

EBOOK

The AI-Ready Engine

How to Build Teams that Learn and Ship 10× Faster



CREATED BY

Marc Boudria, Chief Innovation Officer
at [BetterEngineer.com](https://www.betterengineer.com)

betterengineer

Table of Contents

Preface

Changing Fast Without Losing Who You Arep. 3

From “AI-Ready” to “AI-Throughput”

Why Small Teams Are Winningp. 4

Where You Really Stand

The AI Maturity Model & Scorecardp. 11

The Building Blocks of an AI-Ready Team

People, Process, Platform, Policyp. 17

Hiring AI-Ready Talent

Prediction over Pedigreep. 21

Designing AI-Ready Teams

How to Organize the Workp. 27

Building an AI-First Culture

Courage with Controlsp. 34

Your First 6–12 Months

A Practical Roadmapp. 37

Keep Evolving

Staying AI-Ready Over Timep. 45

Bringing It All Together

And How BetterEngineer Can Helpp. 47

The practical kit for shipping AI safely

AI-Ready Templates.....p. 54

Preface

Changing Fast Without Losing Who You Are

If you want to harness modern, AI-enabled workflows, you must begin with an uncomfortable truth: your organization needs to look inward with honesty and discipline. That means understanding the real challenges your teams face, what your customers actually need, and what the business must become in order to grow and remain competitive. The future demands change; remaining static is not an option.

The central question, then, is not whether to change, but how to change without losing what is essential to your organization's identity. There is no universal answer. Every company will take a different path because identity, risk appetite, and constraints differ. Leaders should be cautious of anyone promising a single solution, particularly platforms that suggest you can "solve AI" by placing all of your data into a retrieval-augmented generation (RAG) system and assigning an agent to it. This is simply the latest expression of the "one-size-fits-all-that-fits-none" syndrome.

A more durable path is both simpler and more demanding. It requires managing your knowledge in a way that is independent of any specific platform. Provenance, access, and disciplined refresh cycles must become part of how you work every day. That is how you remain sovereign: able to change tools, switch vendors, and adapt your architecture while preserving your standards and your sense of self.

This book sets realistic guardrails for that journey. It is designed to help you:

- Assess your current state: readiness, risks, and early wins.
- Organize people and cadence so that experiments actually close and value ships.
- Build light controls—courage combined with controls—so speed and safety rise together.
- Treat knowledge as infrastructure, rather than unstructured content buried in a chat index.
- Design a 6–12 month roadmap that creates momentum without mortgaging the future.

You will not find "silver bullets" in the chapters that follow. Instead, [you will encounter patterns: pragmatic ways to change, to learn, and to adapt, without losing the thread of who you are.](#)

CHAPTER 1



From "AI-Ready" to "AI-Throughput"

Why Small Teams Are Winning



From “AI-Ready” to “AI-Throughput” – Why Small Teams Are Winning

Why small, well-run teams are outpacing bigger budgets

Over the last year, we have repeatedly seen a striking contrast. On one side, small groups of five to eight engineers ship credible AI-enabled features at a steady pace. On the other, organizations with dozens of engineers struggle to move beyond prototypes and slideware. The difference is rarely in credentials or tools. It is in learning throughput: how quickly a team can move from an idea to a tested change in the real world, absorb the result, and try again. [1]

When throughput is high, even a modest team compounds its advantage week after week. Learning accumulates. Practices improve. Confidence grows. When throughput is low, adding more people and more tools does not solve the problem; it simply multiplies indecision, coordination cost, and frustration.

Hiring Data Scientists Alone Will Not Make You “AI-Ready”

When AI first rises to the top of the agenda, one common response sounds reassuringly decisive: “Let us hire a few data scientists, and we will be AI-ready.”

In practice, this approach often produces expensive prototypes and very limited product impact. The pattern underneath is usually the same:

- There is no knowledge fabric. Teams cannot reliably find or trust the data and knowledge they need to run experiments safely.
- There are no evaluation habits. Teams lack a shared definition of what “good” looks like, so experiments do not truly conclude.
- There is no product operations function for AI. Features are launched without sufficient telemetry, rollback plans, or clear decision owners.
- There is a culture mismatch. Capable people hesitate to ship when informed, reversible failures are treated as liabilities rather than contributions.

Hiring and expertise matters, but without the right system around that talent, new hires cannot meaningfully increase your shipping tempo. The rest of this book is about building that system.

