

TEACHING GUIDE

Operating Intelligence

How Enterprises Build AI That Lasts · Brandon Chiazza

Chapter summaries, learning objectives, frameworks, discussion questions, and applied assignments for every chapter, with three course maps.

First edition, Silent Way Press, 2026. Paperback ISBN 979-8-9963373-0-9. Kindle ISBN 979-8-9963373-1-6.

Examination copies and the companion artifacts: operatingintelligence.org/#instructors

Questions and guest sessions: press@silentwaypress.com

Teaching Guide

Operating Intelligence: How Enterprises Build AI That Lasts by Brandon Chiazza. Silent Way Press, 2026. Paperback ISBN 979-8-9963373-0-9, Kindle ISBN 979-8-9963373-1-6. 477 pages.

How to use this guide

Each chapter entry gives a summary, three learning objectives, the frameworks and artifacts the chapter introduces, the Laws it develops, four discussion questions, one applied assignment that produces a deliverable, and one line worth putting on a slide. The summaries describe what the chapter argues, in the book's own terms, so an instructor can decide what to assign without rereading. The discussion questions require the chapter's argument to answer. The assignments build on one another: a student who completes them in order finishes the term with a scored Twelve Laws Self-Assessment, a Context Layer Diagnostic, a one-page AI Proof-of-Value Charter, a control map, and a value review of the same initiative.

The three companion artifacts (the Twelve Laws Self-Assessment, the AI Proof-of-Value Charter, and the Context Layer Diagnostic) are free at operatingintelligence.org and may be reproduced for classroom use. Chapter 6 is available as a free sample PDF at operatingintelligence.org/sample-chapter.pdf. Instructors can request a paperback examination copy at operatingintelligence.org/#instructors. Course maps for a twelve-week seminar, a six-session executive module, and a three-hour workshop follow the chapter entries.

The Twelve Laws of Enterprise AI are the book's spine. Each chapter entry lists which Laws it develops, so a course built around the Laws can be sequenced from the front matter alone.

Part I: Strategic Foundations, Part II: Data, Context, and Governance, Part III: Strategy to Execution

Part I, Strategic Foundations. Short by design, the Part exists to install the premises the rest of the book builds on. Chapter 1 asks what makes this technological moment different, Chapter 2 asks what executive judgment AI now requires, and Chapter 3 shows how to diagnose where an organization stands. Two premises carry forward, according to the part intro: proof of value, not proof of concept, is the operating discipline that separates a demonstration from organizational change, and most organizations are still at Level 1 or 2 of the AI Maturity Model in most functions.

Part II, Data, Context, and Governance. Three chapters build the book's central architectural commitment in order. Chapter 4 makes knowledge usable, Chapter 5 makes it available at runtime as the context layer, and Chapter 6 makes it trustworthy. Parts III through VI, the intro says, assume this work has been done, and an organization shipping AI without the context layer is shipping AI that operates on documents instead of on how the organization works. MyCity anchors the cautionary argument, the AI Data Trust Model supplies the governance frame, and the Architecture Decision Matrix is the working artifact.

Part III, Strategy to Execution. Part III runs from destination through selection to proof and sequence. Chapter 7 develops the vision, Chapter 8 turns it into an evidence-based portfolio, and Chapter 9 converts the portfolio into a roadmap with Charters, scale gates, platform dependencies, and explicit scale-or-stop decisions. Two public cases carry the Part, DBS Bank in Chapter 7 as a vision that cascaded and Morgan Stanley in Chapter 9 as the sequencing discipline that must follow. Both, the intro is careful to say, are read for their structure rather than their headline numbers, and both rest on published accounts rather than independently verified outcomes.

Chapter 1: The Inflection Point: Why This Wave Is Different

Summary. The chapter opens with the exposure numbers from an OpenAI and University of Pennsylvania study: roughly eighty percent of the United States workforce could have at least ten percent of their tasks affected by large language models. Chiazza reads these as exposure figures rather than displacement forecasts. AI, on the chapter's account, is a general-purpose technology in the Bresnahan and Trajtenberg sense and the first to reach a broad range of cognitive tasks directly, so the dividend depends on organizational redesign rather than on model access. Paul David's account of factory electrification supplies the historical pattern, and a commercial underwriter whose day is rearranged rather than abolished supplies the modern one. Also introduced are the shift from deterministic to probabilistic systems, role decomposition, the prediction-and-judgment decoupling from Agrawal, Gans, and Goldfarb, the Task Decomposition vs. AI Applicability Framework, and a Coasean reading of second-order effects on firm boundaries.

Learning objectives. - Distinguish task exposure from occupational displacement, and explain why the chapter treats the exposure numbers as a measure of reach rather than of outcome. - Apply the Task Decomposition vs. AI Applicability Framework to a professional role, scoring its tasks on structure, data availability, risk tolerance, and judgment dependence. - Evaluate why the assurances a deterministic system offered (testing, trust, explainability) do not transfer to a learned system, and what organizational muscle the chapter says replaces them.

Frameworks and artifacts. - The general-purpose technology framing (Bresnahan and Trajtenberg), with Paul David's account of the electric motor and the factory - Figure 1.1, Industrial Revolutions Comparison - The shift from deterministic systems to probabilistic intelligence, and Figure 1.2, AI Capability Curve: Deterministic to Probabilistic to Agentic - Role decomposition, illustrated by the commercial underwriter - The prediction-and-judgment decoupling (Agrawal, Gans, and Goldfarb), illustrated by the statewide procurement supplier-participation case - Figure 1.3, Workflow Transformation Model - The Task Decomposition vs. AI Applicability Framework, in three stages (Role Decomposition, Applicability Scoring, Applicability Mapping) and four scoring dimensions (Structure, Data Availability, Risk Tolerance, Judgment Dependence), and the AI Applicability Spectrum - Figure 1.4, Task vs. Role Automation Spectrum - Second-order effects on decision velocity, coordination costs (Coase and Williamson), and competitive dynamics

Laws developed. Law 1 (models are the smallest part of enterprise AI advantage), introduced through the dynamo analogy. Law 12 (operating-model change is the price of AI adoption, not its consequence), introduced in the discussion of task automation versus role transformation.

Discussion questions. 1. The chapter says the ATM did not eliminate bank tellers and the spreadsheet did not eliminate accountants. What in its argument lets you predict which of today's knowledge-work occupations will follow that pattern and which will follow the displacement pattern Acemoglu and Restrepo describe? 2. Why does the chapter insist that trust in a probabilistic system must be calibrated rather than binary, and what does it say a workforce has to learn that a traditional IT function was never built to teach? 3. In the statewide procurement case, the system made the assumptions behind a supplier-participation goal cheap to vary, and the chapter says the accountability "moves nowhere at all." Using the prediction-and-judgment decoupling, explain why that is the intended outcome rather than a limitation. 4. Choose a role in an organization you know. Which of its tasks would the chapter's four scoring dimensions push toward full automation, and which would they keep human-only? Where does the chapter's treatment of tacit knowledge change your scoring?

Applied assignment. Role decomposition brief. Select one professional role in an organization you know well and produce a one-page task decomposition following the chapter's three stages. List the role's constituent tasks (the chapter expects a couple of dozen for most roles), score each on the four applicability dimensions, and place each on the AI Applicability Spectrum. Close with a short paragraph naming which tasks would move to the system, which would stay with the person, and what the redesigned role concentrates on, using the chapter's commercial underwriter as the model for that paragraph.

Quotable line. "AI adoption is a work-design problem before it is a technology problem." (line 83)

Chapter 2: How Executives Should Think About AI

Summary. The chapter opens with a CIO asked by the board how fifty pilots are going, and the honest answer is “I don’t know.” Most of the technology works, and nobody has decided which pilots to scale, who would own the result, or what the organization would give up to make room. Chiazza argues that filing AI under IT mistakes the delivery mechanism for the thing delivered, and offers the quality management revolution, not an ERP implementation, as the better analogue. Three connected disciplines follow: the AI Value Pyramid, which sorts investments by how long the advantage lasts, decision rights pushed toward the people doing the work, and proof of value, defined against proof of concept and made the unit of investment, with its seven elements written as the Proof-of-Value Charter. Closing sections name operating intelligence, five common executive missteps, the Executive Decision Lens, and the AI Portfolio Balance Matrix.

Learning objectives. - Distinguish a proof of concept from a proof of value, and name the seven elements that make the distinction operational. - Evaluate an AI initiative against the six dimensions of the Executive Decision Lens, treating Proof-of-Value Readiness as a hard gate. - Diagnose which of the five executive missteps a pilot-heavy program exhibits and what the chapter prescribes for each.

Frameworks and artifacts. - The IT misclassification and the Total Quality Management analogue (Deming) - The AI Value Pyramid (Efficiency, Capability Expansion, Strategic Reinvention), with Figure 2.1 and the Framework Summary table - The redistribution of decision rights, illustrated by the regional manager who reviews only the deals the system flags, and Figure 2.2, Operating Model Shift Map - Proof of value and its seven elements (baseline, value hypothesis, user population, evaluation method, adoption criteria, risk boundaries, and the scale, redesign, or stop decision), written down as the Proof-of-Value Charter - Operating intelligence, defined as the capacity to turn AI capability into organizational outcomes repeatedly, and to know that it has - Common Executive Missteps (five) - The Executive Decision Lens (six dimensions), with Figure 2.3 - The AI Portfolio Balance Matrix (Quick Wins, Smart Investments, Transformational Bets, Questionable Bets) and its efficient frontier, with Figure 2.4

Laws developed. Law 3 (a proof of concept shows that the technology can perform, a proof of value shows that the work improves), stated in the chapter as its own line. Law 12 (operating-model change is the price), applied to the pilot-proliferation pattern.

