THE FOREIGN MIND

Ender's Reframe for AI Leadership

Chapter Sampler for Endorsement Review

By Steve Harlow

About This Sampler

Thank you for considering an endorsement of *The Foreign Mind*. This sampler includes:

1. **Preface:** The book's origin story and how to use it
2. **Chapter 1: The AI Leadership Hypothesis:** The core argument with evidence
3. **Chapter 4: Empathic Strategy:** A sample pillar chapter showing structure and depth

Total reading time: Approximately 45-60 minutes

These selections give you the book's core thesis, writing style, and practical application. The full manuscript includes 19 chapters plus 5 appendices (~90,000 words).

For endorsement inquiries: Steve Harlow | sgharlow@gmail.com

Preface

I didn't set out to write a leadership book.

I set out to understand why some people thrive with AI while others struggle. After watching hundreds of professionals (engineers, executives, consultants, creators) work with AI tools, I noticed something that didn't fit the conventional narrative.

The pattern wasn't technical skill.

Some of the most technically sophisticated people I observed produced mediocre AI results. They knew the tools intimately. They could explain the architecture. They followed best practices religiously. And their outputs were consistently... fine. Competent. Unremarkable.

Meanwhile, others with less technical background were producing exceptional work. They weren't prompt engineering wizards. They couldn't explain transformer architecture. But something about how they approached AI collaboration produced results that stood apart.

The difference, I eventually realized, was leadership.

The exceptional practitioners weren't treating AI as a tool to master. They were treating it as an intelligence to lead, different from human intelligence, but intelligence nonetheless. They brought to AI the same capabilities that had made them effective with people: clear communication of intent, appropriate calibration of trust, honest assessment of results, accountability for outcomes.

This realization led me to a hypothesis that became the foundation of this book: **AI effectiveness is fundamentally a leadership capability, not a technical skill.** The competencies that enable effective human leadership enable effective AI collaboration. Technical training addresses the smallest part of the challenge. Leadership development addresses the leverage point.

But how do you teach leadership for AI collaboration?

The challenge is that AI is genuinely different from any intelligence we've worked with before. It's not human. It's not the fictional AI of movies, neither the benevolent helper nor the existential threat. It's something new: a Foreign Mind that processes information in ways our intuitions don't naturally grasp.

Leading across that kind of difference requires more than techniques. It requires a new way of thinking about collaboration itself.

I found that new way of thinking in an unexpected place: Orson Scott Card's Enderverse.

I first read *Ender's Game* in 1987. Like millions of readers, I was captivated by Ender Wiggin's journey from isolated child to reluctant commander. I reread it over the years, seeing new layers in the leadership dynamics. I read it again as a technology executive, finding guidance for challenges Card never anticipated.

When I began grappling with AI collaboration, Ender's story illuminated the problem in ways no management framework could.

Ender succeeded not through technical mastery of weapons, but through understanding minds different from his own. He communicated intent rather than instructions. He calibrated trust based on demonstrated capability. He learned from every encounter, wins as rigorously as losses. He maintained ethical grounding even when effectiveness would have been easier without it.

These same capabilities, I realized, were what separated exceptional AI practitioners from merely competent ones.

The Enderverse gave me more than examples. Card's concept of the Hierarchy of Foreignness, his framework for categorizing how different other minds are from our own, provided exactly the vocabulary I needed. His exploration of the Empathy Paradox (that true understanding creates both power and obligation) captured the ethical dimension that purely technical approaches miss.

This book uses Ender's journey as a guide to developing AI leadership capability. Not because science fiction is a management manual, but because the challenges Ender faced (understanding foreign minds, communicating across difference, calibrating trust, maintaining humanity while wielding power) are precisely the challenges AI collaboration presents.

How to Use This Book

This book is organized in four parts.

Part I: The Reframe establishes the core thesis. Chapter 1 presents the AI leadership hypothesis with evidence and examples. Chapter 2 introduces the Enderverse as our pedagogical framework. Chapter 3 explores the paradox at the heart of AI mastery: that understanding creates both capability and ethical obligation.

Part II: The Five Pillars develops the core competencies. Two chapters for each pillar: Understanding (how AI thinks), Communication (translating intent), Delegation (calibrating trust), Development (building capability over time), and Accountability (owning outcomes). Each chapter opens with an Ender example, develops practical frameworks, and identifies anti-patterns to avoid.

Part III: Contexts applies the pillars across different settings. Chapter 14 addresses individual mastery. Chapter 15 addresses leading AI-augmented teams. Chapter 16 addresses organizational transformation.

Part IV: Integration addresses the deeper challenges. Chapter 17 explores maintaining identity in an AI age. Chapter 18 develops future-proof capabilities that transfer regardless of how AI evolves. Chapter 19 issues the call to lead: to develop capability deliberately and use it responsibly.

The Appendices provide practical resources: framework references, case studies, assessment instruments, workshop designs, and an anti-pattern field guide.

You can read straight through, or use the book as a reference. Each chapter stands alone while building on what came before. The frameworks are designed for daily use, not just reading.

A Note on Ender

Some readers will know Ender Wiggin's story intimately. Others will be encountering him for the first time. This book is written for both.

I provide enough context that you'll understand each example without having read the novels. But I also encourage you to experience Card's work directly. *Ender's Game* and *Speaker for the Dead* are masterworks that reward multiple readings. What you'll find in them goes far beyond what I can capture here.

I've been careful to represent the Enderverse accurately, but my interpretations are my own. Card wrote about leadership, ethics, and understanding across difference. He didn't write about AI. The applications I draw are extensions of his themes, not summaries of his conclusions.