[1] Nicole Forsgren, Jez Humble, and Gene Kim, *Accelerate: The Science of Lean Software and DevOps* (Portland, OR: IT Revolution Press, 2018).



From “AI-Ready” to “AI-Throughput” – Why Small Teams Are Winning

What We Mean by “Throughput”

In this book, AI-throughput is the weekly rate at which your company converts hypotheses into shipped, reversible decisions backed by evidence.[2] Put more simply, it is the frequency with which your teams turn ideas into concrete changes in the product, learn from what happens, and can easily undo those changes when necessary.

Throughput is not merely speed. It is speed with closure. Each experiment ends in a decision—ship, shelve, or iterate—and a concise record that others can learn from. Throughput increases when you run smaller, well-scoped bets, evaluate outcomes against clear rubrics, and make rollback straightforward and low-risk.

The Throughput Flywheel: Ship, Learn, Repeat

One useful way to visualize this is to imagine your organization as a flywheel with five blades. Each rotation delivers learning; each blade can be tuned by leadership.

- Small bets. Ideas are framed as minimal, testable changes: a narrow workflow, a single user segment, or a short shadow-mode trial.
- Quick evaluations. Criteria for success are defined in advance—acceptance tests, guardrail checks, cost and latency targets, and a simple human spot-check.
- Weekly demos that end in decisions. Each week, teams show working behavior. Every demo ends with a clear call—ship, shelve, or iterate—captured in a one-page Decision Log.
- Ship wins or roll back cleanly. Wins are shipped; missteps are rolled back safely with a feature toggle or version pin. In both cases, a brief note records what was tried, what was learned, and what will happen next.
- Logs that fuel the next bet. Those notes feed a shared knowledge hub, so the next team begins from a more advanced starting point rather than repeating the same exploration.[3]

Leaders accelerate this flywheel not by demanding “more innovation,” but by removing friction: unclear roles, scope creep, missing data, excessive fine-tuning, or extended debate over minor details before success has even been defined.

[2] Eric Ries, *The Lean Startup: How Today’s Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses* (New York: Crown Business, 2011).

[3] John Boyd, “*The Essence of Winning and Losing*,” unpublished briefing, 1995.



From “AI-Ready” to “AI-Throughput” – Why Small Teams Are Winning

The Five Pillars Leaders Can Control

Throughput is not an accident. It is the result of deliberate choices across a small set of levers. We will explore each of these pillars in later chapters, but at a high level, they form the “layer cake” leadership can actually control.

People – Curiosity With Safety

The first pillar is the mindset of the people doing the work. Hiring and rewards should favor:

- Curiosity loops (ask → try → reflect)
- Bias awareness (the ability to explain where a result might be wrong)
- Psychological safety (a willingness to express uncertainty early)

Teams ship meaningful work when they trust that informed, reversible failures will be treated as contributions to learning, rather than as personal failures.

Process – Decisions Over Meetings

The second pillar is process. Many organizations rely on speculative meetings—discussions about ideas that have not yet been tested. A throughput-oriented organization replaces much of that speculation with weekly demos and a concise Decision Log.

Work is structured so that it passes naturally through a small number of checkpoints:

- Hypothesis
- Test
- Result
- Decision
- Named owner

Cycles are kept short, typically one to two weeks. If an initiative cannot be evaluated meaningfully within that window, the scope is likely too large and should be sliced further.

Platform – A Simple, Explainable Foundation

The third pillar is platform. At minimum, you need:

- A shared knowledge hub, where prior experiments, prompts, patterns, and pitfalls are recorded
- Safe access to the data required for experimentation
- A “right-tool-for-the-job” menu of models and services

In some cases, the right choice may be a large hosted model. In others, an open-source model or a smaller, on-device model is sufficient. The primary responsibility of leadership is not to chase every new acronym, but to create clarity about options and constraints so that teams do not stall in tooling debates.



From “AI-Ready” to “AI-Throughput” – Why Small Teams Are Winning

Policy – Light Guardrails That Enable Speed

The fourth pillar is policy. Here, a short document can go a long way. A single page of Greenlight Rules often delivers most of the benefit. It should define:

- What data you will never use
- When and how AI-assisted content must be labeled
- What must be logged for each experiment or feature
- Who can approve exposure to real customers

Well-designed guardrails increase confidence to run more experiments. Poorly designed ones become red tape that prevents them. The goal is always the former.