Discussion questions. 1. The chapter says the CIO’s honest answer to the board is “I don’t know.” Using its argument, explain why that is not a technology failure and what decisions would have to be made before a better answer became possible. 2. Efficiency gains are described as real, measurable, and the fastest to show ROI. Why does the chapter still say programs seeking lasting advantage should not stop there, and what does the second tier require that a competitor running the same models does not get for free? 3. Take the seven elements of a proof of value. How many does a typical first-draft initiative manage to name, according to the chapter, and why does it say the seven “work as a system” rather than as a checklist? 4. Pick a pilot or AI initiative in an organization you know and score it on the Executive Decision Lens. Does it clear the Proof-of-Value Readiness gate? If not, which of the seven elements is missing, and who in that organization would have to supply it?

Applied assignment. Decision Lens memo. Choose one live or proposed AI initiative and write a one-page memo to its executive sponsor. Score it on all six dimensions of the Executive Decision Lens, with one sentence of evidence per score. Then test it against the seven proof-of-value elements and state plainly whether it clears the Proof-of-Value Readiness gate. If it does not, name the missing element and what it would take to supply it. The AI Proof-of-Value Charter companion artifact serves as the checklist for the seven elements.

Quotable line. “The pilot-proliferation pattern is a symptom of that misclassification: when AI is filed under technology, the technology gets funded and the operating model does not.” (line 9)

Chapter 3: The AI Maturity Model

Summary. Many organizations have crossed the experimentation threshold. Far fewer have crossed the one that matters, which the chapter defines as the move from a collection of point solutions to an enterprise capability with shared infrastructure, governance that survives any single initiative, and proof-of-value pathways that produce decisions. The AI Maturity Model names five levels, from Experimentation through Operationalization, Scaling, and Transformation to Autonomous Systems, and describes what proof of value looks like at each. Most organizations sit at Level 1 or 2, and the stall between Levels 2 and 3, the Platform Gap, is where the work stops being what produced the early wins. The Maturity Assessment Framework scores seven dimensions, including the context layer as its own row. Each level is then read by the artifacts an organization can produce on demand rather than by adjectives, and three stall patterns with their interventions close the chapter.

Learning objectives. - Diagnose an organization's maturity dimension by dimension, locating the binding constraint rather than averaging into a single grade. - Distinguish the artifacts that mark each level from the adjectives that describe it, and test a claimed level against what the organization can produce on demand. - Evaluate which of the three stall patterns an organization exhibits and what investment the chapter says the crossing requires.

Frameworks and artifacts. - The AI Maturity Model, five levels (Experimentation, Operationalization, Scaling, Transformation, Autonomous Systems), each with the form proof of value takes at that level - Figure 3.1, AI Maturity Curve, with the Platform Gap between Levels 2 and 3 - Bounded autonomy as the defining condition of Level 5 - The AI Maturity Assessment Framework and its seven dimensions (Data Infrastructure, Technology Platform, Context Layer, Governance, Talent & Skills, Org Integration, Value Realization) - Figure 3.2, Capability Progression by Level, and the dimension-by-level table - Diagnosing maturity by artifact: the table of artifacts present and absent at each level - Figure 3.3, Governance Depth Model - Common Stall Patterns and Interventions (three) - Figure 3.4, Transformation Pathway

Laws developed. Law 12 (operating-model change is the price), which the chapter says holds at every level and whose unit of redesign expands from the workflow to the enterprise at Level 4.

Discussion questions. 1. The chapter says the Level 2 to Level 3 crossing “stalls them because the work required to cross is not the work that produced the early successes.” What is that different work, and why can no platform team do it alone? 2. Why does the chapter name the context layer as its own dimension rather than folding it into data infrastructure or technology platform? What condition does a separate row force the assessment to see? 3. DBS reports its time-to-value fell from roughly eighteen months to two or three, and the chapter is careful to say this is the bank's own figure and that DBS publishes no maturity model. What does the chapter claim the reduction is consistent with, and what would a Level 2 organization be unable to do no matter how many models it accumulated? 4. Assess an organization you know against the artifact table. Which Level 3 artifacts can it produce on demand? Which exist only in slide form? What does that say about the level its own self-assessment reports?

Applied assignment. Maturity profile. Produce a scored maturity assessment for an organization you know, following the chapter's closing instruction. For each of the seven dimensions, record a level

rating, the artifact behind that rating, the missing artifact blocking the next use case, and the investment needed to remove that constraint. Present it as a one-page table with a short paragraph naming the binding dimension and which of the three stall patterns it corresponds to. Where a rating rests on an artifact that exists only in slide form, say so and lower the rating. The Twelve Laws Self-Assessment companion artifact can be scored alongside as a second readout.

Quotable line. “If the artifact does not exist, or exists only in slide form, the organization has not crossed into the level the slide describes.” (line 83)

Chapter 4: Data Readiness: The Real Bottleneck

Summary. A logistics CIO spent two years and tens of millions on a data platform and declared her company ready to “do AI.” Six months later, twelve of fifteen prototypes had stalled at the same point, where the model met the actual data environment. Data, not the model, is the bottleneck, the chapter argues, and the failure sits in a substrate the organization never built for runtime AI access. Enterprise knowledge falls into three categories, structured, unstructured, and tacit, and the third is the most valuable and the least managed. Five tacit-knowledge elicitation methods follow, then the ground-truth problem and the four sources from which a defensible evaluation standard is built. The Data Readiness Scorecard scores nine dimensions, including a Workflow / Permission Context dimension that traditional assessments do not see, and the chapter insists it be read by its lowest score rather than its average.

Learning objectives. - Distinguish structured, unstructured, and tacit knowledge, and explain why the chapter says the last cannot be captured by writing down what experts know. - Apply the Data Readiness Scorecard to a data domain and read the result by its binding constraint rather than its average. - Diagnose, for each weak scorecard dimension, the specific production failure the chapter says it produces.

Frameworks and artifacts. - Figure 4.1, Data Readiness Stack - The Three Categories of Enterprise Knowledge (structured, unstructured, tacit), with Polanyi and Hayek as the anchors, and Figure 4.2, Knowledge Type Map - Five tacit-knowledge elicitation methods: structured expert interviews, case-based reasoning sessions, annotated examples, decision journaling, pair-with-AI sessions - The Ground Truth Problem and the four sources of an evaluation standard (expert review panels, curated evaluation datasets, continuous feedback loops, red-team exercises), with Figure 4.3, Ground Truth Feedback Loop - The Data Readiness Scorecard, nine dimensions scored 1 to 5, read by constraint - “What fails when each dimension is weak,” a production failure for each of the nine - The payment-reporting integration for a public power organization, as the Integration illustration - The bank pattern (six months in, foundation work still required) and Figure 4.4, Enterprise Data Friction Map

Laws developed. Law 4 (baseline before you build), introduced through the ground-truth problem. The ground-truth artifact is the material the baseline is built from, and the hesitation to commit to an imperfect baseline is the failure.

Discussion questions. 1. The chapter cites the NewVantage survey finding that culture, not technology, blocks the data-driven enterprise, and then offers a “better reading.” What is it, and how does it change what a program leader should fund first? 2. Why does the chapter say the strategy of “we’ll write down what they know” reliably fails? Which of the five elicitation methods avoids that failure, and what does the output of elicitation feed instead of a knowledge document? 3. In the payment-reporting integration, the interface was “the easy part.” What had to be settled first, and why does the chapter say an assistant later asked to summarize payments could be internally coherent and still wrong? 4. Score a data domain in an organization you know on the nine dimensions. What is the lowest score, and what does the chapter say that weakness looks like on the day the system reaches production?

Applied assignment. Scored readiness assessment. Choose one data domain that a specific AI use case would depend on and score it on all nine dimensions of the Data Readiness Scorecard, with a one-line evidence note per score. Identify the lowest-scoring dimension, quote the chapter's description of the failure that weakness produces, and state whether the domain is deployable, workable with documented remediation, or blocked, using the chapter's thresholds. Close with a written approximate baseline for the use case's current cycle time or error rate, and a sentence on what makes the approximation defensible.

Quotable line. "An average hides the dimension that will stop you." (line 93)

Chapter 5: Data Architecture for AI

Summary. Early AI architecture work starts with the wrong question, which is where the data lives. Data lake or warehouse, Postgres or vector database, are storage decisions one level below the architecture, the chapter argues. The architecture is how organizational context reaches the AI system at runtime, carrying the identity of the person asking, fresh enough to be true, and traceable to a source someone owns. The context layer is defined as that governed, runtime-accessible layer, with five properties and a list of what it is not (documents, prompts, RAG, a vector database). From there it covers the Context Delivery Architecture and its four patterns, the medallion architecture and where it stops, storage and indexing choices, and the graph-versus-vector tradeoff. A six-step construction sequence and three decisions from the Architecture Decision Matrix make the layer buildable. A hybrid-search illustration over legislative content closes the chapter.

Learning objectives. - Distinguish the context layer from the mechanisms that deliver from it or protect it, using the chapter's five properties. - Apply the six-step construction sequence to one workflow and identify which step a given organization has skipped. - Evaluate the three Architecture Decision Matrix decisions (source authority, permission propagation, freshness policy) for a specific use case.