The Invitation

AI is changing how we work. That much is obvious.

What's less obvious is that this change requires leadership development, not just technical training. The skills that make you effective with AI are the skills that make you effective with any intelligence different from your own. Developing them makes you a better leader in every context, not just with AI.

This book is an invitation to develop those skills deliberately. To approach AI as a Foreign Mind that can be understood, communicated with, and collaborated with effectively. To build capability progressively through challenge and reflection. To maintain your humanity (your ethical grounding, your professional identity, your core capabilities) even as AI becomes central to your work.

The tools will change. They always do. The leadership fundamentals will endure.

Let's develop them together.

Steve Harlow Tucson, Arizona December 2025

PART I: THE REFRAME

Chapter 1: The AI Leadership Hypothesis

The Puzzle

Tom Brennan had been writing code for nineteen years.

He'd architected systems that processed forty million transactions daily. He'd mentored three generations of engineers. When his company adopted AI coding assistants, he approached them the way he approached everything: systematically, precisely, with the rigor that had made him one of the most respected technical minds in the organization.

He read the documentation. All of it. He studied prompt engineering guides, attended the vendor webinars, bookmarked the research papers. He constructed his prompts like he constructed his code: explicit, unambiguous, technically precise.

Precision had always been his advantage. In code reviews, in architecture debates, in technical documentation, precision was what separated the excellent from the merely competent. If a system wasn't responding to precise inputs, the system was the problem. This was an axiom Tom had never had reason to question.

"Generate a Python function that accepts a list of transaction objects, filters for transactions exceeding the threshold parameter, groups by merchant category code, and returns a dictionary with category codes as keys and aggregated totals as values. Use type hints. Handle edge cases for empty lists and null values. Follow PEP 8 conventions."

The AI returned something functional. But not quite right. The error handling was superficial. The aggregation logic missed a nuance in how their system defined "merchant category." Tom refined his prompt. Added more specifications. More constraints. More technical precision.

Forty-five minutes later, he had acceptable code. Code he could have written himself in twenty minutes.

He studied the interaction afterward, looking for where his prompt had been insufficiently specific. There must be a gap, some ambiguity he'd failed to eliminate. He added three more sentences to his template and filed it away for next time.

"The AI doesn't understand context," he told his team the following week. "You have to specify everything explicitly. It's useful for boilerplate, but for anything complex, you're better off doing it yourself."

His team nodded. They trusted Tom. If he said AI assistance was marginally useful at best, that was probably true.

Three floors up, Claire Morrison was preparing for a board presentation.

Claire had spent her career in operations and strategy. She understood systems at a business level: how departments connected, how incentives shaped behavior, how decisions cascaded through organizations. She did not understand, and had never particularly tried to understand, the technical details of software development.

When her company rolled out AI tools, she'd attended the basic orientation and nothing more. She didn't read documentation. She didn't study prompt engineering. She simply started talking to the AI the way she talked to her team.

But before she typed her first word, she paused.

How will this be interpreted by something that doesn't know our company? The thought arrived automatically, the same thought she had before any conversation with someone new to her world. *It doesn't know that our board chair spent fifteen years running supply chain at Amazon. It doesn't know that "Q3 performance" means something specific here, that we bet the quarter on a supplier diversification strategy everyone said was too risky. It doesn't know any of that.*

She started typing, but differently than she would have an hour ago.

"I need to present our Q3 supply chain performance to the board next week. I have the data, but I'm struggling with the narrative. The numbers are good, but they don't capture why they're good—the decisions we made in February that are finally paying off. Our board chair has deep supply chain expertise from her Amazon years, so I can't be superficial, but two other board members are finance-focused and won't follow technical details. Help me think through how to structure this so the board understands the strategic significance, not just the metrics."

What followed wasn't a single prompt-response exchange. It was a conversation.

The AI's first suggestion was generic: a standard "situation, action, results" framework. Claire read it and felt the gap immediately. *That's what I asked for. But it's not what I need.*

"That structure makes sense generically," she typed back, "but it doesn't account for the skepticism we faced in February. Half the leadership team thought supplier diversification was a mistake. The board approved it reluctantly. Part of my job here is to vindicate that decision without making the skeptics feel attacked. How do I thread that needle?"

The AI adjusted. Claire pushed back again when the language felt too corporate. She added context when the AI made assumptions that didn't fit her organization. She noticed when a suggestion was almost right and articulated precisely what would make it fully right.

Ninety minutes later, she had a presentation framework that was sharper than anything she'd developed on her own. More importantly, she had a deeper understanding of her own strategic narrative. The AI hadn't just produced output. It had helped her think.

"I don't know how I worked without this," she told her chief of staff. "It's like having a brilliant collaborator who's available whenever I need them."

Tom Brennan knows more about AI than Claire Morrison ever will. He understands the technical architecture, the token limits, the model behaviors. He approaches AI with precision and expertise.

Claire Morrison approaches AI with nothing more than the way she approaches people.

Claire gets dramatically better results.

Why?

The conventional answer is that Tom needs more training. Better prompts. More practice with the tools. The conventional answer is wrong.

Tom's limitation isn't technical knowledge. It's something else entirely. Something hiding in plain sight.

Claire isn't succeeding despite her lack of technical expertise. She's succeeding because of a different kind of expertise, one she's developed over decades without ever thinking of it as relevant to AI.

The skill that makes Claire effective isn't prompt engineering.

It's leadership.

This isn't the story organizations tell themselves about AI success. The story is that technical people master technical tools. The story is that precision produces results. The story is that more knowledge about AI leads to better outcomes with AI.

The story is wrong. And the cost of believing it is enormous.

