageable compliance requirements.

When identifying pilot opportunities, think about friction points that could be addressed with AI, and capitalize on inertia within the organization while encouraging rapid iteration and experimentation. That will enable a central foundation from which teams can continue to scale the adoption of AI across processes and tools.

Common use cases include:

Example use cases		
External, revenue-oriented	Internal, cost- and risk-oriented	
Better customer experiences	Increased team productivity & creativity	Optimize business processes
Advanced chatbots: <i>Address customer issues and queries in real-time</i>	Code generation: <i>Rapidly create and optimize programming code</i>	Fraud detection: <i>Proactively identify and mitigate fraudulent activities</i>
Agent assist: <i>Deliver enhanced support levels without increasing headcount</i>	Data analysis: <i>Interpret complex datasets and generate insights</i>	Document processing: <i>Parse and summarize text to identify key information</i>

Scaling framework

Successful organizations develop clear “graduation criteria” that determine when pilots are ready for broader deployment, and to prevent projects from getting stuck in the “pilot” phase. They create standardized playbooks that capture learnings and accelerate future rollouts, while building reusable components that speed implementation.

Potential graduation criteria:

Performance thresholds

- Accuracy metrics
- Speed improvements & latency metrics
- Cost efficiencies

Operational readiness

- System stability
- Support infrastructure
- Team capability

Risk management infrastructure

- Security compliance
- Data protection
- Operational controls

TECHNOLOGY

Technical foundation

Evaluate your existing infrastructure through an AI lens. This means looking beyond basic computing resources to understand data architecture maturity, integration capabilities, and the tooling needed for AI development and deployment.

You will need more than just large volumes of data. Your data should be high-quality, accessible, and compliant with regulatory requirements. Creating and maintaining such data demands sophisticated data governance frameworks, clear ownership structures, and automated processes for continuous data quality management.

Technical implementation roadmap: increasing application complexity

Scaling the scope and complexity of AI projects across your organization requires a growth in technical sophistication. Much like the evolution of web applications from static pages to sophisticated distributed systems, AI implementations follow a similar journey of increasing capability and complexity. Successfully deploying AI in production environments typically follows a natural progression through several distinct levels of technical maturity.

Level 1: Basic implementation - getting started

The journey begins with straightforward implementations that focus on direct interactions between users and the AI model. At this stage, organizations typically focus on:

- Simple chat interactions with clear input/output patterns
- Basic prompt engineering to improve responses
- Direct model responses without complex processing
- Single-turn conversations without extensive context

This level is ideal for initial proof-of-concept projects and building team familiarity with AI capabilities. Think of it as similar to creating static web pages – it's a crucial first step that builds foundational understanding.

Level 2: Intermediate implementation - adding intelligence

As organizations gain confidence and experience, they typically move to more sophisticated implementations that enhance the AI's capabilities:

- Structured prompts and templates for consistent outputs
- Integration of basic tools for extended functionality
- Knowledge retrieval systems (RAG) for improved accuracy
- Multi-turn conversations with context management
- Basic workflow integration

This stage is where many organizations begin seeing significant business value, as the AI system can handle more complex tasks and integrate with existing business processes.

Level 3: Advanced implementation - building AI agents

The most sophisticated implementations transform AI from a simple query-response tool into an intelligent agent that can execute complex tasks autonomously:

- Multiple tool integration for diverse capabilities
- Complex multi-step workflows
- Agent-based systems with decision-making ability
- Advanced memory and context management
- Sophisticated error handling and self-correction

UNDERSTANDING TOOLS AND AGENTS

As organizations progress in their AI journey, two key concepts become increasingly important: tools and agents. These capabilities represent the frontier of enterprise AI implementation, enabling systems that can not only understand requests but take action to fulfill them.

