PolyAPI

# Top 6 Considerations for Adopting AI: A First Hand Perspective

PolyAPI

## Abstract

Adopting AI presents a dual challenge: while technical implementation becomes more accessible, strategic navigation grows increasingly complex. The rapid proliferation of models, tools, and frameworks has elevated market expectations, creating steep entry barriers for new adopters. Companies must build expertise through iterative learning, as skipping steps leads to costly missteps. Early adopters hold an advantage, leveraging contextual reasoning and simple, fixed-context tasks to maximize general-purpose AI. Strategies for AI adoption emphasize starting small, prioritizing achievable use cases, and iterating rapidly. By relying on pre-built models and advancing to custom solutions, organizations balance cost, performance, and control while scaling AI effectively.

## About the Author

Before founding PolyAPI in 2023, Darko Vukovic amassed over 16 years of experience in API and integration technologies, holding pivotal product leadership roles at industry-leading companies, including Oracle, Google Apigee, MuleSoft, and Okta.

At Oracle, Darko served as the Director of Product Management Integrations & APIs, where he led the product management team focused on the integration and APIs of OPERA Cloud. In this role, he was instrumental in pioneering the Oracle Hospitality Integration Platform (OHIP), a cloud-native solution that centralizes and streamlines integration capabilities, providing partners with unimpeded access to data and functionality within OPERA Cloud Property Management.

In 2023, Darko founded PolyAPI, an AI-powered platform designed to streamline API discovery and integration for developers. As the CEO, he leads a team of veterans from MuleSoft, Apigee, Oracle, Disney, and Riot Games, all dedicated to enhancing enterprise development productivity.

Darko's extensive experience and leadership in the API and integration domains have positioned him as a thought leader in the industry. His commitment to innovation continues to drive advancements in how enterprises leverage technology for digital transformation.

PolyAPI

# Avoiding Missteps in AI Adoption

Adopting AI is both easier and harder than ever before. Technically, it's getting easier. But strategically, it's becoming far more complex. The crux of this challenge lies in the growing paradigm of choice and rising expectations. As time passes, the number of AI models, hosting options, tools, frameworks, and advanced use cases led by market leaders continues to multiply. Expectations for AI's performance, reliability, and impact have soared, making it harder for new entrants to meet market demands. This dynamic creates an ever-widening gap between the "Do's" and the "Do-Nots."

Take PolyAPI's journey as an example. We began incorporating AI into our product in earnest in March 2023. Since then, we've made significant strides forward. If a new entrant were to step into our space today, they'd likely start with the same basic use case we supported nearly two years ago. But that initial use case wouldn't evoke the "Wow" factor it did back then. The market's perception of AI has shifted. Expectations have evolved, and what was once remarkable has now become table stakes.

This creates a serious challenge for new entrants. It's natural to want to skip ahead—to meet the current market's elevated expectations. But skipping steps isn't possible. The road to AI competence is paved with layers of learnings, failed iterations, and technical complexities that can't be bypassed. Each advancement builds on prior experience. And while others are playing catch-up, companies like ours aren't standing still. We're moving forward, making it even harder for late entrants to close the gap.

Unlike UIs or APIs, which can be visually inspected and emulated, AI's inner workings are largely hidden. You can see the result, but you can't easily deconstruct how it's achieved. This makes AI significantly harder to copy. Early adopters are uniquely positioned to establish a lasting advantage, and in AI, the "first-mover advantage" is more pronounced than with most prior technologies.

There's also a psychological factor at play. Today's market is flush with "AI experts," pre-built models, and competing technologies, a mix that fosters analysis paralysis. New entrants are bogged down by too many options, unable to move quickly or decisively. Meanwhile, customers' expectations have soared, and the room to make mistakes has largely evaporated. Users won't tolerate the same glitches or missteps they might have overlooked two years ago. AI that "sort of works" is no longer good enough.

Then there's the external pressure. Marketing and executive teams increasingly see AI as a branding asset, a shiny, promotional talking point. This can warp decision-making. Instead of focusing on strategic, long-term use cases, companies feel pressured to show

“AI capabilities” to signal relevance. This approach often prioritizes surface-level appeal over meaningful impact, further disadvantageing late entrants.