Product Operations for AI – Minimal Habits to Ship Safely

The fifth pillar is product operations for AI. You do not need a research lab to operate AI responsibly, but you do need a small set of consistent habits:

- Define “good” in advance. Each experiment has a brief acceptance checklist (quality, safety, cost, and latency).
- Monitor with basic telemetry. You know how often a feature is used, whether users complete tasks more quickly, and whether cost or error rates spike.
- Make reversals inexpensive. Feature flags, version pins, and a rollback runbook make it straightforward to return to a known-good state.

These habits make informed failure inexpensive and AI-throughput high.[4]

[4] Gene Kim, Jez Humble, Patrick Debois, and John Willis, *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*, 2nd ed. (Portland, OR: IT Revolution Press, 2021).



From “AI-Ready” to “AI-Throughput” – Why Small Teams Are Winning

Metrics That Matter

Throughput is easier to improve when you can see it. Rather than building a complex dashboard, start with a small set of measures that fit on a single page. Select one or two to begin; expand as your organizational capability grows.

- Cycle time to insight. The number of days from idea to evidence.
- Experiment velocity. The number of closed experiments per week, each with a logged decision.
- Percentage of decisions augmented by AI. The share of key product or operations decisions that used AI-assisted analysis or generation.
- MTTR-AI (mean time to recovery for AI features). How quickly you can roll back or remediate a degraded, model-driven feature.
- Ritual adoption. The percentage of teams consistently using weekly demos and Decision Logs.

Definitions:

- A closed experiment has a pre-stated goal, an evaluation, and a captured decision.
- An augmented decision is a decision where AI materially informed options or execution, and its use was acknowledged in the log.
- A rollback is a reversible change, such as a feature flag or version pin, that restores prior behavior in minutes rather than days.

What Changes When You Optimize for Throughput

When an organization takes throughput seriously, several things begin to change.

Leaders and teams spend less time debating hypotheticals; the next demo becomes the forum for reality. Hiring decisions become clearer, because you are looking for people who can define a test, select an appropriate tool, and evaluate results, not only talk persuasively about ideas.

Culturally, the focus shifts from fear to craftsmanship. A well-logged, reversible failure is treated as a contribution; it improves the collective understanding of what works and what does not. Roadmaps become lighter and more accurate. Frequent, smaller bets with visible outcomes consistently outperform large, infrequent reveals that arrive late and over-promised.

Cost discipline improves as well. Teams choose appropriately sized tools for a defined job, instrument usage, and retire what does not move a key performance indicator. The result is not only better AI features, but a healthier, more resilient engineering organization.



From “AI-Ready” to “AI-Throughput” – Why Small Teams Are Winning

Practical Moves for the Next Two Weeks

If you lead a team or an organization, you do not need to wait for a major reorganization to begin. Over the next two weeks, you can:

1. Select one workflow. For example: support ticket summarization, drafting a customer email, or generating QA test cases. Define a two-week experiment around it.
2. Adopt the Decision Log. Use the template referenced in our last chapter “AI-Ready Templates”, and ensure that every demo ends with a clear decision—ship, shelve, or iterate—plus a short paragraph of notes.
3. Choose one metric. Select a single KPI from the metrics list above and measure it weekly.
4. Create a visible safety net. Publish your one-page Greenlight Rules and make it explicit that there will be no blame for reversible failures that are logged.
5. Schedule a weekly demo hour. Reserve a 45-minute demo session each week and invite key stakeholders. Narrated demos, with clear decisions, are more valuable than slide presentations.

How This Connects to the Rest of the Book

The chapters that follow build on this foundation.

- In Chapter 2 (Maturity Model & Scorecard), you will assess where throughput currently stalls and which pillar requires attention first.
- Chapters 3–6 provide concrete playbooks for People, Process, Platform, Policy, and Culture, so you can raise throughput without being overwhelmed by tooling choices.
- Chapter 7 sequences quick wins, medium bets, and longer-term initiatives so that you can build momentum without overcommitting.

CHAPTER 2



Where You **Really** **Stand**

The AI Maturity Model & Scorecard

2

Where You **Really Stand** The AI Maturity Model & Scorecard

Establishing a Baseline You Can Act On

Most organizations carry two competing stories about their AI posture: “we are behind” and “we are experimenting all over the place.” Both can be true. Neither is specific enough to guide the next move.

This chapter gives you a simple, honest baseline so you can stop trying to address everything at once, and instead focus on the few constraints that are throttling learning throughput.

Why Use a Maturity Model?