Frameworks and artifacts. - The context layer, defined with five properties: Governed, Runtime-accessible, Source-authoritative, Permission-scoped, Operationally aligned - The public-sector procurement mini-bid illustration, in which one missing relationship between contract and program separated records the system could find from a pool it could assemble - Hayek and Scott as the two outside ideas that frame the layer's purpose and its hazard - The Context Delivery Architecture and its four patterns: Retrieval-Augmented Generation, tool calls into structured systems, graph-based retrieval over relationship data, governed prompt context, with Figure 5.1 - The Medallion Architecture (Bronze, Silver, Gold) and where it stops, with Figure 5.2 - Storage and Indexing Patterns (vector storage, hybrid retrieval, graph storage, structured systems of record, object storage and document stores) - Figure 5.3, Data-to-Inference Flow - Knowledge Graphs vs. Vector Retrieval, the subway map and the music-recommendation system, with Figure 5.4 - Constructing a Context Layer: the six-step construction sequence and the minimum-viable production context layer - The Architecture Decision Matrix and the three decisions that carry disproportionate weight: source authority, permission propagation, freshness policy - Hybrid search for legislative content (a keyword index kept for precision, a semantic-retrieval path added, and a fusion step)

Laws developed. Law 2 (the context layer is the part of the AI stack that does not depreciate when the model changes), given here in its architectural form. Law 1 (models are the smallest part of enterprise AI advantage), shown at the scale of a single component in the hybrid-search illustration.

Discussion questions. 1. The chapter lists documents, prompts, RAG, a vector database, a context window, a memory store, and a knowledge graph as things the context layer is not. Pick two. What does each deliver from or protect, and why can a team stand one up and believe the layer now exists? 2. In the mini-bid illustration, "association is not eligibility." Explain, in the chapter's terms, why a supplier name found in a document and a supplier returned by an eligibility query carry different evidentiary

weight, and why retrieval quality cannot close that gap. 3. Step 4 says permission scoping is a property of the retrieval call rather than a filter on its output. Why does the chapter treat filtering the generated answer as too late, and what does that imply for a team that added a permission model after the pipeline was already in service? 4. Choose a workflow in an organization you know and walk it through the six construction steps. Which step would that organization most likely skip, and what does the chapter say the cost of skipping it looks like two months later?

Applied assignment. Context layer decision record. Take one workflow an AI system would serve in an organization you know and run the Context Layer Diagnostic companion artifact against it, scoring all twenty questions across its seven sections. Report the section minimum as well as the total, following the artifact's own methodological note. Then write a one-page record of the three Architecture Decision Matrix decisions for that workflow: which source is authoritative for the content the workflow turns on, how identity will propagate into retrieval, and what freshness the decision requires. Name which of the six construction steps the diagnostic shows is missing.

Quotable line. “An organization that has implemented medallion and called the work finished has built the floor, not the room.” (line 87)

Chapter 6: Data Governance for AI

Summary. A small business owner asks New York City’s chatbot whether her landlord can refuse a Section 8 voucher, and the bot says yes. Source-of-income discrimination has been illegal in the city since 2008. AI needs governance traditional data programs were not built to provide, the chapter argues. Generated text draws on sources whose authority the user cannot see, and provenance can go opaque between retrieval and generation. The AI Data Trust Model names five pillars, Data Provenance, Access Control, Quality Assurance, Usage Governance, and Auditability. The MyCity, MyAI case is read as a failure across four of them, with a sixth dimension, operational trust, added by the case, and Diane Vaughan’s normalization of deviance names the pattern. Then the chapter supplies a three-tier Governance Operating Model, written shutdown and escalation criteria in five areas, an AI Production Shutdown Decision Record, and six enforcement points along the AI data pipeline.

Learning objectives. - Diagnose a public AI failure against the five pillars of the AI Data Trust Model and identify which pillars the documented record shows failing. - Write shutdown and escalation criteria for an AI system covering the five areas the chapter requires. - Distinguish a control from a disclaimer, and locate where along the pipeline each of the six enforcement points operates.

Frameworks and artifacts. - The AI Data Trust Model, five pillars: Data Provenance, Access Control, Quality Assurance, Usage Governance, Auditability, with Figure 6.2 - The MyCity, MyAI case, from the October 2023 launch to the February 2026 shutdown, and Figure 6.1, MyCity Timeline and Governance Gap Map - Operational trust, the sixth dimension the case foregrounds - Normalization of deviance (Vaughan) as the name for the disclaimer-modification pattern - The Governance Operating Model, three tiers (central AI governance office, domain governance teams, AI system teams), with Figure 6.3 - The county contract-management illustration: a blocking supplier condition, a protest rule tied to one transition, and an advisory insurance flag - Shutdown and Escalation Criteria in five areas (pre-launch acceptance gates, incident classification, decision rights, review cadence, escalation triggers), with NIST’s AI Risk Management Framework as the reference - Artifact: AI Production Shutdown Decision Record (seven fields) - Enforcement Points Across the AI Data Pipeline, six of them (source registration, ingestion validation, retrieval authorization, prompt assembly, output validation, logging and audit capture), with Figure 6.4, Lineage and Control Layers - Figure 6.5, Policy Enforcement Architecture

Laws developed. Law 6 (a disclaimer is not a control), at the output validation enforcement point. Law 10 (every AI system needs a written rule under which it will be taken offline), in the shutdown criteria section. Law 11 (authority should match accuracy), in the operational-trust reading of MyCity. Law 2 is also assigned to this chapter by the front matter, and the chapter develops it by supplying the governance that makes the context layer trustworthy.

Discussion questions. 1. The chapter says the MyCity case “does not turn on any single technical flaw.” Walk through the four pillars the case fails and explain, for each, what would have had to exist before launch rather than after the Markup investigation. 2. Access control is the one pillar MyCity does not fail, and the chapter says the public design served to “shift the burden onto usage governance.” What does usage governance require of a public-facing system that access control cannot

supply? 3. In the county contract-management design, three governance conditions required three different pieces of code. Why does the chapter say a control built without the “active and unresolved” qualification fails in both directions, and why do both failures look like the control is working? 4. Take an AI system in production, or proposed, in an organization you know. Who today can take it offline, against what written criteria? If the answer is a budget review or a press conference, which of the five shutdown-criteria areas is missing?

Applied assignment. Shutdown decision record. For one AI system in production or proposed in an organization you know, complete the chapter’s AI Production Shutdown Decision Record, filling all seven fields: system scope, acceptance gates, incident classes, decision rights, review cadence, escalation triggers, and external-report response. Where the organization has no current answer for a field, write “none” and add one sentence on what the MyCity record suggests the consequence of that gap would be. Attach a short paragraph naming which of the six enforcement points the system currently implements and which it leaves to disclaimers.

Quotable line. “Governance fails when policy is captured in documents that the runtime does not read.” (line 151)

Chapter 7: Developing an AI Vision

Summary. A leadership team can spend months debating AI and never define what it wants AI to change. An AI vision, the chapter claims, answers a different question: what will be meaningfully different about how the organization operates, competes, serves customers, makes decisions, and manages risk when AI is embedded in the work. A table separates vision, strategy, and roadmap by the question each answers and the failure each is prone to. The AI Vision Cascade then connects enterprise vision to strategic objectives, capabilities, and initiatives through a discipline of traceability. Atop the cascade, the AI North Star must pass six tests and needs named ownership. The Enterprise AI Ambition Map maps Optimize, Augment, Transform, and Reinvent onto the Value Pyramid. DBS Bank is read as a vision that cascaded, and five failure patterns and an anonymized financial-services case show what happens when the cascade is missing.

Learning objectives. - Distinguish an AI vision from an AI strategy and an AI roadmap by the question each answers and the failure each is prone to. - Write an AI North Star and test it against the chapter's six tests, including context-layer realism. - Diagnose which of the five vision failures an organization's stated AI ambition exhibits, and where its cascade breaks.

Frameworks and artifacts. - The Vision, Strategy, and Roadmap distinction table - Figure 7.1, Strategy Alignment Model - The AI Vision Cascade, four levels (Enterprise AI Vision, Strategic AI Objectives, AI Capabilities, AI Initiatives), with Figure 7.2 and Figure 7.3, Vision to Initiative Flow - The five-question test for a Level 1 vision (business value, operating-model change, differentiation, context layer, responsible use) - Model-facing, operating, and context-layer capabilities at Level 3 - The AI North Star and its six tests (Specificity, Ambition, Business grounding, Operating-model clarity, Context-layer realism, Responsible-use boundaries), plus ownership and durability - The Enterprise AI Ambition Map (Optimize, Augment, Transform, Reinvent) and its mapping to the AI Value Pyramid, with Figure 7.4 - The DBS Bank case, with an interpretive application of the cascade - Common Failures, five of them: the slogan vision, the use-case vision, the technology vision, the platform-only vision, the context-free vision - Case Note: When the Vision Cascade Was Missing (an anonymized mid-cap financial-services firm)

Laws developed. None. The front matter's "Developed in" lines assign no Law to Chapter 7, and the chapter names none. Instructors may note that the context-layer realism test rests on the same ground as Law 2 and the operating-model clarity test on Law 12, both developed in other chapters.

Discussion questions. 1. The chapter says the DBS ambition to become an "AI-fuelled bank" reads as "no less aspirational" on its face than New York's AI Action Plan. What is the test the chapter uses to separate them, and what evidence would you need to apply it to a third organization? 2. Strategic objectives are "not use cases." Using the chapter's examples, explain the difference between "launch a knowledge assistant" and "expand access to specialist knowledge across distributed teams," and why the cascade breaks if the two are confused. 3. In the anonymized financial-services case, the firm answered the four-level question with a slogan, a use-case list, a vendor inventory, and silence. The chapter says the cost of the missing cascade was the three years rather than the pilots. What does that mean for how vision work should be timed relative to pilot funding? 4. Write down the AI vision of an

organization you know, as its leadership would state it, and run it through the six North Star tests. Which test does it fail first, and which of the five failure patterns does that failure most resemble?