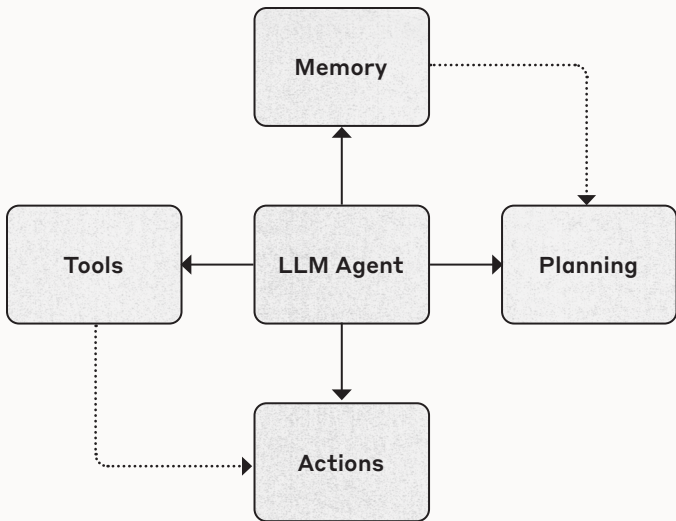
Tools

Tool use (sometimes called function calling) represents a fundamental leap in AI capability. Rather than simply responding with text, AI models can interact with external systems and functions to accomplish specific tasks. Tools are defined functions that the AI can call to perform specific actions, enabling AI to access real-time data, allowing direct integration across systems, and supporting complex business processes.

Agents

An LLM Agent is a system that combines a large language model with the ability to take actions in the real world or digital environments. Key components typically include:

1. The base LLM for reasoning and planning
2. Tools/functions the agent can use (like web searches, APIs, or computer access)
3. A decision-making framework to choose appropriate actions
4. Memory systems to maintain context
5. Goal-oriented behavior to complete tasks



Common examples include agents that can browse the web, interact with APIs, or operate computer systems to accomplish tasks. Anthropic will continue to evolve Claude with ongoing research and training to make Claude even more reliable, safe, and effective as an agentic assistant.

SECURITY AND COMPLIANCE

You'll need a comprehensive security framework that addresses everything from data privacy to model security, while ensuring all systems meet regulatory requirements.

Key security & compliance components:

Data protection

- Encryption standards
- Access controls
- Privacy measures

Compliance management

- Regulatory requirements
- Industry standards
- Internal policies

Monitoring & auditing

- Activity logging
- Performance tracking
- Compliance reporting

“At WPP, we are constantly seeking ways to push the boundaries of creativity and deliver exceptional results for our clients. By working with world-leading technology companies like AWS and Anthropic, we are harnessing the power of cutting-edge AI to enhance our processes, empower our people and drive innovation.”

Stephan Pretorius, Chief Technology Officer at WPP

PLANNING AND FUTURE- PROOFING YOUR AI IMPLEMENTATION

Project implementation roadmap: Building and scaling with success

Successfully embedding AI into enterprise operations requires a measured, progressive approach. The most successful enterprise AI programs evolve like compound interest - each phase multiplies the value of previous investments. This journey unfolds across four key phases, each one expanding both technical depth and operational breadth.

Note: Many of the companies we work with are able to significantly accelerate this implementation timeline with the right level of motivation and partnership, condensing this 13 month+ implementation process into a few months, if not weeks. FeatherSnap integrated Claude in Amazon Bedrock with its smart bird feeder in under 90 days, while DoorDash built a voice-operated self-service generative AI contact center solution in only 2 months.

Phase 1: Foundation building (months 1–3)

The initial phase focuses on establishing the fundamental infrastructure and organizational framework necessary for sustainable AI adoption. During these crucial first months, organizations tend to concentrate on three core objectives:

- Establishing a robust governance structure that balances innovation with risk management
- Defining comprehensive technical requirements aligned with business objectives
- Building and empowering a core team with diverse skills and clear mandates

This foundational phase is critical because it prevents the three most common causes of AI project failure: misaligned governance, technical debt, and talent gaps. By establishing clear oversight structures, technical standards, and team capabilities early, organizations avoid the costly practice of retrofitting these essential elements after problems emerge.