Resources are finite. A clear snapshot of where you stand allows you to direct energy where it multiplies speed: toward the habit, tool, or ritual that, if improved, will unlock the largest number of stalled experiments.

You can think of this maturity model as triage for throughput. It is not a vanity score; it is a way to identify the weak links in how your organization learns.

The Five Levels of AI-Readiness

This model describes five levels of AI-readiness. The intent is not to create a new label for your company, but to provide a shared vocabulary for where you are today and where you might go next.

Level 1 - Ad Hoc

Individuals tinker. There is no shared place for learning, no evaluation rules, and experiments rarely close. Wins are incidental, not the result of a system.

Level 2 - Pilot-Curious

A few small demos are repeated. Some basic rules exist, but logging is inconsistent and experiments remain open without a clear decision. Leaders ask for results, but there is no weekly rhythm.

Level 3 - Programmatic

Weekly demos occur, decisions are logged, and basic metrics are reviewed. One or two AI-enabled wins run in production with clear owners and rollback plans. Momentum is visible.

Level 4 - Platformized

Teams share a lightweight knowledge hub and reuse patterns. Product operations basics exist for AI features. Experiments and productionization run at two different speeds across multiple pods.

Level 5 - Perpetual Learning

The organization runs a portfolio of bets with strong evaluation habits, fast rollback, cost discipline, and a culture that shares patterns with new teams. Learning throughput becomes part of the organization's identity, not a one-off initiative.

You do not need to reach Level 5 to create value. Most companies see outsized gains simply by moving from Level 2 to Level 3, and then selectively from Level 3 to Level 4 on their highest-leverage workflows.

2

Where You **Really Stand** The AI Maturity Model & Scorecard

The Six Readiness Dimensions (Scored 0–3)

To turn this model into a practical tool, we score six dimensions that correlate strongly with throughput. Each dimension receives a score from 0 to 3 and a one-line note. The total score is less important than the shape of the profile, and in particular, the lowest two dimensions. Those are your bottlenecks.

Scoring rubric (0–3)

- 0 = **absent**
- 1 = **emerging and inconsistent**
- 2 = **consistent in one area or pod**
- 3 = **consistent across teams with reuse**

Scores are for focus, not for pride. An honest 0 is more valuable than a flattering 2.

Readiness Dimensions

- People (mindsets).** Do team members run curiosity loops, surface potential bias risks, and narrate uncertainty early?
- Process (cadence).** Do you have weekly demos, a Decision Log, and small, testable bets?
- Platform (foundation).** Is there a shared knowledge hub, safe data access, and a simple “right-tool” menu (for example, large API, open-source model, or small/on-device model) explained in plain English?
- Policy (guardrails).** Do you publish Greenlight Rules, know who approves what, and keep an audit trail without slowing work?
- Product Ops for AI.** Are minimal evaluation habits, basic telemetry, and reversibility (feature flags, version pins) in place?
- Culture (behaviors).** Are reversible failures with logs recognized as learning, are demos public, and is learning visible (for example, showcase cadence or “most instructive failure” highlights)?

2

Where You Really Stand The AI Maturity Model & Scorecard

What a 0–3 looks like

	People	Process	Platform	Policy	Product	Culture
0	Individuals hide uncertainty.	Brainstorms and slides, no demos.	Findings live in chat threads.	Fear or confusion about what is allowed.	“Looks good” is the only acceptance criterion.	Failure is quiet and career-risky.
1	A few people speak up; it is personality-dependent.	Demos happen ad hoc; decisions are often deferred.	A wiki exists but is rarely used; data access requires heroic effort.	Draft rules exist but are not referenced.	Some experiments define success after the fact.	Individuals share learnings in private channels.
2	Teams regularly narrate assumptions in demos.	Weekly demos and a Decision Log in one group.	A basic knowledge hub; two or three common patterns are reused.	A one-page Greenlight Rules document is used in demos.	Small evaluation checklists and basic telemetry in one pod.	Teams present demos openly; post-mortems are blameless.
3	Managers coach curiosity loops; bias checks are normalized.	Company rhythm: short cycles, closed experiments, visible decisions.	Most bets start from prior patterns; access is self-serve and safe.	Rules are part of onboarding; audits are lightweight and routine.	Evaluation and rollback runbooks are standard; MTTR-AI is tracked.	Company showcases wins and instructive failures on a regular cadence.