Applied assignment. One-page cascade. For an organization you know, draft an AI North Star of one to three sentences and test it against the chapter's six tests, noting any it fails. Then build the four-level cascade beneath it: three to five strategic objectives with a measurable outcome each, the capabilities those objectives demand (including at least one context-layer capability), and the initiatives that build them. Every initiative must trace upward. Close by naming the accountable executive sponsor and the refresh cadence, as the chapter's ownership section requires.

Quotable line. "If an initiative cannot name the capability it builds, it is probably a tool experiment."
(line 143)

Chapter 8: Identifying High-Value Use Cases

Summary. The first sign of an undisciplined AI program is too many ideas, and the trouble begins when the list is mistaken for a portfolio. One distinction governs the chapter: a use case that scores well has earned the right to be tested, not the right to scale. Drawing on Davenport and Ronanki and on Agrawal, Gans, and Goldfarb's point, application, and system solution taxonomy, it names five traps, from novelty masquerading as value to context treated as content. The Use Case Scoring Framework evaluates six dimensions, with context-layer leverage embedded as a sub-test in each. The Use Case Funnel moves from ideation through screening, assessment, and portfolio construction to selection for proof of value, with a discovery pass before ideation illustrated by a New York City procurement program. DBS, Aviva, and a media operator's point-solution trap close the chapter as three contrasting cases.

Learning objectives. - Distinguish point, application, and system solutions, and identify which one a proposed use case actually is. - Apply the Use Case Scoring Framework to candidate use cases after writing local scoring anchors, recording confidence alongside each score. - Evaluate a portfolio for the two opposite failures the chapter names, too safe and too ambitious, and for its fit with the organization's maturity level.

Frameworks and artifacts. - The point solution, application solution, and system solution taxonomy (Agrawal, Gans, and Goldfarb), illustrated with a claims fraud flag - Five traps: novelty masquerades as value, political sponsorship substitutes for portfolio logic, feasibility is discovered too late, pilots are launched without baselines, context is treated as content - The Use Case Scoring Framework, six dimensions (Business Value, Strategic Alignment, Data Readiness, Technical Feasibility, Organizational Readiness, Speed to Value), with scoring anchors supplied for Business Value only - Context-Layer Leverage as a sub-test embedded in each dimension - The Use Case Funnel, five stages (Ideation, Screening, Assessment, Portfolio Construction, Selected for Proof of Value), with Figure 8.1 - The discovery pass before ideation, illustrated by the New York City procurement program (six steps drawn, fourteen found, and the forty-person agency) - Figure 8.2, Workflow Opportunity Heatmap - Figure 8.3, Use Case Prioritization Matrix - Figure 8.4, Feasibility vs. Value Model, four quadrants - Building a Balanced Portfolio: the four investment types and the three-category table (workflow efficiency, capability expansion, strategic transformation) - The operating sequence (derive, baseline, develop, evaluate), with selection completing the derive step - Case Study: DBS and Portfolio Discipline at Scale, Case Note: Aviva and the Value of Domain Focus, and Case Note: A Media Operator's Point-Solution Trap

Laws developed. Law 3 (proof of concept versus proof of value), made operational at the point of selection. Law 4 (baseline before you build), named in Trap 4.

Discussion questions. 1. The chapter's claims example runs the same fraud flag through a point solution, an application solution, and a system solution. Why does it say most programs "fund point solutions, advertise them as application solutions, and hope they will produce the value only system solutions produce"? 2. Only Business Value has scoring anchors in the chapter. Why does the chapter call that "a burden the team must pick up," and what goes wrong when two people score Organizational

Readiness from private definitions? 3. In the New York City procurement discovery, the team designed separation of duties into a workflow that a forty-person agency could not complete. The chapter says the control was not the problem. What was, and what should discovery have established? 4. List three AI ideas currently circulating in an organization you know. For each, which of the five traps is it closest to falling into, and does it pass the chapter's screening questions? Which one would you select for proof of value, and why is that not the same as selecting it for scale?

Applied assignment. Scored use-case assessment. Take three candidate AI use cases from an organization you know. First write local anchors for what a 1, a 3, and a 5 look like on each of the five dimensions the chapter leaves unanchored. Then score all three on the six dimensions, recording a confidence level and the evidence behind each score, and apply the context-layer sub-test table. Place them on the Feasibility vs. Value Model and select one for proof of value. Before that selection, run the Context Layer Diagnostic companion artifact on the chosen workflow, as the artifact's own guidance recommends ahead of Portfolio Construction.

Quotable line. "A confident score based on a weak assumption is worse than an uncertain score that exposes what must be learned." (line 62)

Chapter 9: Designing the AI Roadmap

Summary. An AI pilot can look successful and prove almost nothing. Leaders nod at a working demo, then the hard questions arrive about baselines, overrides, permissions, and whether the workflow owner agreed to change the process. A Gantt chart can only record commitments and cannot test them, so the chapter treats the roadmap as an operating mechanism instead. The Operating Sequence, derive, baseline, develop, evaluate, runs for every funded use case, with baselining as the step most often skipped and approximations accepted. Three delivery phases, Pilot, Scale, and Platform, organize the work, and the AI Proof-of-Value Charter is the pilot artifact, tested by seven Charter Quality Tests and ending in a scale, redesign, pause, or stop decision. Morgan Stanley and MyCity close the chapter, after the Portfolio-Based Roadmap Model, seven dependency categories, and the Build vs. Buy Decision Model.

Learning objectives. - Write an approximate baseline for a use case and document what makes it defensible, following the chapter's list. - Apply the seven Charter Quality Tests to a draft Proof-of-Value Charter and decide whether it is ready to enter the roadmap. - Evaluate a build, buy, hybrid, or partner path for a capability using the five-factor decision framework and its decision matrix.

Frameworks and artifacts. - The Operating Sequence: Derive, Baseline, Develop, Evaluate, with Figure 9.1 - The rule for a defensible approximate baseline (unit of work, pre-AI period, case mix, source, width of uncertainty, sign-off) and the rule against updating it with results observed after the system went in - The Three Phases of AI Delivery (Pilot, Scale, Platform), with Figure 9.2, Delivery Phases - The AI Proof-of-Value Charter as the pilot artifact and the operating contract between pilot team and executive sponsor - Charter Quality Tests, seven of them: baseline, user, value, evaluation, adoption, risk, context - Scale, Redesign, or Stop: the conditions under which a use case should be stopped or redesigned, and the four-option exit-gate table (Scale, Redesign, Pause, Stop) - The Portfolio-Based Roadmap Model and its three horizons, with Figure 9.3, AI Roadmap Horizon Model - Roadmap Dependencies, seven categories, with Figure 9.4, Dependency Map - The Build vs. Buy Decision Model, five factors and the practical decision matrix, with Figure 9.5 - Case Study: Morgan Stanley's Eval-Driven Generative AI Roadmap (Assistant and Debrief) - Sidebar: The MyCity Chatbot and the Need for Stop Criteria, with the questions a Charter would have forced

Laws developed. Law 4 (baseline before you build), installed as a roadmap discipline in step two of the Operating Sequence. Law 3 (proof of concept versus proof of value), converted from principle to roadmap mechanism through the Charter.

Discussion questions. 1. The chapter says a demo and a proof of value both control their conditions, "for opposite reasons." What are the two reasons, and how would you tell from a pilot's design which one it is doing? 2. Why does the chapter forbid updating a baseline with results observed after the system went in? What does the baseline become if a team does that? 3. Phases and horizons "both come in threes and both describe progression." What different questions do they answer, and what goes wrong for a portfolio that starves Horizon 3 versus one that overinvests in it? 4. Take one AI capability an organization you know is considering. Score it on strategic differentiation, context sensitivity, and commercial maturity, and read the likely path off the chapter's decision matrix. Where

does the “harder question” about whether a vendor solution can respect source authority and escalation rules change your answer?

Applied assignment. One-page charter. Using the AI Proof-of-Value Charter companion artifact, draft a charter for the use case selected in the Chapter 8 assignment, or for one an organization you know is about to pilot. Fill every field, including the approximate baseline with its defensibility notes, the adoption signal, the risk boundaries, and the stop criteria. Then run the seven Charter Quality Tests and mark each pass, fail, or partial. State which exit-gate decision (Scale, Redesign, Pause, Stop) the evidence you have today would support, and which of the chapter’s seven dependency categories would have to be resolved before the pilot could start.

Quotable line. “A sponsor who names in advance the evidence that would end the work has told the team that the evidence is going to be read, and a team that believes its evidence will be read gathers it differently.” (line 150)

Part IV: Building AI Systems, Part V: Operating AI at Enterprise Scale, Part VI: Value, Adoption, and Future

Part IV, Building AI Systems. The intro describes Part IV as the book’s technical core, written at the level a leader needs to govern the work. Its movement runs from capability through control, grounding, and action to coordination across Chapters 10, 11, and 12, with implementation mechanics held in the Technical Appendix. PolicyAssist threads through all three chapters, and the evaluation harness introduced in Chapter 10 is the connective tissue that makes prompts, retrieval, and agents governable.

Part V, Operating AI at Enterprise Scale. The intro frames Part V as the section that distinguishes a working prototype from a working program, and states its movement in three imperatives: choose the stack deliberately, secure the runtime boundaries, control operating costs without undermining trust. Chapter 13 assembles the tooling architecture, Chapter 14 maps the security surface and the seven-boundary Control Architecture, and Chapter 15 installs runtime ownership with the cost discipline. Law 6 is anchored here and Law 8 recurs throughout, with MyCity echoing in Chapter 14 as the deployment-control lesson.