So, here’s my first piece of pragmatic advice: **Prioritize action and experience over perfection and scale**. Don’t wait for the “perfect” model, framework, or vendor. The landscape will continue to shift, and perfection will remain elusive. Instead, start with a narrow, achievable use case, learn as you go, and iterate rapidly. Early action compounds into experience, and that experience is the most uncopyable advantage you can build in AI.

In this article, I will walk you through our journey, and I will, to the best of my ability, put myself in your shoes today and provide advice on what I would do now if I were just getting started. With expectations higher than ever and the room to make mistakes largely evaporated, taking early, decisive action has never been more critical.

PolyAPI

## Consideration #1: A Valid Use Case

The first pitfall in AI adoption is selecting a use case for which AI is not well suited. My experience is rooted in applying Generative AI, so my perspective is limited to a specific subset of use cases. The key is to build conviction that the use case you're addressing is appropriate for AI. With experience, I've learned to identify where AI can and cannot be relied upon in our domain. Interestingly, I do not fully understand how the models are trained or operated, and that's okay. You don't need to understand the inner workings of AI to recognize its strengths and limitations.

There are two distinct patterns I can confidently recommend where general-purpose generative AI models perform well:

- **Instruction + Context**
- **Instruction + User Prompt + Context**

These patterns represent context-bound problems. In essence, you supply the AI with the information it needs to operate within your domain—the AI isn't expected to "know" your problem in advance. This "just-in-time" approach allows you to maintain control over the context, significantly reducing complexity. Expecting the AI to have pre-existing knowledge of your domain requires a higher level of complexity and skill, which many teams (ours included) may not possess.

To the end user, however, it appears as though the AI "knows everything," but in reality, the system you've built is responsible for constructing the context. This hybrid approach is powerful. It's almost as effective as training a dedicated model, with the added advantage of enabling model substitution on the fly or supporting multiple models simultaneously. Best of all, it's much simpler to implement.

Another crucial consideration is task granularity. Always ask AI to do **one thing at a time**. You can break larger workflows into smaller steps, using the AI's output as input for subsequent prompts. For example, when PolyAPI helps you locate an API in your enterprise catalog, it's not one AI request, it's three sequential prompts, each round trip adding more context and refining the result.

Here are some concrete examples of how we use AI at PolyAPI today. Notice how they all fit the pattern of providing an instruction with the necessary context to produce a result. These use cases showcase the power of general-purpose models, without requiring model training or complex customization:

- Given an API call (HTTP info) and its Postman config, generate a name, description, and variable descriptions, meeting a specific format and conditions.
- Given an OpenAPI spec, generate descriptions for each operation and argument.
- Given a variable name and value (non-secret), generate a description for this variable.
- Given a user prompt, extract relevant English-equivalent keywords to query our catalog.
- Given a set of functions, score them on a defined scale for their applicability to a specific user prompt.
- Given a prompt and a set of functions, generate an example in a specific programming language that solves the problem, following a particular code structure.
- Given a prompt and relevant documentation, answer the user's question on how to use Poly.

Here are some upcoming AI-driven capabilities we have on our roadmap:

- Given a schema from the catalog and an observed schema at runtime, summarize the change.
- Given code using one function, summarize the runtime impact of switching to another function.
- Given code using one function, rewrite the code to use another function.

You'll notice that all these use cases are **fixed context** problems, where the AI is being asked to perform tasks based on the context you supply. This approach allows general-purpose models to thrive. They're not "experts" in your domain—they're general reasoning engines. By focusing on "contextual reasoning" instead of "knowledge-driven reasoning," you unlock the ability to harness general-purpose models for high-impact results.

This is what I'd call **Level 1 AI adoption**. It's the starting point, and it's more than enough to generate significant value. The complexity of AI applications increases rapidly as you go deeper. If you're tempted to skip this stage and jump straight into more advanced methods, I'd caution you: You're likely jumping in over your head. Start with simple, context-driven prompts and iterate from there. It's a pragmatic approach that's effective, scalable, and sustainable.