From Scores to Decisions: The Prioritization Matrix

Once you have scored each dimension, list your two lowest-scoring areas. For each, rate:

- Impact. How much would improving this dimension increase the number of closed experiments?
- Effort. How difficult is it to improve (time, cost, political capital)?
- Risk if ignored. What breaks if this dimension remains weak (for example, security, intellectual property, customer trust, or wasted spend)?

Choose one high-impact, low-effort move for the next 30 days. Defer perfection. Throughput improves when small frictions disappear.

2

Where You **Really Stand** The AI Maturity Model & Scorecard

Your 30/60/90 From a Cold Start

The following timeline illustrates what progress can look like over the first 90 days.

Day 0–30

- Publish Greenlight Rules.
- Start weekly demos.
- Adopt the Decision Log.
- Close at least two experiments (each with a ship/shelve/iterate decision and notes).

Day 31–60

- Stand up a simple knowledge hub (a shared folder and index page is enough).
- Run four closed experiments.
- Log decisions.
- Roll back one feature once to prove reversibility.

Day 61–90

- Productionize one or two wins behind a feature flag.
- Start tracking, on a single page:
 - Cycle time to insight
 - Experiment velocity
 - MTTR-AI for at least one feature

Short Vignettes: What Progression Looks Like

These brief examples illustrate what movement between levels can look like in practice.

Level 2 → Level 3: A Regional B2B SaaS Team

- Before. Ad hoc demos, frequent “we will decide next sprint,” and no record of outcomes.
- After 60 days. Weekly demos with a Decision Log. Four closed experiments. One feature shipped with a rollback plan. Cycle time to insight dropped from 21 days to 7; the sales team now uses two AI-assisted briefs each week.

Level 3 → Level 4: A Consumer Marketplace Pod

- Before. Strong cadence in one pod, but other teams repeatedly reinvent the wheel.
- After 90 days. A lightweight knowledge hub captures prompts, patterns, and post-mortems. Two-speed delivery is formalized: experiments run in a sandbox lane, productionization in a hardening lane. MTTR-AI fell from hours to minutes; two additional pods reused the same acceptance rubric to ship copy-assist features safely.

2

Where You **Really Stand** The AI Maturity Model & Scorecard

How to Run the Scorecard

- **Who participates.** One or two leaders plus the people doing the work. Avoid a purely top-down, self-congratulatory session.
- **Timebox.** Allocate 45–60 minutes. Score each dimension, discuss the two lowest, complete the Prioritization Matrix, and commit to one 30-day move.
- **Artifacts.** Print the Scorecard (six rows × 0–3), the rubric descriptors, and the Prioritization Matrix. Capture a photo or scan and add it to the knowledge hub.

Bridge to what's next

Use your score to choose which pillar to address first in Chapter 3 (Building Blocks)—People, Process, Platform, Policy, Product Operations, or Culture—and then map those changes into Chapter 7 (Your 6–12-Month Roadmap). The goal is not to achieve a perfect score. The goal is to identify the specific constraints that are currently limiting your AI-throughput, and to remove them deliberately.

CHAPTER 3



The Building Blocks of an **AI-Ready Team**

People, Process, Platform, Policy

3

The Building Blocks of an **AI-Ready Team** People, Process, Platform, Policy

You do not need a research laboratory to make AI useful. What you need are a small number of clear levers—people, process, platform, and policy—pulled consistently over time. When these elements are aligned, your team’s learning throughput rises even if budgets remain flat. When they are neglected, you accumulate half-finished pilots, fragmented knowledge, and a skeptical team that becomes reluctant to try again.

This chapter examines each of these levers in turn.

1. People (Mindsets and Safety)

High-performing AI teams are not defined solely by who knows the latest framework or acronym. They are defined by how people behave under uncertainty.

Three observable behaviors matter most:

1. Curiosity loops. Effective team members naturally move through a pattern of ask → try → reflect → adjust. In interviews, prioritize work samples where candidates describe an iteration, not only a final success. In one-to-one conversations, a simple question—“What did you change after the last demo?”—often reveals how they learn.
2. Bias awareness. In AI-enabled systems, the ability to articulate where results might be wrong, misleading, or unfair is critical. Make “What could be off?” a standing prompt in reviews and design discussions. You are looking for engineers and leaders who can reason about failure modes, not only celebrate impressive outputs.
3. Narrated uncertainty. Teams move faster and more safely when people are rewarded for saying, “I do not know yet.” As a leader, you model this. Treat early flags and uncertainty as contributions, not as weaknesses. Over time, this builds the psychological safety required for honest evaluation and faster iteration.