Part VI, Value, Adoption, and Future. The intro’s line is that value is promised, then tested, then adopted, then banked. Chapter 16 distinguishes projected, realized, and banked value, Chapter 17 examines how workflow design and user trust affect adoption, and Chapter 18 asks which investments remain useful as models change. The Twelve Laws are restated in Chapter 18 as three commitments, with the Maturity Model as the diagnostic for what to build next.

Chapter 10: The Technology, at the Level Leaders Need

Summary. The chapter opens with a financial-services team that spent three weeks chasing cost projections off by an order of magnitude, a bug that turned out to be tokenization and that took so long to find because no one on the team had a working model of what tokens were. From there it builds the working model a leader needs to govern a live system: tokens, embeddings, the context window, inference, and why hallucinations follow structurally from next-token prediction. It separates the context window (a per-request budget) from the context layer (a governed organizational asset) and introduces PolicyAssist, the composite policy assistant that runs through Part IV. The second half treats prompting as production engineering, with the evaluation harness as the discipline that makes prompt changes testable and reversible. The chapter's claim is that these primitives depreciate fastest while the harness and the context layer carry across model generations.

Learning objectives. - Explain why hallucinations are a structural property of next-token prediction and name the five elements of the minimum viable defense. - Distinguish the context window from the context layer and state why a larger window does not substitute for context engineering. - Specify what an evaluation harness must contain before a prompt change is allowed to ship.

Frameworks and artifacts. - PolicyAssist, the composite running example for Part IV - Figure 10.1, Token Lifecycle - Figure 10.2, Context Window Model - The context window versus context layer distinction - The minimum viable defense: grounding, citation, abstention, evaluation, human review where stakes warrant - Functional handling of hallucinations (workflow, role, and decision-rights design) - The Quality–Cost–Latency Tradeoff Triangle (Figure 10.3) - Model Selection Map (Figure 10.4) and the model-tier configuration table - The LLM Design Decision Sheet (eleven fields) - The Five-Component Prompt Template and the Prompt Iteration Loop (mechanics in Technical Appendix A.2) - Function calling and structured output, with three recurring patterns - Failure Type Map (Figure 10.5) - The evaluation harness (seven elements)

Laws developed. Law 1 (models are the smallest part of enterprise AI advantage), Law 5 (design for abstention), and Law 8 (the evaluation harness is the precondition for everything else). All three are assigned to this chapter in the front matter and named in the text.

Discussion questions. 1. The chapter says the context window is where context arrives and the context layer is what the system retrieves from. Why does doubling the first only change the per-request budget, while the chapter calls doubling the second “the one that lasts”? (*Answerable by a non-engineer executive.*) 2. The SaaS assistant's accuracy collapsed after a provider's minor version upgrade, and the team learned of it through a churn cohort. Using the seven elements of the evaluation harness, identify which element would have caught it first and what the rollback would have looked like. 3. The minimum viable defense has five elements. Take a public-facing AI system your organization runs or is considering, identify which element is weakest, and describe what authoritative-voice failure would look like there. 4. The Modali Accelerate content-restriction control had to prove two things: violations get rejected, and clean content still succeeds. Why does the chapter call a control that also blocks the corrected document “an outage with a rationale,” and what does that imply for how release gates are tested?

Applied assignment. Complete an LLM Design Decision Sheet for one AI application your organization has deployed or proposed. Fill all eleven fields, including baseline, value hypothesis, model tier, and the scale/no-scale threshold. Then write a one-page memo explaining which fields the team could not fill from existing documentation and what that gap says about whether the application is governable. Use PolicyAssist’s design paragraph as the reference for the level of specificity expected.

Quotable line. Line 77: *Building enterprise AI without a working theory of hallucinations is building enterprise AI without a working theory of risk.*

Chapter 11: Retrieval-Augmented Generation

Summary. The chapter opens on a familiar architecture diagram in which “the RAG pipeline” is drawn as the system, and argues that the diagram mistakes a delivery route for the asset that travels over it. RAG is one mechanism among several for exposing a slice of the context layer to a model at runtime. The chapter then develops RAG as a discipline: retrieval strategy (dense, sparse, hybrid), reranking, context assembly with citation tagging and a written precedence rule for conflicts, and the evaluation harness as the least practiced and most depended-on part of the work. It lays out a six-step ground-truth workflow, six RAG failure types with their detection methods, and the anti-patterns that make evaluation scores unreliable. A legislative-search engagement at a public-business network shows hybrid retrieval in production, with the retrieval work assessed on its own merits apart from the wider program’s outcome.

Learning objectives. - Position RAG as one context-delivery mechanism and explain what the context layer contributes that the pipeline does not. - Describe the six-step ground-truth workflow and why inter-annotator agreement is measured with a chance-corrected statistic. - Match each of the six RAG failure types to the test in the harness that detects it.

Frameworks and artifacts. - Figure 11.1, RAG Architecture (offline ingestion and online query paths) - PolicyAssist’s worked pipeline configuration - Retrieval strategies: dense, sparse (BM25), and hybrid with reciprocal rank fusion - Reranking approaches: cross-encoder, LLM-as-reranker, learning-to-rank - Figure 11.2, Retrieval Flow - Context assembly: ordering, citation tagging, deduplication and conflict handling, token budgeting, permission scoping - Why AI evaluation is different (four properties) - The six-step ground-truth workflow - The evaluation harness as continuous discipline - Anti-patterns to avoid - The six RAG failure types - Figure 11.3, Grounded Response Loop - Real-World Example: Legislative Search at a Public-Business Network

Laws developed. Law 2 (the context layer does not depreciate when the model changes), Law 5 (design for abstention), and Law 8 (the evaluation harness is the precondition for everything else). All three are assigned in the front matter and named in the text.

Discussion questions. 1. The chapter says the retrieval mechanics will be replaced over the asset’s life while the governed context they deliver from will not. In your organization’s knowledge base, what would count as the asset and what would count as the route? (*Answerable by a non-engineer executive.*) 2. PolicyAssist resolves conflicting chunks with a precedence rule the organization agreed in advance, and abstains when precedence stays unresolved. Why does the chapter insist the system cannot be left to pick, and who in your organization would have to write that rule? 3. The chapter compares AI evaluation to clinical-trial methodology and lists four properties that make it different from unit testing. Which of the four makes a 92 percent faithfulness score a measurement rather than a guarantee? 4. In the legislative-search case, the team kept Solr and added a semantic path beside it. What would have been lost by replacing keyword retrieval outright, and how did the staged evaluation of real subscriber queries earn the change?

Applied assignment. Design a ground-truth workflow for one retrieval-backed assistant your organization could build. Follow the chapter’s six steps: define the evaluation question, sample

representative queries, build the rubric with worked examples from your own corpus, plan two-reviewer grading, choose the agreement statistic and justify its threshold from the cost of a labeling error, and set the refresh cadence. Deliver the rubric and a one-page plan. Use the Context Layer Diagnostic's Source Authority section to identify which sources the rubric must treat as authoritative.

Quotable line. Line 120: *Teams skip it because it produces nothing demonstrable, and pay for it later in the incidents it would have caught.*

Chapter 12: Agents, Autonomy, and Orchestration

Summary. The chapter opens with the difference between a chatbot that misleads a user and an agent with tools that can act on a wrong answer before anyone reads it. Its central claim is that agent design is the discipline of bounding autonomy so capability and accountability scale together. It builds the four-component Agent Architecture and the plan-execute-observe-revise loop, and sets out five principles of tool design with authorization enforced at the tool wrapper. It treats the Model Context Protocol and Skills as standardization decisions and sorts human oversight by risk and reversibility. The Cursor support-agent incident shows authority outrunning accuracy. The second half examines multi-agent coordination, including a composite coordination failure in which every specialist succeeds and the system still recommends a procurement violation, and argues that the smallest architecture that produces the workflow gain is usually not multi-agent.

Learning objectives. - Describe the Agent Architecture and explain why the Tool Interface, more than the planner, determines how much damage the system can do. - Apply the three principles of bounded autonomy (permissions before capability, reversibility scales autonomy, escalation as a first-class outcome) to a proposed agent. - Decide, using the chapter's four alternatives, whether a workflow requires multi-agent orchestration or something smaller.

Frameworks and artifacts. - Figure 12.1, Agent Architecture (Planner, Executor, Memory, Tool Interface) and the plan-execute-observe-revise loop - The agent harness, distinguished from the evaluation harness - ReAct (worked trace in Technical Appendix A.4) - Five principles of tool design and authorization - The Model Context Protocol (MCP) as tool-surface standardization - Skills as capability packaging, with three design rules - Memory tiers and the distinction between what the system may recall and what it may treat as authoritative - Three principles of bounded autonomy - Agent evaluation: trajectory, per-step, adversarial - Figure 12.2, Human Oversight Points - Figure 12.3, Agent Interaction Patterns (sequential, hierarchical, peer-to-peer, competitive) - The Orchestration Layer (workflow-engine, LLM-coordinator, hybrid) - Workflows versus agents, from Anthropic's Building Effective Agents - Anatomy of a Coordination Failure and the four integration-layer controls - The confused deputy

Laws developed. Law 7 (bounded autonomy) and Law 8 (the evaluation harness is the precondition for everything else). Both are assigned in the front matter and named in the text.