PolyAPI

## Consideration #2: Selecting the Right Model

Choosing the appropriate AI model is a pivotal decision that can significantly impact the success of your implementation. Based on our experience, leveraging pre-built, general-purpose models is often the most pragmatic approach during the initial stages of AI adoption. While the allure of fine-tuning or creating custom models can be strong, the practical challenges and costs associated with these efforts often outweigh their benefits, especially early on.

We explored fine-tuning OpenAI models to optimize performance for specific use cases. However, we found that adding well-structured context to a base model consistently produced better results. Fine-tuning requires substantial upfront investment, both in terms of time and resources, with no guaranteed improvement in output. Additionally, the capital expenditure needed to build and train a fine-tuned model rarely justifies the marginal savings it might offer in operation. For example, the model would need to perform significantly better or cost significantly less over time to recoup the initial investment.

To navigate these challenges effectively, my advice is straightforward: start with pre-built models. These models provide sufficient flexibility and power for most initial use cases, allowing you to gain experience, refine your understanding of costs, and evaluate the true needs of your business. Once you've reached a level of maturity where advanced customization is justified, consider licensing fine-tuned or custom models.

When advancing to this stage, it's crucial to collaborate with experts who specialize in model development and fine-tuning. Licensing these models can be more expensive than using base models, but the enhanced performance and tailored results often justify the cost. If you're investing in a model built specifically for your business, retaining intellectual property rights should be a priority. Conversely, if you're using a general-purpose fine-tuned model, be mindful that your competitors may have access to similar capabilities.

Finally, consider the implications of using your data to train models. It's essential to negotiate terms that ensure your data isn't repurposed in ways that could disadvantage your business. For instance, allowing a vendor to use your data to improve their base model might inadvertently benefit your competitors. A safer approach is to permit the vendor to apply the same methodology with other clients' data but restrict the use of your data in training models that are shared externally.

By starting with general-purpose models and incrementally exploring advanced options, you can balance performance, cost, and strategic control. This measured approach not only minimizes risks but also positions your organization for sustainable, scalable AI adoption.

PolyAPI

## Consideration #3: AI Provider, Hosting, and Privacy

Selecting the right AI provider and hosting approach is a critical decision that affects not only performance and cost but also trust and flexibility. Providers offer different hosting options, some models are hosted exclusively as a service, while others allow self-hosting or a hybrid of both. Additionally, multiple providers might support the same model, further complicating the decision-making process.

In our case, we trust OpenAI to host our models, using their services to minimize friction and ensure rapid deployment. For organizations prioritizing privacy, providers like Microsoft have become popular options due to their strong reputation for safeguarding sensitive data. The choice of provider often hinges on balancing operational ease with compliance and trust factors specific to your industry or customer base.

**Privacy Considerations –** Privacy is perhaps the most crucial factor to address when choosing an AI provider. It's imperative to have a clear understanding of how any data you supply, whether as input for model context, fine-tuning, or training is used. Ensure you explicitly review whether your data will be incorporated into training datasets for either the same model or other models. Misaligned expectations on data usage can have significant implications, both in terms of customer trust and competitive positioning.

**Flexibility in Provider and Hosting Choices –** Our approach to this challenge has been to design our platform with flexibility at its core. Our service is configurable, allowing us to support a variety of models and AI providers. This enables us to meet diverse customer requirements. For instance, one customer might use a specific model from OpenAI, another might prefer a model hosted by Microsoft, while a third might choose to deploy their own model entirely. This adaptability allows us to pivot as market conditions, customer needs, or providers' terms of service evolve.

While not all organizations may require this level of flexibility, I strongly recommend designing your architecture to avoid hardcoding reliance on a single provider. A rigid dependency on one provider could leave you vulnerable to changes that disrupt your operations or fail to meet your customers' expectations.

**Looking Ahead –** Building flexibility into your solution ensures you can adapt to unforeseen changes in the market and better align with your customers' evolving priorities. Even if you don't anticipate switching providers or hosting approaches now, maintaining the option to do so is a strategic safeguard. As the AI space continues to mature, agility will be a key advantage, enabling you to remain competitive and responsive to emerging challenges and opportunities.