2. Process (Cadence and Closure)

Throughput collapses when work sits between meetings. The goal is to establish short, predictable cycles that end in concrete decisions, not in additional slides.

Weekly Demos With Real Decisions

Begin with a recurring 45-minute block. Each presenter shows something running, no matter how small, and concludes with a decision: ship, shelve, or iterate. Capture this in a Decision Log that records context, options considered, the decision, the owner, and a revisit date. That single page is what converts activity into visible progress. [5]

[5] Eric Ries, *The Lean Startup: How Today’s Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses* (New York: Crown Business, 2011); Nicole Forsgren, Jez Humble, and Gene Kim, *Accelerate: The Science of Lean Software and DevOps* (Portland, OR: IT Revolution Press, 2018).

3

The Building Blocks of an **AI-Ready Team** People, Process, Platform, Policy

Two-Week Experiments by Default

As a default, design experiments that can be evaluated within two weeks. If a slice of work cannot be meaningfully evaluated in that timeframe, it is probably too large and should be divided. Requests for “more data” or “more time” are acceptable, but they should be logged with a specific next evaluation date.

Micro-Rituals That Accumulate

Small, consistent rituals help sustain this cadence:

- Weekly recap (approximately 15 minutes). The team lead posts three short bullets in the knowledge hub: what shipped, what was learned, and what is next.
- Monthly experiment review. Once a month, review the last set of Decision Logs. Which bets paid off? Which stalled? Which patterns are worth reusing?
- Clear “done” for experiments. An experiment is only “done” when you have a result—positive, negative, or mixed—and a recorded decision. A polished demo, on its own, is not sufficient.

Artifact note

A sample Decision Log template is provided in the Templates section.

3. Platform (Simple and Explainable)

When we talk about “platform” in this context, we are not referring to a complex architecture diagram. For most organizations, it is more helpful to think in terms of three elements that everyone can name and understand.

Shared Knowledge Hub

First, you need a single place where experiments, prompts, rubrics, post-mortems, and patterns are recorded. At the beginning, this can be as simple as a folder with an index page. The goal is reuse: the next team should start with prior learning, not from a blank page.

Safe Data Access

Second, you need predictable, well-understood access to the data required for experimentation.

- The right people can reach the right data with minimal friction.
- It is clear who can read what, how to request access, and where the “red lines” are (for example, no unredacted personally identifiable information outside a sandbox).

If access routinely takes more than a day, you will suffocate momentum. [6]

[6] Martin Kleppmann, *Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems* (Sebastopol, CA: O’Reilly Media, 2017); Chip Huyen, *Designing Machine Learning Systems: An Iterative Process for Production-Ready Applications* (Sebastopol, CA: O’Reilly Media, 2022).

3

The Building Blocks of an **AI-Ready Team** People, Process, Platform, Policy

“Right-Tool” Menu in Plain Language

Third, you need a short, accessible description of the tools available. Publish a simple page that explains, in plain language:

- When to use a large hosted model
- When to consider an open-source model
- When a smaller, on-device model is sufficient

Add two constraints that prevent endless debates:

- A cost limit for experiments
- A latency limit for user-facing ideas

If a proposed test exceeds those limits, it must either be justified with stronger evidence or scaled down.

Practical starter moves

- Create an `/experiments` section in your knowledge hub and seed it with two high-quality Decision Logs and an example AI Experiment Worksheet.
- Nominate one person per pod to keep links organized (prompts, inputs, outputs). Rotate this role monthly so that everyone learns the structure and no single person becomes a bottleneck.
- Add a “Retired” folder for patterns that did not pay off, with a brief note on why. This reduces the risk of “zombie” rework on ideas that have already been tested.

4. Policy (Enabling Speed Without Smothering It)

The purpose of policy in AI work is confidence, not control. People move faster when they clearly understand what is permitted.

Greenlight Rules (One Page)

A concise Greenlight Rules document can provide most of what teams need. It should cover:

- **Data.** What you will never use (for example, unredacted PII), what requires explicit approval, and what is acceptable in sandbox environments.
- **Labeling.** When and how AI-assisted content must be labeled.
- **Logging.** What must be recorded for every experiment (inputs, outputs, owner, decision) and where that record resides.
- **Human-in-the-loop moments.** Where a human must review or approve output (for example, customer-facing communications, legal documents, or sensitive classifications).
- **Exposure levels.** The stages of exposure—internal only, pilot group, general availability—and who can authorize each step.
- **Rollback authority.** Who has the authority to disable or roll back a feature if it misbehaves.