Discussion questions. 1. The chapter says an agent cannot be safer than the tools it is permitted to call. For an agent your organization might deploy, list the tools it would need and sort them by reversibility. Which ones would require human authorization under the chapter's rule? (*Answerable by a non-engineer executive.*) 2. In the Cursor incident, the support agent's statements "operated as policy" to customers. What made that write-equivalent authority, and which control from the chapter would have had to hold? 3. The coordination failure produced a recommendation to violate procurement policy with no agent reporting an error. Why does the chapter argue that better component engineering makes this failure quieter, and what does the rule that a coordinator returns either a synthesis or a contradiction, never a synthesis that suppresses one, change about the coordinator's output schema? 4. The Accelerate drafting-workflow example separates capacity from

activation. Why is the unit of evaluation the sequence of requests and not any single request, and what would an equivalent test look like for an assistant your organization runs?

Applied assignment. Write an action-boundary specification for one agent use case in your organization, following PolicyAssist's model: the typed tools it may call, read versus write separation, the reversibility class of each write, the human approval gate, the escalation triggers, the kill switch, and the per-session cost and time budgets. Then draft ten adversarial evaluation cases (injection-laden documents, scope-violation queries, cases that should trigger escalation) the harness would run on every tool or prompt change. Deliver both as a two-page design note.

Quotable line. Line 49: *An agent cannot be safer than the tools it is permitted to call.*

Chapter 13: AI Tooling Ecosystem

Summary. The chapter opens with a prototype that looks better than it is, a model API wired to a document store with a chat interface, working well enough to create momentum until production asks who can see which documents, where prompts are logged, and what happens when the provider changes a version. Those are architecture questions, and the chapter's claim is that the tooling decision is about what runtime the organization needs to operate a use case safely, economically, and continuously. It introduces the Five-Layer Tooling Stack (model access, context and retrieval, application and orchestration, infrastructure and integration, experience and workflow embedding) with observability, governance, and cost as cross-cutting disciplines, runs PolicyAssist down the layers to expose where specification thins out, offers eight framework-selection questions, and applies proof-of-value discipline to tooling choices. Modali Accelerate closes the chapter as a disclosed case the reader is told to discount.

Learning objectives. - Name the five layers and three cross-cutting disciplines of the Five-Layer Tooling Stack and explain what distinguishes an enterprise stack from a prototype stack. - Apply the eight framework-selection questions to a tooling choice and explain why they recommend categories and not vendors. - Map runtime ownership onto each layer of the stack for a system your organization runs.

Frameworks and artifacts. - The six questions a production AI stack should make answerable - The Five-Layer Tooling Stack (Figure 13.1, Production AI Tooling Stack) - Three cross-cutting disciplines: Observability and Evaluation, Governance and Security, Cost and Lifecycle Management - MCP as a platform capability at Layers 1 and 2 - PolicyAssist mapped across the five layers - Eight framework-selection questions and Figure 13.2, Framework Selection Tree - Proof of value for tooling decisions (the seven Charter fields applied to tooling) - Runtime ownership of the tooling stack - Figure 13.3, Runtime Architecture - Vendor and provider risk review list - Figure 13.4, Interoperability Map - Modali Accelerate: A Disclosed Case

Laws developed. The front matter assigns no Law to this chapter, and the chapter names none by number. It restates Law 3's wording verbatim at line 182 (a proof of concept shows that the technology can perform, a proof of value shows that the work improves) and introduces runtime ownership as an expression of the book's third commitment, which Chapter 15 develops under Law 12. Part V's introduction notes that Law 6 is anchored in this Part and Law 8 recurs throughout it.

Discussion questions. 1. The chapter lists nine production questions the prototype cannot answer, from who can see which documents to whether the business owner can tell if the work improved. Pick an AI tool your organization uses and answer them. Which ones cannot be answered, and what does that say about the stack? (*Answerable by a non-engineer executive.*) 2. Why does the chapter say the main model-access decision "gets settled in a vendor demo and contradicted three months later," and what must the stack support so the answer can change every quarter? 3. PolicyAssist is specified in detail at Layers 1 through 3 and thinly at Layers 4 and 5. Why does the chapter say this reflects the ordinary sequence of the work, and what goes wrong when the cross-cutting bands are fitted afterward? 4. The Mid-Atlantic state engagement wrote a policy before selecting tooling, and three

requirements fell out of it. What would a demo-first selection have missed, and how does that reorder a procurement process you know?

Applied assignment. Run a Five-Layer Tooling Stack audit of one AI application in your organization. For each layer, record what is in place, who owns it, and which of the six production questions it can answer. Then apply the eight framework-selection questions to the orchestration framework in use and state whether it exposes or hides the controls the workload requires. Deliver a one-page stack map with a named owner per layer and a runtime-system owner for the whole, following the chapter's ownership rule.

Quotable line. Line 17: *A runtime architecture determines whether the application still works once the organization depends on it.*

Chapter 14: Security, Risk, and Deployment Models

Summary. The chapter opens with the MyCity, MyAI chatbot's twenty-eight months under official city branding and the administration's response of strengthened disclaimers and narrowed scope in place of taking the system offline. Its claim is Law 6: a disclaimer is not a control, and a control prevents, routes, blocks, escalates, logs, or verifies. *Moffatt v. Air Canada* extends the lesson to the private sector. The chapter then shows how AI dissolves the boundary between code, data, and input, maps the AI-specific threat surface (prompt injection, sensitive information disclosure, excessive agency, improper output handling, poisoning, supply-chain risk, unbounded consumption), surveys where regulators have landed across five categories, and compares four deployment models in the Deployment Model Matrix. The Control Architecture places controls at seven boundaries along the runtime path. Incident response is written as six questions answered before launch, and proof of value is extended to include risk boundaries.

Learning objectives. - Distinguish a control from a disclaimer and locate each of the seven boundaries of the Control Architecture on the runtime path. - Compare the four deployment models on data control, operational burden, and best-fit workloads, and explain when self-hosting earns its burden. - Draft the six-question incident response plan for an AI system before it launches.

Frameworks and artifacts. - The runtime-path risk surface (eleven items) - Where Regulators Have Landed: five categories of regulation and guidance - Figure 14.1, AI Threat Surface Map - The AI-specific threat surface: prompt injection, sensitive information disclosure, excessive agency and tool misuse, improper output handling, data and model poisoning, supply-chain and provider risk, unbounded consumption - Four deployment models and the Deployment Model Matrix (Figure 14.2) - The newsroom source-trust routing rule, written against data classification - The Control Architecture and its seven boundaries (Figure 14.3) - Figure 14.4, Secure AI Runtime Architecture - User-scoped retrieval and authorization propagation - Incident response for AI systems: six questions - Proof of value with risk boundaries

Laws developed. Law 6 (a disclaimer is not a control), Law 7 (bounded autonomy), and Law 10 (every AI system needs a written rule under which it will be taken offline). All three are assigned in the front matter and named in the text.

Discussion questions. 1. Take an AI system your organization exposes to customers or the public. List what stands between a wrong answer and the user. Which items on that list communicate risk, and which change what the system can do? (*Answerable by a non-engineer executive.*) 2. The tribunal in *Moffatt v. Air Canada* rejected the argument that the chatbot was a separate legal entity. How does that ruling change the way a deployer should read Law 6, and why does the chapter say the operating lesson survives even though the decision binds no other jurisdiction? 3. The chapter says self-hosting “feels more secure almost by definition” and then argues the feeling attaches to where the model sits and not to how it is operated. Using the newsroom example, explain why the routing rule must be written against the classification of the material the task touches. 4. User-scoped retrieval must be enforced before context enters the prompt. Why is an output filter insufficient, and what does the board-memo example show about how unauthorized knowledge leaks without the document ever being displayed?

Applied assignment. Produce a Control Architecture review for one AI system in your organization. For each of the seven boundaries, record the controls in place, the controls missing, and the owner. Then write the six-question incident response plan for that system, naming the runtime owner, the incident categories, and the containment options. Close with a recommended deployment model from the Deployment Model Matrix and a one-paragraph justification that addresses the regulatory regimes the system falls under. Deliver as a three-page memo.

Quotable line. Line 7: *A control prevents, routes, blocks, escalates, logs, or verifies.*

Chapter 15: Operating the AI System: Runtime Ownership, Cost, and the Operating-Model Discipline

Summary. The chapter opens by observing that operating-model change is the commitment the book has named more than delivered, and that a production AI system stops looking like an architecture diagram after a week and starts looking like a roster. Its claim is that runtime ownership and cost discipline belong together because cost economics are visible only when someone owns them at runtime. It defines the runtime owner's authority over quality, cost, risk, adoption, and improvement, specifies the six artifacts a handoff must transfer, and names two staffing patterns for standing it up. The cost half separates demand from amplification, lays out the seven-layer AI Cost Stack, gives a basic cost model with cost per useful outcome, and works PolicyAssist through three configurations that cut per-query inference and retrieval cost by about 62 percent under the Quality–Cost–Latency Tradeoff Triangle. Nine optimization levers and a ten-field cost governance model close the chapter.

Learning objectives. - Define runtime ownership and list the six artifacts a build-to-operate handoff must transfer. - Separate demand from amplification in an AI system's usage and assign each of the seven Cost Stack layers to fixed, variable, or human-operational cost. - Read the PolicyAssist configurations through the Quality–Cost–Latency Tradeoff Triangle and explain why no optimization ships without an evaluation run.