PolyAPI

# Consideration #4: Multi-Step Orchestration Engine

When implementing context-based workflows with AI, it's crucial to treat the AI as a collection of modular "functions," each designed to perform a singular task. These functions can then be orchestrated in a sequence to accomplish a broader workflow. This approach ensures clarity, improves reliability, and allows for targeted optimization of each step in the process.

In our solution, we use a single model for all operations, but it's entirely possible, and sometimes advantageous, to use multiple models, each tailored to specific functions. For instance, one model could prioritize speed for initial tasks, another could emphasize quality for generating outputs, and a third might optimize for cost. This flexibility enables fine-tuned workflows that balance performance and efficiency.

Take, for example, our AI assistant that generates tailored examples for user prompts by leveraging our catalog of functions. It operates in three distinct steps:

1. **Keyword Extraction**: The AI processes the user's prompt and extracts relevant English-equivalent keywords.
2. **Result Scoring**: Using traditional search operations, the AI evaluates and ranks potential matches, selecting the most relevant results.
3. **Example Generation**: The AI uses the selected functions to generate a specific programming example that addresses the user's prompt.

Each step in this sequence is carefully managed, with explicit handling for failure cases. If a step yields unsatisfactory results, the workflow has mechanisms to identify and address the issue before proceeding. This incremental, step-by-step orchestration not only improves the user experience but also eliminates common pitfalls like hallucinations, situations where the AI generates misleading or incorrect outputs by misinterpreting context.

Attempting to combine these tasks into a single step would introduce significant risks. A single-step approach would overwhelm the AI with too much context, leading to confusion, hallucinations, and poor-quality results. For example, providing the AI with details for thousands of functions from our catalog in one pass would not only exceed the model's context window but also drive up costs and degrade performance. By breaking the process into logical steps, we keep the context window small, minimize processing overhead, and achieve higher accuracy.

**Key Principles for Multi-Step Orchestration**

1. **Incremental Progression**: Build workflows step by step, feeding only sanitized and relevant context into each subsequent step.
2. **Failure Management**: Define conditions for negative cases at every step to ensure failures are detected and managed gracefully.
3. **Context Minimization**: Limit the amount of data passed into each step to optimize costs and performance while avoiding errors caused by exceeding context limits.

By treating AI workflows as a series of modular operations, you can achieve greater control, higher-quality results, and a scalable framework that can adapt to new use cases. This approach is foundational to effective AI implementation and ensures a sustainable, cost-efficient deployment.

PolyAPI

# Consideration #5: Performance and Cost Optimizations

Optimizing for performance and cost is a critical aspect of integrating AI into your workflows. While there are numerous techniques to achieve this, and undoubtedly more to be discovered, the fundamental principle revolves around efficiently managing **tokens**, which directly influence both cost and response speed.

**Key Factors to Consider:**

- **Cost**: This depends on the number of tokens you send to the AI model (input) and the number of tokens it generates in response (output). Minimizing both input and output tokens directly reduces costs.
- **Performance (Response Speed)**: The speed of the AI's response is primarily affected by the number of tokens it generates. To achieve faster responses, aim to keep the output as concise as possible.

## Practical Example of Token Optimization

Consider a scenario where the AI is tasked with selecting a function from a list of 40 options. Each function has a unique identifier (ID), but if those IDs are 30+ character GUIDs, the AI must output a significant volume of tokens to return results. By assigning shorter, two-digit IDs to these functions for the purpose of the AI's scoring, the token count for the output is drastically reduced by approximately 15x.

Moreover, if only functions scoring above a specific threshold are needed, filtering the list beforehand can reduce the result set by another 5–10x. Combined, these optimizations can improve response times by over 30x, cutting several seconds from the AI's output generation time and simultaneously lowering costs.

## Streaming for Perceived Performance

Another powerful optimization technique is streaming results intended for human consumption. For example, when the AI generates code snippets or explanations, the user doesn't need to wait for the entire response to be complete before engaging with it. Instead, the AI can stream the response, allowing the user to start reading or interacting as soon as the first tokens are returned. This approach significantly enhances perceived performance, as the user experiences minimal delay between input and output.