The Technical Fallacy

When organizations adopt AI, they almost always make the same mistake.

They treat it as a technology problem.

Consider David Chen, VP of Engineering at a mid-size fintech. When his CEO announced an AI transformation initiative, David took ownership with the systematic rigor that had made him successful.

He launched a comprehensive training program. Engineers completed certification courses in prompt engineering. The team built a prompt library, hundreds of templates for common tasks, refined through iteration, documented with best practices. They tracked adoption metrics religiously: daily active users, prompts per engineer, time-to-completion ratios.

Six months later, the metrics looked impressive. AI usage was up 400%. Engineers were completing certain tasks faster. The prompt library had grown to over a thousand entries.

Productivity gains were harder to measure.

His engineers could produce more code. Whether they were producing better outcomes was a question no one had figured out how to answer. The most sophisticated engineers, the ones David had expected to lead the transformation, remained skeptical. They used AI sporadically, grudgingly, and with persistent frustration.

"The tools are overhyped," one senior architect told him. "They're fine for boilerplate. For anything that requires real thinking, you're better off doing it yourself."

David had heard this assessment before. He'd heard it from Tom Brennan.

The mistake isn't that technical training is useless. The mistake is that technical training addresses the smallest part of the problem.

Consider how most AI training programs are structured. They teach which buttons to press. Which parameters to adjust. How to format prompts for optimal results. They might even cover advanced techniques like chain-of-thought prompting or retrieval-augmented generation.

This is like teaching someone to drive by explaining the mechanics of internal combustion engines.

Technically accurate. Practically inadequate.

The organizations getting exceptional results from AI aren't the ones with the most sophisticated technical training. They're the ones who recognized, early, that AI effectiveness depends on capabilities that have nothing to do with AI.

To understand why technical training produces limited results, consider the AI Capability Stack. Think of it as four layers, each building on the one below.

The foundation is **Access**: can you use the tools at all? Do you have accounts, permissions, basic technical literacy? David's program nailed this layer. Every engineer had access. Every team had licenses. The infrastructure was solid.

Above that sits **Technique**: knowing how to operate the tools effectively. Prompt structure, parameter settings, feature utilization. This is where most training focuses. David's prompt library lived here. His engineers knew the techniques, many of them better than engineers at companies with three times the budget.

Then comes **Judgment**: evaluating AI output critically. Knowing when to trust results, when to verify, when to reject. Identifying hallucinations, biases, limitations. This is where David's program began to thin. There were guidelines, but judgment is hard to train through documentation. Some engineers developed it through experience. Many didn't. Layer 3 sits at the transition point between technical skill and leadership capability. It requires understanding AI's limitations and calibrating trust appropriately, both behaviors that depend on leadership competencies rather than technical knowledge.

At the top sits **Leadership**: engaging AI as a collaborator. Communicating intent clearly, calibrating expectations appropriately, adapting your approach based on results, taking accountability for AI-assisted outcomes. David's program didn't address this layer at all. It wasn't on his radar. Leadership was something for management training, not AI adoption.

The uncomfortable truth the stack reveals:

Organizations spend almost all their AI development resources on Layers 1 and 2. Access and technique. These are the easiest to train, the easiest to measure, and the least predictive of actual effectiveness.

Layer 4, leadership, receives almost no systematic development. It's treated as either irrelevant to AI adoption or too soft to train.

Yet Layer 4 is where the difference is made.

Tom Brennan operates at Layers 1 and 2 at an expert level. His access is unrestricted, his technique is sophisticated. He understands model architectures that most users have never heard of. But he approaches AI as a tool to be commanded, not a collaborator to be led. He specifies outputs rather than communicating intent. He expects compliance rather than building toward trust.

Claire Morrison barely operates at Layer 2. Her technique is basic, her understanding of AI mechanics minimal. She couldn't explain how a large language model works if her career depended on it. But she operates at Layer 4 instinctively. She engages AI the way she engages her team: with clarity, context, adaptability, and accountability.

David Chen, reviewing his program metrics and wondering why the transformation felt incomplete, sensed something was missing. He couldn't name it.

The capability that matters most is the capability organizations develop least.

The Leadership Evidence

I am not claiming that leadership is a helpful metaphor for thinking about AI. I am not claiming that leadership skills are nice to have alongside technical AI skills. The claim is stronger and more specific:

Leadership competencies predict AI effectiveness better than technical aptitude.

This is an empirical observation, not a philosophical position. It emerges from watching hundreds of professionals interact with AI systems across industries, roles, and technical backgrounds. The pattern is consistent enough to be called a law:

The people who get the best results from AI are the people who would get the best results from a capable but unfamiliar human collaborator.

If you can communicate clearly to a new team member, you can communicate clearly to AI. If you can calibrate trust appropriately with a contractor you've just hired, you can calibrate trust appropriately with an AI system. If you can take accountability for outcomes produced by people who report to you, you can take accountability for outcomes produced by AI assistance.

The skills transfer directly. Not by analogy. Directly.

Consider the core competencies that define effective leadership. Each one has a precise and practical application to AI collaboration.

Perspective-taking is the ability to understand how others will interpret your communication. Effective leaders don't just say what they mean—they consider how their words will be received by different audiences, with different contexts and assumptions.

Tom constructed his prompt from his own technical frame. Every specification made perfect sense to him. He never paused to consider how the AI (trained on different data, optimized for different objectives, lacking his nineteen years of context) would interpret his words. His prompt was precise from his perspective. It was ambiguous from the AI's perspective because he never considered the AI's perspective at all.

Claire paused before her first word. *How will this be interpreted by something that doesn't know our company?* She anticipated misunderstanding before it happened. She provided context that she knew would be missing. She didn't assume shared understanding. She built it.