Phase 2: Pilot implementation (months 4–6)

With the foundation in place, organizations then move into the experimental phase, where theory meets practice. This phase is characterized by:

- Launching carefully selected initial projects that balance impact with manageable risk
- Developing comprehensive metrics frameworks to measure both technical and business outcomes
- Establishing leadership and organizational trust in the AI outputs and AI-assisted workflows
- Creating systematic feedback loops to capture learnings and enable rapid iteration

Success in this phase depends on maintaining a balance between ambitious goals and practical limitations, allowing teams to learn and adapt quickly.

Phase 3: Strategic scaling (months 7–12)

As pilot projects demonstrate success, organizations enter a critical scaling phase. This period focuses on:

- Expanding successful pilots across different business units and use cases
- Optimizing processes based on early learnings and operational data
- Building internal capability through training and knowledge transfer

This phase requires careful attention to change management and systematic documentation of best practices to enable successful replication of early wins.

Phase 4: Broad enterprise adoption (months 13+)

The final phase focuses on embedding AI capabilities deeply into the organization's operational fabric:

- Refining strategies based on comprehensive data and experience
- Enhancing automation to improve efficiency and reduce operational overhead
- Scaling successful patterns across the enterprise while maintaining quality and control
- Identifying additional use cases across departments, with new pilot implementations

“We’ve reduced the time taken to create Clinical Study Reports from 12 weeks to 10 minutes, with higher quality outputs and a fraction of the team. In terms of value, each day sooner a medicine gets to market can add around \$15 million in revenue to the company.”

Waheed Jowiya, Digitalisation Strategy Lead at Novo Nordisk

Stage 2:

Create business value

In our experience the highest performing AI teams choose the optimal use case to demonstrate value quickly and scale their AI strategy from there. When you've identified your use case, and success metrics to measure its success, you'll need to select the right partner and LLM to ensure you deliver on its potential.

IDENTIFY USE CASES WITH POTENTIAL FOR PILOT SUCCESS

The most suitable project to start with in your business should have the following characteristics:

Well suited to LLM capabilities. It should leverage core LLM strengths – such as processing unstructured data, content classification, or format transformation.

Meaningful and measurable success metrics. Don't just focus on technical performance. Look for use cases which directly impact key business indicators e.g. reduced processing time, increased throughput, or improved accuracy.

Clear return on investment. Demonstrating concrete ROI builds organizational confidence and supports broader adoption.

Business critical, but low security risk. Your first use case shouldn't carry extreme operational or security risks. This allows you to establish governance frameworks and build institutional knowledge without putting critical operations at risk.

Abundant data. Do you have enough data to support the use case, and is it in the format needed, with permission to use it?

Minimal disruption to existing processes. Consider parallel deployment—running AI-enhanced processes alongside existing workflows until performance and reliability are proven.

Scalable and duplicable. The knowledge and processes developed will become valuable assets as you scale your AI initiatives.

DEFINE YOUR SUCCESS CRITERIA

Good success criteria are:

1. **Specific.** Rather than pursuing broad objectives like ‘improved performance,’ leading organizations define specific, actionable targets. For instance, they might target ‘95% accuracy in customer inquiry classification’ or ‘reduction in average resolution time from 45 to 30 minutes.’
2. **Measurable.** Use quantitative metrics or well-defined qualitative scales. Consider ways to measure abstract concepts – transforming goals like ‘ethical AI deployment’ into concrete metrics such as ‘less than 0.1% of outputs flagged for bias across 10,000 interactions.’
3. **Aligned with business objectives.** The strongest programs tie AI performance to core business objectives, whether that’s operational efficiency, revenue growth, or customer satisfaction.
4. **Time bound.** Create measurement frameworks that track progress across multiple time horizons, from quick wins to long-term transformation goals.