Well-articulated Greenlight Rules give teams confidence to move quickly while staying within clear ethical and regulatory boundaries.[7]

[7] National Institute of Standards and Technology, *Artificial Intelligence Risk Management Framework (AI RMF 1.0)* (Gaithersburg, MD: U.S. Department of Commerce, 2023).

CHAPTER 4



Hiring AI-Ready Talent

Prediction Over Pedigree

4

Hiring AI-Ready Talent Prediction Over Pedigree

Hiring for Throughput, Growing for Reuse

Hiring based primarily on prestigious schools and well-known company names often misses the people who actually move the needle. In an AI-enabled organization, the individuals you need are those who can:

- Turn a vague request into a clear test
- Run that test safely
- Judge the results with discipline
- Decide what to do next week

Your hiring and training system should identify and develop these builders, and then reward them for creating patterns that other teams can reuse.[8]

What You Are Really Screening For

You do not need to be deeply technical to understand what “good” looks like in AI-ready talent. The most important signals are observable behaviors, not buzzwords.

Systems Thinking

Can the candidate explain how a change will affect users, data flows, and costs? Ask them to walk through this out loud or to sketch a simple flow. You are listening for an ability to see consequences beyond a single function or model call.

Problem Slicing

Can they break a broad request into a small, testable change that could reasonably be evaluated in two weeks, without hand-waving? Effective AI practitioners think in slices, not in monolithic “AI initiatives.”

Evaluation Mindset

Do they talk about how they will measure success before they discuss specific tools? Can they name a small set of checks (quantitative and qualitative) they would use to determine whether an experiment is working?

Bias Awareness

Can they explain how results might be wrong, misleading, or unfair, and suggest practical steps to reduce that risk? This does not require advanced ethics training, but it does require maturity about failure modes and trade-offs.

Clear Writing

Can they write a short decision note that someone in another team would understand without further explanation? Clear writing is often a reliable proxy for clear thinking, especially in distributed teams.[9]

[8] Geoff Smart and Randy Street, *Who: The A Method for Hiring* (New York: Ballantine Books, 2008).

[9] Nicole Forsgren, Jez Humble, and Gene Kim, *Accelerate: The Science of Lean Software and DevOps* (Portland, OR: IT Revolution Press, 2018); see also case studies of distributed teams at companies such as Amazon and GitLab.

4

Hiring AI-Ready Talent Prediction Over Pedigree

Asynchronous Collaboration

Are they comfortable writing down assumptions, decisions, and next steps so that work can move forward across time zones? In AI-enabled systems, durable written context is as important as code.

An Interview Question That Surfaces What Matters

Instead of opening with “Tell me about yourself,” consider a more targeted prompt: “Walk me through an experiment you ran—your idea, how you tested it, how you measured it, what you shipped or rolled back, and what changed afterward.” You are listening for closure and learning. A polished success story is less important than a clear narrative of hypothesis, test, evaluation, and decision.

A practical work sample (90–120 minutes)

Brief (sent in advance)

You are assisting a team that wants to auto-summarize customer support chats. Using a small set of anonymized transcripts, please:

1. Show how you would find and organize the knowledge those summaries should use.
2. Compare two options: one large hosted model versus a smaller or open-source option.
3. Define simple acceptance checks (quality, safety, cost, and speed).
4. Recommend a two-week test.
5. Draft a one-page Decision Log entry.

Expected output

- A few bullet points on the knowledge they would use and how they would structure it (plain language, not code).
- A short table comparing the two options (pros and cons with rough cost and speed estimates).
- Acceptance checks and at least one or two guardrails (for example, no leakage of sensitive information; labeling low-confidence results).
- A recommended test slice (for example, “shadow mode on one category of support chat”) with a demo date and rollback plan.
- A draft Decision Log.

4

Hiring AI-Ready Talent Prediction Over Pedigree

Rubric (score each 0–3)

- Hypothesis clarity. Is the goal clear and testable?
- Trade-off literacy. Do they understand and articulate quality versus cost versus speed?
- Evaluation design. Do their success checks reasonably match the problem?
- Reasoning transparency. Can others follow their steps?
- Next-step sizing. Is the proposed test small and reversible?

This is a prediction over pedigree: you watch the candidate do the job you will need them to perform in the coming months.