Clear communication is the ability to articulate intent in a way that produces aligned action. This doesn't mean using more words or more specifications. It often means the opposite: finding the essential core of what you're trying to accomplish and expressing it without distortion.

Tom specified outputs. He described, in technical detail, what he wanted the code to do. Claire communicated intent. She explained what she was trying to accomplish, why it mattered, and what constraints she was navigating. Tom told the AI what to produce. Claire helped the AI understand what success would look like.

The difference is not subtle. It is the difference between commanding and leading.

Appropriate delegation is the ability to calibrate how much authority and autonomy to grant based on capability, context, and stakes. Effective leaders neither micromanage nor abandon. They find the right level of oversight for the situation.

Tom constrained everything. Every parameter specified. Every edge case enumerated. No room for the AI to contribute judgment, to surface considerations Tom hadn't anticipated, to find approaches he hadn't prescribed. He delegated execution of a predetermined solution rather than collaboration on a problem.

Claire said "help me think through this." An invitation, not a command. She gave the AI room to contribute—and then engaged with what it contributed, pushing back, refining, redirecting. She delegated appropriately: neither abdicating responsibility nor refusing to let the AI add value.

Trust calibration is the ability to expand or contract trust based on demonstrated performance. Trust isn't binary. It's a dynamic assessment that evolves with evidence.

Tom's trust never evolved. He approached the AI with skepticism, received results that confirmed his skepticism, and concluded that skepticism was warranted. His interaction created no conditions for trust to develop because he never gave the AI opportunity to earn trust.

Claire's trust expanded through the conversation. The AI earned credibility by responding well to her pushback. She extended more latitude as the collaboration proved productive. By the end, she was working with the AI in ways she never would have at the beginning—not because her standards dropped, but because trust had been earned.

Accountability is the willingness to own outcomes even when others contributed to them. Effective leaders don't blame their teams for failures. They take responsibility for results while developing their people's capabilities.

Tom, when his AI interaction produced disappointing results, concluded that "the AI doesn't understand context." The limitation was external. The failure belonged to the tool.

Claire owned the outcome. When the AI's first suggestion was generic, she didn't conclude the AI was limited. She concluded that she hadn't communicated well enough. She refined her input. She took responsibility for the collaboration, not just her half of it.

This transfer isn't accidental. It reflects something fundamental about what AI systems are and how they work.

Modern AI isn't a calculator. It's not a lookup table. It's not even an algorithm in the traditional sense. It's a system that has learned, from vast amounts of human communication, how to engage in human-like collaboration.

This means AI systems are optimized for human leadership behaviors. They respond to clarity, context, appropriate autonomy, and trust calibration because those are the patterns embedded in their training. The human behaviors that work best with AI are the human behaviors that work best with humans—because AI learned from humans interacting with humans.

When you lead AI well, you're not applying a metaphor. You're leveraging how these systems actually work.

The technical specifications matter less than you think. The leadership behaviors matter more than almost anyone recognizes.

The Reframe

If leadership predicts AI effectiveness, everything changes.

Not gradually. Immediately. The implications ripple through how we think about AI development, who we select to lead it, and how individuals should invest their own growth.

The old frame treats AI as a tool to be mastered. Learn the features. Memorize the prompts. Accumulate technical knowledge. Progress is measured by technical sophistication.

The new frame treats AI as a collaborator to be led. Develop your communication. Refine your judgment. Calibrate trust through experience. Progress is measured by the quality of outcomes you can produce with AI partnership.

These are not two perspectives on the same thing. They are fundamentally different approaches that produce fundamentally different results.

Under the old frame, AI training focuses on tool proficiency. Under the new frame, AI development focuses on leadership capability.

This doesn't mean technical knowledge is irrelevant. You need Layer 1 and Layer 2 of the capability stack. You need access and basic technique. But these are prerequisites, not differentiators. They get you to the starting line. They don't win the race.

What wins the race is Layer 4. Leadership. The capability that most organizations ignore entirely.

Under the old frame, AI adoption should be led by technical experts. CTOs, data scientists, engineers with AI specializations. Under the new frame, AI adoption should be led by effective leaders—regardless of their technical background.

This explains a pattern that puzzles many organizations: some of their best AI results come from departments with the least technical sophistication. The marketing team that treats AI as a creative collaborator. The operations group that approaches AI the way they approach process improvement. The sales organization that uses AI like they use every other resource—as something to be leveraged through clear direction and appropriate accountability.

These groups don't succeed despite their lack of technical depth. They succeed because they default to leadership behaviors rather than technical behaviors.

Under the old frame, individual AI development means learning more about AI. Under the new frame, individual AI development means strengthening the leadership competencies that transfer to AI contexts.

If you want to become more effective with AI, don't start with prompt engineering courses. Start with self-awareness about your communication patterns. Examine how you calibrate trust. Reflect on how you take accountability for collaborative outcomes.

The AI skills you need are leadership skills you may already have. Or leadership skills you can develop in contexts that have nothing to do with AI.

I can hear the objection forming.

"This sounds soft. Leadership is fuzzy. I need practical techniques I can apply tomorrow."

I understand the instinct. But the instinct is part of the problem.

Leadership isn't fuzzy. It's precise. The competencies in the previous section aren't metaphors. They're practical behaviors with specific applications to AI contexts. Perspective-taking has concrete techniques. Trust calibration has measurable dimensions. Clear communication can be evaluated and improved.

What makes leadership seem fuzzy is that it can't be reduced to a checklist. It requires judgment, adaptation, and self-awareness. The same is true of effective AI collaboration.