Success metrics for different use cases

Ticket routing	Content moderation	Customer chatbot	Code generation	Data analysis
↑ Routing accuracy rate	↓ False positive rate	↓ Cost per conversation	↓ Time spent on routine coding	↓ Time to insight
↓ Rerouting rate	↓ False negative rate	↑ Conversation completion rate	↓ Bugs and errors in code	↑ Decision accuracy
↓ Time to resolution	↑ Per-category accuracy	↓ Average time to resolution	↓ Project completion time	↑ Ability to handle larger and more diverse datasets
↓ Queue processing time	↓ User churn	↑ First contact resolution rate	↑ Adherence to coding standards	↑ Customer satisfaction with data-driven products or services
↓ Cost per ticket	↓ Appeal volume	↓ Escalations to human agents	↑ Developer productivity	↓ Routine task elimination
↑ CSAT	↓ Cost per review	↑ Percentage of users engaging with chatbot	↑ Code reuse	↓ Time saved per analysis
↑ Volume handling	↑ Community health	↑ CSAT	↑ Test pass rate	↑ Query complexity handling

Working with AWS and Anthropic to apply generative AI and machine learning techniques to its research, Pfizer:

- Cut time from prototype to minimal viable product from 3+ months to **6 weeks**
- Saved **16,000** hours of search time annually
- Reduced infrastructure costs by **55%**

“We made the decision to go with an Anthropic model within AWS because we like the safety that Anthropic provides with their model. If you think about over the course of a year, about 300,000 documents come out and we would review about 20% of those manually through the legal editing process. We processed 300,000 summaries in an automated way in about 6 weeks.”

Jeff Reihl, Global CTO at LexisNexis Legal & Professional

CHOOSING A MODEL

Claude is a family of state-of-the-art large language models developed by Anthropic. Choosing the right Claude model depends on your specific use case and requirements. Here are some key factors to consider:

1. **Balance of capabilities:** Consider the trade-offs between intelligence, speed, and cost for your particular needs.
2. **Task complexity:** For a balance of intelligence and costs Claude 3.5 Sonnet may be most suitable, while Claude 3 Opus excels at the most complex tasks. For simpler tasks or

high-throughput scenarios, Claude 3.5 Haiku could be more appropriate.

3. **Response time:** If near-instant responsiveness is crucial, Claude 3.5 Haiku might be the best choice.
4. **Cost considerations:** Cost is generally a trade off against capabilities - the more complex the task, the more you'll need to spend on a model to ensure consistent high performance.
5. **Context window:** All Claude 3 (including 3.5 models) offer a 200K token context window, which may be important for processing long documents.

“Across all our use cases—from code generation to technical discussions—we’re seeing 5-10% improvements in accuracy, comprehensiveness, and readability over the previous version of Claude 3.5 Sonnet. What’s truly impressive is the new careful reasoning comes without the usual latency hit we see in other models. This combination of improved performance and efficient processing makes it ideal for powering multi-step, agentic software development.”

Taylor McCaslin, Group Manager, Product - Data Science
AI/ML, Gitlab

Stage 3:

Build for production

Now that you've identified the best use case and settled on a model it's time to build something real. Moving to production means thinking through all the details that will help your Claude implementation run reliably and scale smoothly. Whether you're building customer-facing features or internal tools, let's look at what it takes to get your use case production-ready.

PROMPT ENGINEERING

You'll need to build a strong prompt for your test case. Here's how we'd structure it to ensure you get the most out of the model:

- 1 Task + role content
- 2 Background data, documents & images
- 3 Detailed task description & rules
- 4 Conversation history (if applicable) or user input
- 5 Immediate task description or request
- 6 Output formatting
- 7 Pre-filling the response (if required)

Here's what our prompt might look like if we wanted Claude to help us classify customer support tickets:

System	<p>1 The assistant will be acting as a customer support ticket classification system.</p> <p>The task is to classify the ticket according to the rules.</p>
User	<p>2 You will classify a customer support ticket into one of the following categories: <code><categories>{{categories_list}}</categories></code></p> <p>3 Here are some important rules for the classification system: <code><rules>{rules}</rules></code></p> <p>4 Here is the support ticket that you need to classify: <code><ticket>{{ticket}}</ticket></code></p> <p>5 You should respond with the correct classification for the ticket in the requested format</p> <p>6 Put your response in the following format: <code><response></code> <code><category>Your classification choice goes here</category></code> <code></response></code></p>
Assistant (prefill)	<p>7 <code><response><category></code></p>

If you find yourself stuck at the “blank page problem,” we have tools on the [Anthropic Console](#) to help generate strong starting prompts or even improve prompts you already have or want to carry over from other models.