Frameworks and artifacts. - Runtime ownership: the five areas the owner holds (quality, cost, risk, adoption, continuous improvement) - The handoff: six artifacts transferred from build to operate - Two staffing patterns: role consolidation and runtime-team-of-one - Runtime economics: demand versus amplification - Figure 15.1, Token Flow - The AI Cost Stack, seven layers (Figure 15.2) - Fixed, variable, and human-operational costs - The Basic Cost Model and the cost-per-useful-outcome formula - A Worked Illustrative Example: PolicyAssist at Scale, Configurations A, B, and C - The Quality–Cost–Latency Tradeoff Triangle in operating form - Figure 15.3, Cost, Quality, and Risk in Operation - Six overlooked cost drivers - Nine optimization levers (Figure 15.4) - Waste versus investment in cost optimization - AI Cost Governance, ten fields, and its relation to FinOps - Proof of value and scale economics: the cost half of the evaluate step

Laws developed. Law 12 (operating-model change is the price of AI adoption), assigned in the front matter and named at the chapter's opening. The chapter also names Law 8 when reading the worked example, since the evaluation harness gates every routing and retrieval change.

Discussion questions. 1. The chapter says a system in production looks like a roster: who answers when the dashboard turns red, who changes the routing rule, who tells finance the bill went up. Fill in that roster for an AI system in your organization. Where does it default, and to whom? 2. Why does the chapter distinguish demand from amplification, and what happens to a cost-reduction effort that optimizes the half the team does not control? 3. In the worked example, operations cost stayed flat while inference and retrieval fell by about 62 percent, moving the full bill by roughly 4 percent. What does the chapter say the lesson is about, and why does the 5 percent review sample matter more than the headline saving as volume grows? 4. Human review is called the largest hidden cost and the one

teams leave off the spreadsheet because it sits in someone else's headcount. How would you cost it for a workflow you know, and which optimization lever addresses it?

Applied assignment. Build a Basic Cost Model for one AI application, following the chapter's seven-layer structure and the rule that each charge is counted once, in one layer. Estimate monthly inference cost with the chapter's formula, then state cost per useful outcome and define what counts as a useful outcome. Identify which layers are fixed, variable, and human-operational, and propose two of the nine optimization levers with the quality guardrail each must preserve. Attach a completed ten-field cost governance model naming the budget owner and the runtime owner. Deliver as a spreadsheet and a one-page reading of it.

Quotable line. Line 50: *The pattern is reproducible enough that the question to ask before launch is not "is the system ready?" but "is the runtime owner ready?"*

Chapter 16: Measuring Value and ROI

Summary. The chapter opens with a verbal failure. A team reports four million dollars of value, and the team, the finance officer, and the operations leader each mean a different number. Its claim is that projected, realized, and banked value must never be substituted for one another, and that most programs leak between realized and banked, where no one is accountable for translating operational gain into the cost base. Measurement is framed as the evaluate step of the operating sequence (derive, baseline, develop, evaluate), and a Mid-Atlantic state procurement engagement shows what agreeing two measures before development buys. The chapter introduces the Benefits Chain, four responses when projected and realized diverge, three additions that probabilistic quality requires of measurement, Goodhart's Law as the hazard of any single reported number, the KPI Hierarchy, and the Attribution Challenge, which it insists be settled during the Charter conversation.

Learning objectives. - Distinguish projected, realized, and banked value and explain why the realized-to-banked gap is an operating-model failure. - Trace a Benefits Chain from AI capability to financial value and identify the evidence each link requires. - Design a measurement plan that pairs financial claims with evaluation-harness scores and guards against Goodhart's Law.

Frameworks and artifacts. - Projected, realized, and banked value - The operating sequence: derive, baseline, develop, evaluate - The Mid-Atlantic state procurement case (two measures agreed before development) - The Benefits Chain (Figure 16.1): AI Capability, Operational Impact, Business Outcome, Financial Value - Figure 16.2, Value Waterfall - Three recurring patterns (projected exceeds realized, realized exceeds projected, realized does not convert to banked) - Four responses when projected value does not match realized value - Three additions for probabilistic systems: continuous quality measurement, quality-adjusted value, drift triggers - Goodhart's Law and its defenses - The KPI Hierarchy (Figure 16.3): technical, operational, strategic - The Attribution Challenge and Figure 16.4, Measurement Model

Laws developed. Law 3 (proof of concept versus proof of value), Law 4 (baseline before you build), Law 8 (the evaluation harness is the precondition for everything else), and Law 9 (realized value is not the same as banked value). All four are assigned in the front matter and named in the text.

Discussion questions. 1. Think of an AI or automation initiative in your organization that was declared a success. Which of the three numbers was being reported, and what would the other two have shown? 2. The DBS Bank figure of over SGD 1 billion is described as a projected number published as one. Why does the chapter say a well-run program does not escape the three-number problem, and what distinguishes it from a poorly run one? 3. The state procurement program has seen no assessment returned under the new process and still declines to publish a rate. What does that restraint protect, and why does the chapter say two measures that pull against each other are close to impossible to add afterward? 4. The chapter says attribution should be chosen during the Charter conversation, before anyone has a result to protect. In the state case, what design choice would have separated the policy's effect from the system's, and what would it have cost?

Applied assignment. Take one AI initiative your organization has run or proposed and complete the back half of the AI Proof-of-Value Charter for it. Map its Benefits Chain with the evidence each link

requires, state projected value with visible assumptions, and report realized and banked value separately, or explain why each cannot yet be stated. Add the paired quality envelope from the evaluation harness and one drift trigger. Name the attribution method and a second measure that pulls against the headline number. Deliver a two-page value review.

Quotable line. Line 5: *Substitute one for another in a review and the program stops being an investment and becomes an anecdote.*

Chapter 17: Change Management and Adoption

Summary. The chapter opens with the observation that a capable system nobody uses delivers nothing, and a confidently wrong one delivers worse than nothing by spending down institutional trust. Its claim is that adoption is the operating mechanism by which measured value becomes realized value, and that most failed programs fail at adoption and not at the model. It introduces the four-phase Trust Curve, the Adoption Barrier Model (skill, trust, process, incentive) with the interventions that work and those that do not, and a Responsibility Matrix and Change Enablement Flow for running adoption as a loop. Morgan Stanley's assistant shows adoption designed in. A New York City procurement program shows a champions network spending borrowed trust on a form that overrode a working workaround. Workforce Transition sets four questions executives owe the people whose roles change, and MyCity is revisited as trust regression in three patterns.

Learning objectives. - Diagnose which of the four adoption barriers is stalling a rollout from its symptoms and select the intervention pattern the chapter says works. - Explain why the Trust Curve's phases cannot be skipped and what trust regression looks like when they are. - Answer the four workforce-transition questions for a role that AI is decomposing.

Frameworks and artifacts. - The Trust Curve (Figure 17.1): Demonstration, Supervised Use, Calibrated Reliance, Mature Partnership - Double-loop learning (Argyris and Schön) as the discipline behind the curve - The Adoption Barrier Model (Figure 17.2): skill, trust, process, incentive - The Responsibility Matrix for AI Change (Figure 17.3) - Figure 17.4, Change Enablement Flow - Operating mechanisms against each barrier (what works, what does not) - Case: Morgan Stanley's AI Assistant Adoption Journey - Case: A Change Organization That Worked, and What It Spent - Workforce Transition: the four questions executives owe the workforce - A Note on Regulated-Industry Decision Rights - Trust Regression: authoritative-voice failure, disclaimer-as-shield collapse, visible-accountability vacuum

Laws developed. Law 6 (a disclaimer is not a control), Law 11 (authority should match accuracy), and Law 12 (operating-model change is the price of adoption). All three are assigned in the front matter and named in the text.

Discussion questions. 1. Pick a tool your organization rolled out that people work around. Using the barrier table, which symptom do you see, and which barrier does it point to? What intervention did the organization try, and which column of the table does it fall in? 2. The New York City program documented the phone call to the procurement analyst and designed a required-field form anyway. Why does the chapter say a workaround that survives the system built to replace it was usually doing real work, and what does "borrowed trust" mean for how a champions network should be spent? 3. The chapter says bonus payments for AI use do not work while changes to how work is measured do. Why does one accumulate into a norm and the other not, and what does the billing-by-the-hour example show about incentives that penalize AI use? 4. In regulated industries decision rights push up rather than down. How does that change the way the four-barrier model applies to a bank or hospital, and what parallel framework does the chapter say must sit beside it?

Applied assignment. For one AI deployment in your organization, past or planned, produce an adoption design. Assess the four barriers for the specific user population, propose one intervention per barrier drawn from the chapter’s “what works” lists, and build a Responsibility Matrix that names who is accountable for each. Then choose one role the system decomposes and write the answers to the four workforce-transition questions, including the new task portfolio and who funds retraining. Deliver as a three-page plan with the Change Enablement Flow’s monitoring signals specified.

Quotable line. Line 70: *Trust is calibrated through experience, not declared by management.*

Chapter 18: Future-Proofing and the Next Wave

Summary. The chapter opens with the admission that any specific prediction about which provider or architecture dominates in three years is a guess, and asks instead which investments pay off no matter which model wins. It synthesizes the book around three commitments (the context layer, proof of value, operating-model change) with the Maturity Model as the diagnostic, and argues that each of the three compounds while prompts, vendor-specific integrations, and roadmap predictions decay. A 2026 reading of Agrawal, Gans, and Goldfarb's system-solution argument yields three buyer-side conclusions, including that agentic systems are the next decoupling. The chapter then examines how provider economics reach the buyer through five risks and four structural responses, sets bounded autonomy underneath the reliability line, describes distributed intelligence across model tiers, names four constraints that temper the trajectory, and defines the four dimensions of future-readiness.

Learning objectives. - Explain why the context layer, proof-of-value discipline, and operating-model change compound while other AI spending depreciates. - Identify the five provider risks a buyer inherits and match each to its structural response. - Score an organization on the four dimensions of future-readiness and state which remediation the pattern calls for.