There is no prompt template that will make you effective with AI. There is no magic formula. There are leadership capabilities that, once developed, transfer directly to AI contexts and continue transferring as the technology evolves.

Techniques expire. The prompt patterns that work today may be obsolete in six months. But the ability to communicate intent clearly? The judgment to calibrate trust appropriately? The self-awareness to adapt your approach based on results?

Those endure.

The Path Forward

This book is built on a single hypothesis:

The capabilities that make humans effective at leading other humans are the same capabilities that make humans effective at collaborating with AI.

If this hypothesis is correct (and the evidence strongly suggests it is), then AI development isn't a technical problem. It's a leadership development problem. The organizations that treat it this way will dramatically outperform those that don't.

The remainder of this book develops that hypothesis into a practical framework. I've organized it around what I call the Five Pillars of AI leadership: the core competencies that predict effectiveness across AI contexts and use cases.

Understanding: The ability to accurately model how AI systems process information, without anthropomorphizing or dismissing.

Communication: The ability to translate intent clearly across cognitive difference. Prompt engineering reframed as leadership communication.

Delegation: The ability to calibrate trust and authority appropriately, neither micromanaging nor abdicating.

Development: The ability to build AI collaboration capability over time through progressive challenge and honest assessment.

Accountability: The ability to maintain human responsibility for AI-assisted outcomes, carrying the weight of consequences.

Each pillar represents a distinct leadership capability with specific, trainable behaviors. Together, they form a complete framework for AI leadership.

I didn't invent these pillars. I identified them by watching people succeed and fail with AI across hundreds of contexts. And in searching for a framework to explain why these particular capabilities matter, I found an unexpected guide.

In 1985, Orson Scott Card published a novel about a child genius who becomes humanity's greatest military commander. *Ender's Game* has since become required reading at military academies and is a stalwart on the Marine Corps Commandant's Professional Reading List. Captain John F. Schmitt, who authored the foundational Marine Corps doctrine on maneuver warfare, used the novel to teach leadership principles at Marine Corps University in Quantico.

Why would military strategists study a science fiction novel about a child?

Because Ender Wiggin's effectiveness doesn't come from intelligence alone—the Battle School is full of brilliant children. It comes from a specific set of capabilities that allow him to understand opponents who think nothing like him, communicate under extreme pressure, calibrate trust with allies he barely knows, adapt to situations no one anticipated, and take accountability for consequences that break him.

These capabilities map precisely to AI leadership. Not because Card anticipated AI, but because he identified fundamental principles of effective collaboration with unfamiliar intelligences. The Formics—the alien enemy Ender must defeat—think so differently from humans that communication seems impossible. Yet Ender finds a way. The lessons he learns illuminate exactly what Claire does instinctively and what Tom has never learned.

Chapter 2 will develop this framework fully. For now, note simply that the leadership capabilities you'll develop in this book aren't abstract theories. They're battle-tested principles with a lineage from military strategy to business transformation to AI collaboration.

The Tool Technician

An anti-pattern is a common approach that seems logical but produces consistently poor results. Anti-patterns persist because they feel correct. They match our intuitions. They're what a reasonable person would do—right up until they fail.

The first anti-pattern is the one we've spent this chapter examining. I call it **The Tool Technician**.

The Tool Technician believes AI effectiveness is fundamentally about knowing the right prompts, the right settings, the right technical configurations. They invest heavily in technical training. They collect prompt templates. They study model architectures and token economics.

The approach feels rigorous. It feels like taking AI seriously. It feels, in a world of hype and hand-waving, like the mature response. Technical people solving technical problems with technical solutions.

And it produces limited results.

The Tool Technician plateaus quickly. They plateau because technical knowledge addresses the smallest part of the capability stack. They're optimizing Layer 2 while ignoring Layer 4. They're learning to use the tool more precisely while never learning to lead the collaboration more effectively.

Tom Brennan is a Tool Technician. His technical sophistication is genuine. His dedication is admirable. His results are mediocre. The three facts are connected in ways he doesn't yet see.

You'll meet more anti-patterns as we proceed. Each represents a seductive trap—an approach that feels right but produces limited results. Learning to recognize them in yourself is the first step toward escaping them.

The Core Insight

If you take nothing else from this chapter, take this:

AI is not a technical problem. It is a leadership problem. The skills that make you effective with AI are the skills that make you effective with people. If you want to improve with AI, improve as a leader.

This is counterintuitive. It defies how organizations are investing, how training programs are designed, how individuals are developing. It suggests that most AI development resources are being deployed in the wrong direction.

That's exactly what the evidence shows.

The organizations that recognize this early will develop capabilities their competitors can't match—because their competitors are still optimizing for the wrong variables. The individuals who recognize this will differentiate themselves in ways that persist as technology changes—because their development is rooted in fundamentals, not features.

The tools will change. They always do. The prompts that work today will be obsolete tomorrow. The interfaces will evolve. The capabilities will expand.

The leadership fundamentals will endure.

If you want to become effective with AI, develop your leadership capabilities. The tools will change. The leadership fundamentals will endure.

Reflection Questions

Don't rush past these questions. The value of this book depends on your willingness to examine your own patterns, not just absorb new information.

1. Think of your most effective AI interaction—a time when AI genuinely enhanced your work. What leadership behaviors were you exhibiting? Were you communicating intent clearly? Calibrating trust appropriately? Taking accountability for the outcome?

2. Think of your least effective AI interaction—a time when AI frustrated you or produced disappointing results. What leadership competency was missing? In retrospect, how might a different approach have produced different results?