We find that teams tend to give up on prompting quicker than they should and immediately pivot instead to other perceived solutions such as fine-tuning. However, just a few hours of prompt engineering can very often fix their issue without going down the costly path of fine-tuning a bespoke model that incurs extra costs to train and maintain.

We recommend that you invest in prompt engineering as a key skill, as prompting can often improve a model’s capabilities by a large margin, and more quickly and adaptably than other options such as fine-tuning.

Anthropic has extensive resources available to support you with prompt engineering. Recommended resources include:

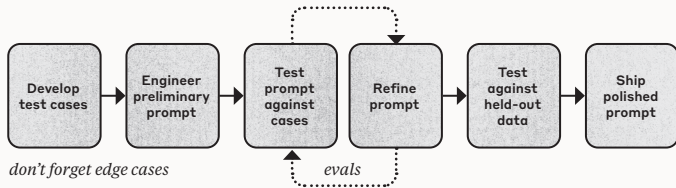
- Our [prompt engineering documentation](#) (tip: you can chat with Claude directly in the Anthropic docs search bar to either learn about prompting or have Claude unblock you on the specific issues you’re encountering)
- Our courses on [core prompting techniques](#) and [production-level prompting](#)

Power tip: You can chat with Claude directly in the Anthropic docs search bar to either learn about prompting or have Claude unblock you on the specific issues you’re encountering.

EVALUATION

During this stage you will evaluate performance of your prompt relative to the success criteria that you've selected. Time spent here will be rewarded with a higher-performing production system, so don't rush through it.

Evaluation is the key to iteration. Without strong evaluation tests, you will have no way to know whether or not the changes you make are having a positive or negative impact, and you will lack infrastructure to confidently assess and adapt to future model upgrades.



More desirable evaluation tests are ones that are:

- Very detailed and specific
- Fully automatable (consider using LLMs as a judge)
- Higher in volume even if lower quality

Less desirable evaluations are:

- Open-ended
- Are not automated and require human judgment
- High quality but at a very low volume

The [Anthropic Console](#) features an Evaluation tool that allows you to test your prompts under various scenarios. The Evaluation tool offers several features to help you refine your prompts:

- **Side-by-side comparison:** Compare the outputs of two or more prompts to quickly see the impact of your changes.
- **Quality grading:** Grade response quality on a 5-point scale to track improvements in response quality per prompt.
- **Prompt versioning:** Create new versions of your prompt and re-run the test suite to quickly iterate and improve results.

You can learn more about building strong evaluations with our [evaluations guide](#) or course on [prompt evaluations](#).

“By optimizing Claude around our industry expertise and specific requirements, we anticipate measurable improvements that deliver high-quality results at even faster speeds. We’ve already seen positive results with Claude 3 Haiku, and fine-tuning will enable us to tailor AI assistance more precisely.”

Joel Hron, Head of AI and Labs, Thomson Reuters

OPTIMIZATION

Once you have run your evaluation tests, you can start to implement optimizations. Here are two improvement strategies that should be part of your toolkit.

1. Few shot examples

This technique involves teaching Claude how to perform the task based on providing a few examples (i.e. more than one) as part of the prompt. Providing examples is one of the most effective ways to improve output quality, especially if you're already following other prompting best practices.

- You can provide examples in context or use RAG for dynamic example insertion
- You should include edge cases in your examples
- Although not essential, you can include an example or two showing how NOT to perform the task
- If you don't have enough examples in your data set, you can also ask Claude to create more for you!