Frameworks and artifacts. - The three commitments, with the AI Maturity Model as diagnostic - Schumpeter's creative destruction and Perez's installation and deployment phases - Three things that compound: the context layer, Proof-of-Value discipline, operating-model change read through Maturity Model progression - A 2026 assessment of system solutions (Agrawal, Gans, and Goldfarb) - Three buyer-side conclusions, including the Bullwhip Effect - Five provider risks: model deprecation, capacity rationing, price moves, terms-of-service shifts, training-data litigation - Four buyer-side moves: architectural abstraction, cost-model conservatism, contractual posture, strategic posture - Figure 18.1, The Bounded-Autonomy Trajectory - Figure 18.2, Distributed Intelligence Map, and Figure 18.3, Edge + Cloud Model - Four capability constraints that temper the trajectory - The Four Dimensions of Future-Readiness and the Future-Readiness Scorecard - Modali Accelerate as an abstraction-layer example - Figure 18.4, Capability Maturity Snapshot (a snapshot, not a framework)

Laws developed. Law 1 (models are the smallest part of enterprise AI advantage), Law 2 (the context layer does not depreciate), Law 11 (authority should match accuracy), and Law 12 (operating-model change is the price). All four are assigned in the front matter and named in the text. The chapter also restates Law 3 alongside Law 2 when naming what compounds.

Discussion questions. 1. The chapter says the strategic question is what to build now that will still be valuable when the predictions change. For your organization, which current AI spending sits in the compounding column and which in the depreciating one? 2. Why does the chapter argue that architecture should assume prediction-layer commoditization rather than protect against it, and what does that imply for a roadmap that treats the proprietary model as the asset? 3. The bullwhip example shows a retailer's better forecast amplifying volatility down the chain. What transmits the volatility, according to the chapter, and why does it say the advantage goes to industries that rebuild contracting and information-sharing first? 4. The capability-side and reliability-side curves diverge, and the gap is

capability the organization is not yet entitled to operate autonomously. What sits on the reliability side, and which of those does a better model relieve?

Applied assignment. Build a Future-Readiness Scorecard for your organization. Define rating criteria for each of the four dimensions (architectural modularity, data readiness, organizational agility, strategic clarity) from operating evidence you can point to, score each, and state which remediation pattern the profile calls for. Then complete the Twelve Laws Self-Assessment for the same organization and reconcile the two readouts in a one-page memo: where they agree, where they diverge, and which two Laws you would invest in first before the next model generation arrives.

Quotable line. Line 101: *Bounded autonomy lives underneath the reliability line, not in the space above it.*

Prologue and Epilogue

Prologue: After the Demo. Teach the Prologue as the book’s opening problem. The executive’s second question, who decides when the answer changes, is the one the whole book answers, and students should hold it while reading every chapter. The Miami stage in 2023 gives the author’s own starting point and explains why “focus on your data” was the unfashionable answer in the room. Paul David’s dynamo analogy carries the argument that the technology was the easy part and the factory the hard one. Ask students to name the three commitments (context layer, proof of value, operating model) before they read Part I, and to test each later chapter against them.

Discussion questions: 1. The team had a good answer to “how do we know this is right?” and no answer to “who decides when the answer changes?” Why does the second question cut in a different direction, and what would an organization need to have built to answer it? 2. Factories that hooked the old overhead shaft to a big electric motor “looked modernized.” What is the equivalent move in enterprise AI, and how would you recognize it in an organization you know?

Epilogue: The Decade That Has Already Started. Teach the Epilogue as the Prologue’s question answered in operational terms: a context layer someone owns, a signed Charter with a scale-or-stop rule, a harness running on every change, a shutdown record written before launch, a value review that keeps three numbers apart. Its five bolded claims work well as a closing seminar. Assign each to a group to defend or attack with evidence from the chapters. The senior leader’s two sentences about betting on what still matters after the models change make a strong final prompt. Pair the Epilogue with the Twelve Laws Self-Assessment so students leave with a scored readout.

Discussion questions: 1. The Epilogue says disciplines are absorbed by doing the work and that nobody has ever absorbed one from a demonstration. Which of the five bolded claims does your organization already act on, and which is it still buying? 2. “Trust is the slowest variable in the system and the most expensive when it fails.” Trace that claim back through Chapters 14 and 17. What would publishing acceptance thresholds before an incident require of an institution you know?

Technical Appendix

The Technical Appendix, subtitled Implementation Mechanics, holds the level beneath Chapters 10 through 12. It covers byte-pair encoding and token inflation, cosine similarity, decoding parameters, transformer generation, latency mechanics, the six hallucination types, in-context learning and chain-of-thought, the Five-Component Prompt Template, the Prompt Iteration Loop, chunking, embedding selection, ANN indexing, query rewriting, RAGAS and LLM-as-judge calibration, the RAG Evaluation Scorecard, and a worked ReAct trace. Assign it in engineering-track and technical-leadership courses. Executive and policy programs can treat it as reference.

Course maps for *Operating Intelligence*

Three ways to teach the book, from a full graduate term to a single workshop. Chapter numbers refer to the first edition (Silent Way Press, 2026). The three companion artifacts (Twelve Laws Self-Assessment, AI Proof-of-Value Charter, Context Layer Diagnostic) are free at operatingintelligence.org and can be assigned as graded deliverables.

A. Twelve-week graduate seminar: Enterprise AI Strategy, Governance, and Operations

For MBA, MPA, MPP, and MS programs. One three-hour session a week, or two ninety-minute sessions. Reading load is 40 to 60 pages a week.

Week	Theme	Read	In class	Deliverable
1	Why this wave is different	Prologue, Chapter 1	The dynamo, the demo, and the two questions the executive asks	Short memo: which Law your organization violates most
2	Executive judgment and maturity	Chapters 2, 3	Operating intelligence defined, the AI Maturity Model	Twelve Laws Self-Assessment, scored for one organization
3	Data readiness and the context layer	Chapters 4, 5	What “AI can use our knowledge” requires	Context Layer Diagnostic, section by section
4	Governance and the MyCity case	Chapter 6	Case discussion: a disclaimer is not a control, the written offline rule	Case brief: three controls the city could have shipped
5	Vision and use-case selection	Chapters 7, 8	The AI North Star, the five traps, the media operator’s point solution	Use-case scoring for three candidate initiatives
6		Chapter 9	Charter fields, the exit gate, scale-or-stop	One-page AI Proof-of-Value Charter

Week	Theme	Read	In class	Deliverable
	Roadmap and the proof-of-value charter			
7	The technology, at the level leaders need	Chapters 10, 11	Models, prompts as production engineering, retrieval as a delivery mechanism	LLM Design Decision Sheet for the Week 6 charter
8	Agents and bounded autonomy	Chapter 12	Permissions, reversibility, escalation	Tool-permission table for one agent workflow
9	Tooling and control architecture	Chapters 13, 14	The stack, the seven boundaries, Moffatt v. Air Canada	Control map for the Week 6 charter
10	Operating the system	Chapter 15	Runtime ownership, cost, the operating-model discipline	Runtime owner charter and cost model
11	Value and adoption	Chapters 16, 17	Projected, realized, banked, the four adoption barriers	Value review of the Week 6 charter
12	Future-proofing	Chapter 18, Epilogue	Readiness, not prediction	Final: board memo recommending scale, redesign, pause, or stop

Assessment: weekly deliverables (40%), the charter and its Week 11 value review (30%), the final board memo (30%). The Technical Appendix is optional for policy students and assigned for MS students in Weeks 7 to 9.

B. Six-session executive module

For certificate programs and executive education, two and a half hours a session, with participants who run or fund AI programs. Pre-reading is 60 to 90 pages a session.

Session	Read	The question participants leave with
1. The inflection and the executive's job	Prologue, Chapters 1 to 3	Where is our program on the Maturity Model, and which Laws are we skipping?
2. Context and governance	Chapters 4 to 6	Do we have a named, governed context layer, and what is our written offline rule?
3. Vision to charter	Chapters 7 to 9	Which of our pilots is a proof of value, and which is a demo?
4. What leaders need to know about the technology	Chapters 10 to 12	What are we authorizing an agent to do, and what can it undo?
5. Running it	Chapters 13 to 15	Who owns the runtime, and what does it cost per outcome?

Session	Read	The question participants leave with
6. Counting it	Chapters 16 to 18	What has been banked, and what has only been realized?

Each session opens with the Twelve Laws Self-Assessment scores participants bring from their own organizations, and closes with a field on the AI Proof-of-Value Charter completed for a live initiative. Participants leave Session 6 with a signed charter.

C. Three-hour workshop: Designing AI Governance and Organizational Strategy

The author's eCornell workshop format. Thirty to sixty senior leaders, four exercises against a single fictional case organization, one Proof-of-Value Framework per group to take home. Pre-reading: the Twelve Laws, Chapter 6 (the free sample chapter), and Chapter 9. Facilitator materials are available on request.

Where the book fits an existing course

- **AI strategy or digital transformation** (business school): Parts I, III, and VI carry the course. Assign Chapter 6 as the governance case.
- **AI governance, ethics, or public policy**: Chapters 6, 14, 17, and 18, with the Prologue. The MyCity record is public and complete, and the author's Cornell teaching case (PUBPOL 5019) can accompany it.
- **Public procurement and government technology**: Chapters 6, 8, 9, 14, and 15. The author was CTO of the New York City Mayor's Office of Contract Services from 2016 to 2022.
- **Technical MS programs** (data engineering, NLP, information systems): Part IV with the Technical Appendix, plus Chapters 15 and 16 so builders can measure what they ship.