3. Who in your organization is most effective with AI? Not who talks about AI the most, or who has the most technical knowledge, but who actually produces remarkable results with AI assistance. What leadership qualities do they demonstrate? How do they approach AI differently than others?

4. Where in your current practice might you be a Tool Technician? Look honestly at your AI use. Are there places where you're optimizing prompts and parameters rather than examining how you're leading the collaboration?

Next: Chapter 2, The Ender Framework

PART II: THE FIVE PILLARS

Chapter 4: Empathic Strategy: Modeling How AI Thinks

The Gift That Precedes Victory

Before Ender ever entered a Battle Room match, he had already won.

Not through confidence or luck. Through understanding. While other commanders studied formations and practiced maneuvers, Ender studied opponents. He watched how they moved, how they communicated, how they responded to pressure. He built mental models of their psychology so complete that their behavior became predictable.

His first commander at Battle School, Bonzo Madrid, was easy to model. Rigid honor code. Status obsession. Predictable escalation when threatened. Ender knew within days exactly how Bonzo would respond to any situation—not because Bonzo was simple, but because Ender had done the work to understand how Bonzo's mind operated.

By the time their final confrontation came, Ender knew precisely what Bonzo would do. The understanding didn't make the outcome less tragic. But it made it inevitable.

Ender's tactical genius wasn't speed or creativity. It was empathy deployed strategically. He could inhabit his opponents' perspectives so completely that he saw battles from their viewpoint before they did. He anticipated not just their moves, but their reasoning.

The same capability separates AI leaders from AI users.

Claire Morrison, from Chapter 1, paused before her first prompt. *How will this be interpreted by something that doesn't know our company?* That pause, that moment of modeling how the AI would process her words, produced results that others couldn't match.

Tom Brennan never paused. He assumed the AI would interpret his technical precision the way a human expert would. His prompts were clear from his perspective. They were ambiguous from the AI's perspective because he never considered the AI's perspective at all.

Understanding precedes effectiveness.

The Projection Trap

The most common failure in AI collaboration is invisible to those who commit it.

Jenn was a product manager known for her clarity. Her PRDs were legendary—detailed, precise, comprehensive. Every stakeholder understood exactly what she meant because she had spent years learning to communicate with engineers, designers, executives, and customers.

When her company adopted AI tools, Jenn approached them with the same confidence. She wrote prompts the way she wrote PRDs: thorough, specific, assuming the reader would understand her intent.

"Draft a product spec for a mobile checkout feature. Target users are existing customers who have used our desktop checkout. Focus on reducing friction—current mobile conversion is 23% below desktop. Consider our existing design system and API constraints. Prioritize speed over feature completeness for the MVP."

Jenn knew exactly what she meant. Reduce friction for familiar users who struggle with mobile. Use existing infrastructure. Ship fast.

The AI produced a competent spec—for the wrong problem. It optimized for reducing checkout steps rather than addressing the actual friction points mobile users experienced. It suggested design patterns that conflicted with the existing system. The "MVP prioritization" resulted in a stripped-down spec rather than a focused one.

Jenn refined her prompt. Added more detail. Specified what "friction" meant, what "existing constraints" included, what "prioritize speed" should look like. Forty minutes later, she had a usable draft.

She could have written it faster herself.

"AI doesn't understand context," she told her team. "You have to specify everything."

Jenn's assessment was accurate but her diagnosis was wrong.

The AI didn't fail because it lacked understanding. It failed because Jenn assumed it would understand the way a human would.

When a human product manager reads "reduce friction for existing customers," they draw on years of experience. They know what friction typically means in checkout flows. They understand why existing customers might behave differently than new ones. They can infer what "our existing design system" constrains without being told.

The AI has none of this. It has patterns learned from vast amounts of text, but it doesn't share Jenn's context. It doesn't know her company, her customers, her constraints. It interpreted her words through statistical patterns rather than experiential understanding.

Jenn was projecting human cognition onto a system that doesn't work that way.

This is the Projection Trap: assuming AI thinks like you do.

The trap is nearly universal because human brains evolved to model other humans. When we communicate, we automatically assume human-like cognition on the other end. Shared context. Similar inference patterns. Comparable reasoning processes.

These assumptions work brilliantly with humans. They fail systematically with AI.

The Human Assumptions That Don't Transfer:

What We Assume	AI Reality	Resulting Failure
"It remembers what we discussed"	Context window is finite; earlier context fades	AI "forgets" crucial context
"It knows what I mean"	AI interprets literally what you say	Output misses unstated intent
"It will ask if confused"	AI often generates confident responses despite uncertainty	You receive plausible but wrong output
"It shares my priorities"	AI optimizes for what you measure, not what you value	Technically correct, practically useless
"It will push back if I'm wrong"	AI has tendency toward agreement	Your errors get reinforced
"It understands context implicitly"	Context must be explicitly stated	AI misses "obvious" information
"It reasons like I do"	AI pattern-matches; different from human reasoning	AI's "logic" seems inconsistent

Every row represents a projection failure. Every projection failure produces frustration, wasted time, or flawed output.

The Projection Trap is particularly insidious because successful interactions reinforce it.

When AI responds well, it seems to understand. The illusion deepens. You start assuming more shared context, more implicit understanding, more human-like cognition. Then you hit a failure that makes no sense—the AI suddenly "forgot" something obvious, or produced output that contradicts what you clearly meant.

The failure didn't come from nowhere. The AI was always operating the same way. Your model was wrong, and success hid the error until it couldn't anymore.

Tom Brennan lived in the Projection Trap for months. His prompts were precisely specified—by human standards. When they failed, he blamed the AI's limitations. He never questioned whether his model of AI cognition was accurate.