“Sometimes people think that generative AI is very costly, it's very expensive, but if you can use it correctly, like some fine tuning for example, that we do use in Bedrock, it gets cheaper than using traditional models.”

Renan de Padua, Head of Generative AI, iFood

2. Chain of Thought (CoT)

This technique involves giving Claude space to “think out loud”. When faced with complex tasks like research, analysis, or problem-solving, chain of thought (CoT) prompting, encourages Claude to break down problems step-by-step, generating more accurate and nuanced outputs.

Note: Due to how LLMs generate responses, CoT prompting is only effective if the model is given space to think out loud *before* it produces its final answer. Providing a rationale after it has already given its answer generally does not improve its response over baseline.

Benefits of letting Claude think

- **Accuracy:** Stepping through problems reduces errors, especially in math, logic, analysis, or generally complex tasks.
- **Coherence:** Structured thinking leads to more cohesive, well-organized responses.
- **Debugging:** Seeing Claude’s thought process helps you pinpoint where prompts may be unclear. You have a more ‘interpretable’ answer which will allow you to more successfully steer the model over time.

Potential downsides

- Increased output length may impact latency.
- Not all tasks require in-depth thinking. Use CoT judiciously to ensure the right balance of performance and latency.

Here’s what our ticket routing prompt looks like now after adding few shot examples and chain of thought tags:

System	<p>1 The assistant will be acting as a customer support ticket classification system.</p> <p>The task is to classify the ticket according to the rules.</p>
User	<p>2 You will classify a customer support ticket into one of the following categories: <code><categories>{{categories_list}}</categories></code></p> <p>3 Here are some important rules for the classification system: <code><rules>{rules}</rules></code></p> <p>4 Use the following examples to help you classify the ticket: <code><examples>{{examples_list}}</examples></code></p> <p>5 Here is the support ticket that you need to classify: <code><ticket>{{ticket}}</ticket></code></p> <p>6 You should respond with the correct classification for the ticket in the requested format</p> <p>7 Think about your answer first before you respond in <code><scratchpad></code> tags. Consider all of the information provided and create a concrete argument for your classification.</p> <p>8 Put your response in the following format: <code><response></code> <code> <scratchpad>Your thinking here</scratchpad></code> <code> <category>Your classification choice goes here</category></code> <code></response></code></p>
Assistant (prefill)	<p>9 <code><response><category></code></p>

- 1 Task context
- 2 Background data, documents & images
- 3 Detailed task description & rules
- 4 Examples
- 5 Conversation history or user input
- 6 Immediate task description or request
- 7 Thinking step by step (CoT if applicable)
- 8 Output formatting
- 9 Pre-filled response (if any)

Stage 4: Deployment

Once your application runs smoothly end-to-end, you can deploy to production.

Do

- Progressively roll out your application
- Set up infrastructure for A/B testing
- Design user-friendly ways for human feedback
- Update your offline evaluations based on production data
- Iterate on prompts

Don't

- Replace your previous system right away
- Consider your offline evaluations as static
- Make a decision based on a single evaluation test
- Do this 100% on your own. Experts (like Anthropic) can help!

“Using AWS and Anthropic’s Claude, we’ve built a solution that gives Dashers reliable and simple-to-understand access to the information they need, when they need it. This has cascading positive impacts on our users and the platform as a whole, and we look forward to expanding to new use cases in the future.”

Chaitanya Hari, Voice/Contact Center Product Lead with DoorDash

DEPLOY WITH LLM OPS

LLMOps is the set of practices and principles for operationalizing Large Language Models (LLMs) in production environments. Consider it a subset of Machine Learning Ops (MLOps) but specifically focused on the unique challenges of deploying and managing LLMs, such as their large size, complex training requirements, and higher computational demands.

In a survey of 1,400 C-suite executives by BCG, 62% cited a shortage of talent and skills as their biggest challenge when it comes to implementing their AI strategies.³ Training and change management are key to an effective AI deployment.