**Key Principles for Performance and Cost Efficiency:**

1. **Minimize Input and Output Tokens**: Only include essential context in your requests and ask for concise, specific responses.
2. **Preprocess and Filter Data**: Reduce the data set the AI needs to process by applying filters or simplifying identifiers beforehand.
3. **Stream Results for Users**: Leverage streaming to improve perceived performance, especially for tasks involving human interaction.

By strategically managing tokens and taking advantage of techniques like streaming, you can achieve substantial reductions in cost and response time. These optimizations not only improve the efficiency of your workflows but also enhance user satisfaction by delivering faster, more cost-effective results.

PolyAPI

# Consideration #6: Promoting Your Service

When your AI-powered service is up and running, it's natural to feel a surge of excitement and a desire to share your achievement with the world. While promotion is essential, how you communicate your accomplishment can make all the difference. To stand out and earn genuine recognition, ensure that the people who built the service play a central role in crafting the content for its promotion.

## Authenticity Over Hype

Avoid falling into the trap of generic AI marketing jargon. Statements like "Our AI service revolutionizes [insert industry] with state-of-the-art intelligence" are overused and often met with skepticism. The current AI landscape is saturated with vague claims, and audiences, especially in technical fields, are seeking substance over spectacle. Instead, focus on detailing:

- **What your service actually does.**
- **How it works and was built.**
- **The value it delivers to your customers and users.**

Providing these details not only differentiates your brand but also establishes credibility and builds trust among your audience.

## The Role of Builders in Promotion

The most compelling and credible content comes from the individuals who designed and implemented the service. They possess the nuanced understanding required to describe the challenges faced, the solutions developed, and the innovative aspects of your approach. Whether it's a blog post, a whitepaper, or a technical webinar, having your developers and architects share their journey will resonate more deeply with both technical and business audiences.

## Value First, Technology Second

While the technical sophistication of your AI service is noteworthy, its real impact lies in the value it brings to your customers and the world. Focus your messaging on:

- **Practical benefits**: How does your service solve real problems or improve processes?
- **Tangible outcomes**: What measurable results have you achieved?

However, don’t shy away from sharing insights into the technology itself. Providing conceptual explanations of how your system works can inspire others and foster respect for your brand within the technical community. Most critical logic and implementation details are hidden from external inference, so sharing how the system functions won’t make it easy for competitors to replicate your success.

## Competition and Transparency

Don’t let fear of competition stifle your transparency. Sharing the conceptual underpinnings of your service does not equate to giving away your intellectual property. Implementing a working AI system involves countless subtleties and complexities that are not easily inferred from high-level descriptions. By being open and accurate about your approach, you can:

- **Build respect** among peers and industry leaders.
- **Establish confidence** in your system’s capabilities.
- **Strengthen your brand** as an innovator in the AI space.

## Final Advice

When promoting your AI service, aim to strike a balance between highlighting its value and sharing the technical

journey behind it. Transparency and authenticity will not only set your service apart but also inspire confidence in your audience. Focus on the problems you've solved and the impact you've created, while celebrating the innovation and effort that brought your vision to life. This approach will ensure your promotional efforts resonate deeply and position your brand as a leader in the AI domain.

## Conclusion

Adopting AI is a journey that demands a delicate balance between technical prowess, strategic foresight, and a commitment to continuous learning. While the challenges of entry are significant, the rewards for those who act decisively and thoughtfully are unparalleled. The considerations outlined in this article—ranging from identifying valid use cases to optimizing performance and costs—are designed to guide you in building a foundation that can scale with your ambitions.

The AI landscape is evolving rapidly, and standing still is not an option. Early movers who prioritize learning by doing, embrace modular and context-driven approaches, and foster transparency in their innovations will not only navigate this complex terrain but also shape its future. Remember, success in AI is less about achieving perfection and more about creating sustained momentum through incremental progress.

By focusing on delivering real value, maintaining flexibility, and sharing your journey authentically, you position your organization not just as a participant in the AI revolution, but as a leader driving it forward. The opportunities are immense—it's time to seize them.

We also would appreciate any feedback about this writing to ensure we are as accurate about this topic as possible, so please do not hesitate to reach out at info@polyapi.io.

Thank you,

Darko Vukovic
CEO & Founder of PolyAPI