Claire Morrison avoided the trap instinctively. She assumed difference rather than similarity. She provided context that "should be obvious" because she didn't assume the AI would infer it. Her success came from accurate modeling, not from superior prompts.

The Dismissal Trap

The opposite failure is equally limiting.

James worked in the same organization as Jenn. When AI tools rolled out, he felt the familiar wariness that had served him well through blockchain, metaverse, and a half-dozen other hype cycles. *Another silver bullet that will disappoint,* he thought. *Better to be practical.*

AI was just another tool, more sophisticated than a spreadsheet, but fundamentally the same category. You told it what to do, it did it, end of story. No need to understand the engine to drive the car.

He used templates his team had developed. When templates produced good results, he moved on. When they produced poor results, he tried different templates. *Find what works, skip the theory.* He'd built a successful career on this pragmatism.

"It's a black box," he told colleagues who wanted to discuss how AI worked. "No point trying to understand it. Just find patterns that work and stick with them."

James's approach felt efficient. He didn't waste time on speculation. He optimized for output.

But when circumstances changed—new AI model, new use case, new constraints—James couldn't adapt. His templates stopped working and he didn't know why. He couldn't debug failures because he had never built a model to debug against. The pragmatism that had served him well with other tools became a ceiling with this one.

His AI capability plateaued quickly and never developed further. He didn't understand why—wasn't he being practical?

This is the Dismissal Trap: treating AI as purely mechanical, a tool that doesn't require modeling at all.

The trap has its own appeal. It feels appropriately skeptical in a world of AI hype. It maintains clear hierarchy—AI is tool, human is user, no blurring of categories. It's efficient, skipping "unnecessary" understanding for practical results.

But dismissal produces the same limitation as projection: inability to adapt.

When you don't understand how AI processes information, you can't diagnose failures. You can't predict when your approach will work and when it won't. You can't adapt to new situations that your templates don't cover.

James never moved beyond Layer 2 of the Capability Stack—technique without leadership. He knew how to use the tools, but he didn't understand them well enough to lead the collaboration.

The Foreignness Spectrum from Chapter 2 applies here: treating AI as an Unknowable Mind when it's actually a Foreign Mind. Unknowable means "don't try to understand—just use templates." Foreign means "genuinely different but comprehensible with effort."

AI is Foreign. It can be modeled. The model requires work to build, but once built, it enables adaptation, diagnosis, and continuous improvement.

The True Model

Between projection and dismissal lies accurate understanding.

You don't need to be an AI researcher to build a useful model of AI cognition. You need enough understanding to predict behavior, diagnose failures, and adapt your approach.

Here's what that understanding involves:

1. Pattern Matching, Not Reasoning

AI identifies patterns in its training data and applies them to your input. What looks like reasoning is sophisticated pattern completion.

This is powerful—AI can recognize patterns across vast domains that no human could master. But it's fundamentally different from human thought. When AI produces a logical-seeming argument, it's generating text that matches the pattern of logical arguments, not actually reasoning through premises to conclusions.

Implication: Don't assume AI has "understood" your reasoning. It has matched patterns that resemble your input. Verify conclusions independently.

2. Context Window as Working Memory

AI has limited "working memory"—a context window that holds your conversation. Earlier parts of long conversations may fade, be summarized, or lose detail.

Implication: Don't assume AI remembers everything you've discussed. In long conversations, restate crucial context. For important work, keep context windows fresh.

3. Optimization Targets

AI is trained to produce responses that humans rate highly. This creates pressure toward helpfulness, fluency, and agreement—sometimes at the expense of accuracy or appropriate pushback.

Implication: AI wants to be helpful. This can mean telling you what you want to hear rather than what's accurate. Build in verification for consequential decisions.

4. Probabilistic Generation

Each word (technically, each token) is a probability-weighted choice based on context. Small changes in input can produce very different outputs. What seems like consistent behavior comes from strong patterns, not from stable "beliefs."

Implication: Inconsistency is normal. If you need consistency, verify it—don't assume it.

5. No Persistent Self

AI doesn't carry state between conversations (with some exceptions for products built to remember). Each conversation is essentially a fresh instance. What seems like personality is pattern, not identity.

Implication: Don't rely on AI "learning" your preferences across conversations unless the system explicitly supports it. Treat each conversation as starting from scratch.

Here's a working model that captures these principles in practical form:

Think of AI as an extraordinarily well-read research assistant who has never actually done anything.

They've read about everything—every domain, every methodology, every perspective. They can pattern-match to what they've read with remarkable sophistication.

But they've never had a job. They've never made a decision with consequences. They don't know your specific context because they've only read about contexts in general.

They're eager to help—perhaps too eager. They'll produce confident answers even when they should acknowledge uncertainty. They'll agree with you when they should push back. They'll give you what you asked for even when what you need is different.

And they start each conversation fresh. Whatever you discussed before, unless it's in the current conversation, doesn't exist for them.

This model isn't technically precise. But it's useful. It predicts behavior, explains failures, and guides adaptation.

Building Your Model

Understanding is not static. It deepens through practice.

Ender didn't model his opponents once and then stop. He observed continuously. He updated constantly. Every interaction was data for refining his understanding.

The same practice applies to AI collaboration:

The Post-Interaction Analysis

After significant AI interactions, ask:

- *How did the AI interpret my prompt?* Was it what I intended, or did it understand something different?
- *Where did the AI surprise me?* What does that surprise reveal about how it processes information?
- *Where did the AI fail?* What was it optimizing for that diverged from my intent?
- *What patterns emerge across interactions?* What consistent behaviors am I learning to predict?