Every level of the organization needs different AI competencies: C-suite requires strategic vision to lead initiatives, managers need skills to guide teams through transformation, and frontline workers need practical tool proficiency. Understanding these needs enables targeted training programs.

Here are five of the most important best practices for LLMOps:

1. Robust monitoring and observability

Implementing comprehensive monitoring is foundational to successful LLMOps. This means tracking not just basic metrics like response times and error rates, but also LLM-specific concerns like token usage and output quality. The key is creating a system that gives you visibility into how your LLMs are actually performing in production, allowing you to catch issues before they impact users.

3. BCG [Five Must-Haves for Effective AI Upskilling](#)

2. Systematic prompt management

Like code, prompts need version control, testing, and proper documentation. Create a central repository where teams can collaborate on prompts, track changes, and understand why specific prompts were designed the way they were. Implement a testing framework to validate prompts across different scenarios, and maintain clear documentation about each prompt's purpose and expected behavior.

3. Security and compliance by design

Build in proper access controls, content filtering, and data privacy measures from the start. Establish clear policies about what data can be used with LLMs and how to handle sensitive information. Regular security audits and compliance checks should be part of your routine operations.

4. Scalable infrastructure and cost management

Design your LLM infrastructure with scalability in mind, but balance this with cost efficiency. Implement effective caching strategies, choose the right model sizes for different tasks, and optimize token usage. Monitor and analyze usage patterns to identify opportunities for cost reduction without compromising performance.

5. Continuous quality assurance

This includes regular testing of model outputs, monitoring for hallucinations, and validating responses against your business requirements. Establish feedback loops with end-users and maintain clear processes for addressing quality issues when they arise.

Each of these practices supports the others – for instance, good monitoring helps inform both quality assurance and cost management, while systematic prompt management contributes to better security and quality. The key is implementing them as part of a coherent strategy rather than as isolated initiatives.

“The adoption of GenAI has helped us develop thousands of experiences and itineraries for 80% less cost than it would if we were to have our writers manually curate those. Furthermore, our writers are now freed up to go focus on finding the next in-destination thing and continue to inspire travelers everywhere.”

Chris Whyde, Senior VP of Engineering at Lonely Planet

Anthropic + AWS

Together, Anthropic and AWS provide a superior AI solution for enterprises. Anthropic brings frontier safety research and advanced AI products, while AWS provides expertise in secure, reliable cloud infrastructure. Combined they simplify AI deployment and governance while accelerating innovation for customers.

Choosing to work with Anthropic and build on Claude means you can deliver AI solutions that are not only capable but also reliable, safe, and aligned with human values, and which can be seamlessly integrated with your existing cloud infrastructure.

Reach out to the Anthropic sales team to learn more about how we can partner with you on a successful AI strategy.

“AI frees up hours each week for employees across various roles. It’s not just about doing more of the same work; it’s about exploring new frontiers. It’s about unlocking new value and possibilities for our employees. Our vision is two-fold: AI support for every customer and employee.”

Varsha Mahadevan, Senior Engineering Manager at Coinbase

APPENDIX

Essential AI Terminology

Tokens: The fundamental units of text that language models process, typically representing word fragments or individual characters (averaging about 4 characters per token in English).

Sampling: The probabilistic process by which an AI model selects its next output token from a distribution of possible choices, with parameters like temperature controlling the randomness of selections.

Pretraining: The initial phase of AI model development where the model learns general language understanding and capabilities from massive datasets before any specialized training occurs.

Fine-tuning: The process of further training a pre-trained model on specific datasets to enhance its performance for particular tasks or domains.

Supervised Learning: A training approach where the model learns from labeled examples, systematically mapping inputs to their correct outputs based on human-provided training data.

Preference Model: A component of modern AI systems that ranks potential outputs based on desired characteristics, helping align the model's behavior with intended outcomes.

Reinforcement Learning: A training method where the model learns optimal behavior through a system of rewards and penalties, improving its performance through trial and error.

ANTHROPIC