Claire Morrison practiced this naturally. When the AI's first suggestion was generic, she didn't conclude "AI is limited." She concluded "I haven't communicated well enough." She analyzed her own prompt, identified the gap, and refined her approach.

Tom Brennan never did this analysis. When AI failed, he refined his prompt without examining why the original failed. He accumulated better prompts but never developed better understanding.

The Prediction Practice

Before sending complex prompts, predict:

- How will the AI interpret ambiguous terms?
- What context will it miss that I haven't provided?
- Where is it likely to produce confident output despite uncertainty?
- What failure modes are likely for this type of task?

Then compare your predictions to actual results.

This practice does two things. First, it surfaces assumptions—you can't predict without making your mental model explicit. Second, it provides feedback—when predictions fail, your model updates.

Over time, predictions improve. The model becomes more accurate. You start anticipating AI behavior before it happens, adjusting your approach proactively rather than reactively.

The Model Refinement Cycle

```
OBSERVE: Note what the AI actually does
HYPOTHESIZE: Propose explanation for AI behavior
PREDICT: Based on hypothesis, predict next behavior
```

```
TEST: Try interaction that tests the prediction
UPDATE: Refine model based on results
REPEAT: Continuously improve model accuracy
```

This is the scientific method applied to AI understanding. Observe, hypothesize, predict, test, update. The cycle never ends because your model can always improve.

The Projector

The anti-pattern that emerges from failing to model AI accurately.

Bonzo Madrid could not model Ender Wiggin.

He saw Ender only through his own frame: a smaller boy who threatened his status, an insult to his honor that demanded satisfaction. He couldn't imagine that Ender thought differently than he expected, processed situations through different values, would respond in ways Bonzo couldn't anticipate.

This blindness was lethal. In their final confrontation, Bonzo expected Ender to fight the way a subordinate challenges a superior—accepting the rules of honor that Bonzo lived by. Ender didn't. He fought to survive, with no concern for honor, using every advantage he could find.

Bonzo couldn't predict this because he couldn't model it. His frame couldn't accommodate a mind that worked so differently from his own.

The AI parallel is **The Projector**.

The Projector assumes AI thinks like a human. They project their own cognition, context, and inference patterns onto a system that works differently.

Warning Signs:

- "Why doesn't it understand what I obviously meant?"
- "The AI is stupid/broken/wrong"
- Repeated failures without approach adjustment
- Frustration that AI "should know" things that were never stated

Root Cause:

Human brains evolved to model other humans. The projection is automatic. We have to work to overcome it.

The Deeper Pattern:

The Projector often has strong communication skills—with humans. Jenn's PRDs were legendary. Her prompts failed not because she couldn't communicate, but because she was communicating for a human audience that wasn't there.

The same strength that makes someone effective with humans can become a liability with AI. The more instinctive your human-modeling skills, the harder you have to work to model something genuinely different.

Correction:

- Accept that AI cognition is genuinely different—Foreign Mind, not Familiar Mind
- Practice post-interaction analysis to surface projection failures
- Build and refine your AI cognition model through observation

- Update your approach based on evidence, not frustration

Self-Check Questions:

1. Am I assuming the AI will infer context I haven't provided?
2. When AI surprises me, do I examine my own assumptions?
3. Can I predict how AI will interpret this prompt before I send it?

Bonzo died because he couldn't model Ender. The stakes in AI collaboration are lower—no one dies from a failed prompt. But the pattern is identical: projection produces failure, and the failure feels like the other's fault rather than your own modeling error.

Ender succeeded because he could model minds that worked nothing like his own. The Formics were genuinely alien—hive consciousness, collective decision-making, values and priorities incomprehensible to humans. Ender modeled them anyway. His accuracy in understanding how they thought is what enabled him to find their vulnerability.

The same capability—accurate modeling of different cognition—is what separates AI leaders from AI users.

The Practice Begins

You cannot lead what you do not understand.

The Understanding pillar rests on this foundation: accurate mental models of AI cognition, built through observation and refined through practice.

This is not a one-time exercise. Your model should evolve with every interaction. The AI isn't changing—but your understanding can always deepen.

The practitioners who develop this capability accomplish things that template-users cannot approach. They produce better outputs because they anticipate how prompts will be interpreted. They diagnose failures because they understand what produced them. They adapt to new situations because their model transfers beyond specific use cases.

They operate at Layer 4 of the Capability Stack: leadership, not just technique.

A Foreign Mind. Genuinely different. Reachable with effort. Worth the work to understand.

Reflection Questions

1. Using the Human Assumptions Table, which projections do you recognize in yourself? Be specific. When has assuming shared context, implicit understanding, or human-like reasoning caused an AI interaction to fail?

2. Practice the post-interaction analysis on a recent significant AI interaction. How did the AI interpret your prompt? Where did it surprise you? What does the surprise reveal about how it processes information?

3. Before your next complex AI interaction, practice prediction. Write down how you expect the AI to interpret key terms, what context it will miss, and what failure modes are likely. Compare predictions to results. What does the comparison reveal about your current model?

Next: Chapter 5, Calibrating Otherness

End of Sampler

Thank you for reading this chapter sampler. The full manuscript includes:

- **16 additional chapters** covering all Five Pillars, application contexts, and integration
- **5 appendices** with frameworks, case studies, assessments, and workshop designs
- **13 anti-patterns** with recognition and correction guidance
- **Reflection questions** at the end of every chapter

For the complete manuscript or to discuss endorsement, please contact:

Steve Harlow sgharlow@gmail.com linkedin.com/in/stevengharlow

---

"For everyone learning to lead minds different from their own. And for Ender, who showed us how."