The 12-Week Upskilling Plan: From Orientation to Real Value

You do not turn people into AI-ready operators with a single workshop. You do it through a simple, repeatable path that combines literacy, guided practice, and ownership.

Weeks 0–2 · Literacy and Safety

- Read. Greenlight Rules, Decision Log examples, and the AI Experiment Worksheet.
- Exercise. Improve an internal task (for example, generating test data or summarizing internal documents) and close the loop with a small experiment and brief write-up.
- Shadow. Join the weekly demo; share one assumption and one small test idea.

Weeks 3–6 · Guided Experiments

- Pair with a mentor and run two closed experiments using the Worksheet.
- Include a basic cost and latency estimate and a rollback plan.
- Present at least one demo and receive feedback on both the success checks and the narrative.

Weeks 7–12 · Ship a Real Win Behind a Feature Flag

- Select the most promising experiment and ship it behind a feature flag with limited exposure.
- Add simple telemetry (how often it is used, time saved, cost).
- Write a short runbook: how to set it up, key guardrails, and how to roll it back.
- Present “what we would do differently next time” to the team.

Artifacts

Printable versions of the 12-Week Curriculum and the AI Experiment Worksheet are included in the Templates section.

4

Hiring AI-Ready Talent Prediction Over Pedigree

Latin America as a Strategic Edge

Nearshore engineering talent in Latin America can be a significant advantage, particularly for AI-related work:

- Time zone overlap. Daily demos and shared rituals are possible without late-night calls.
- Strong asynchronous habits. Many experienced engineers in the region are already accustomed to clear commit messages, document-first communication, and working comfortably in English and Spanish.
- Cost-effective seniority. You can staff experienced individual contributors and technical leads who mentor others through the 12-week plan, without exceeding budget constraints.

Onboarding for Asynchronous Excellence

To make this advantage real:

- Provide Greenlight Rules, a short glossary, and two strong Decision Logs on Day 1
- Define overlap hours (for example, two to three hours per day) and a weekly demo time that everyone respects.
- Encourage bilingual documentation where it adds clarity—headings in English, with optional inline notes in Spanish or Portuguese where useful.

Growth Paths and Promotions Aligned With Throughput

Promotions should reflect how much an individual increases learning throughput and reusable impact, not simply tenure or volume of code. Career paths can be framed in terms of experimentation, reuse, and leadership.

- **IC1 → IC2**
 - Runs one closed experiment with a mentor.
 - Writes a clear Decision Log.
 - Participates consistently in demos.
- **IC2 → IC3**
 - Independently slices work into testable units.
 - Designs simple acceptance checks.
 - Ships and rolls back responsibly.
 - Adds at least two reusable patterns (templates, rubrics, prompts) to the knowledge hub.
- **IC3 → Senior**
 - Coaches a junior engineer through the 12-week plan.
 - Drives reuse across pods.
 - Reduces MTTR-AI (time to recover from AI issues) for at least one feature.
 - Leads a blameless post-mortem.

4

Hiring AI-Ready Talent Prediction Over Pedigree

- Senior → Lead
 - Manages a small portfolio of bets.
 - Balances quality, cost, and speed.
 - Establishes a two-speed lane (experiments versus hardening) for a pod.
 - Improves at least one organization-level KPI (for example, cycle time to insight).

This kind of progression echoes research on “impact-based” career paths that reward outcomes and learning rather than mere output.[10]

Compensation and Incentives

- Incentive structures signal what an organization truly values.
- Recognize closed experiments (with clear decisions), reusable patterns (templates, rubrics, prompts), and mentorship outcomes (mentees completing the 12-week plan).
- Avoid rewards based on lines of code or activity volume alone. Those measures encourage behavior that works against learning throughput.\

Vignette: From Junior to AI Lead in 12–18 Months

Consider a representative trajectory for a new engineer in Latin America.

- **Quarter 1.** The new hire runs two closed experiments (document summarization and test-data generation). They log decisions, shelve one idea, and iterate on the other.
- **Quarter 2.** They ship a small win behind a feature flag (for example, support reply drafts). MTTR-AI is validated with a clean rollback during an incident.
- **Quarter 3.** They mentor another engineer through the 12-week plan. Both contribute patterns to the knowledge hub (a prompt rubric and an acceptance checklist).
- **Quarters 4–6.** They coordinate a small portfolio across two pods, formalize two-speed delivery, and reduce cycle time to insight from 14 days to 5 in their area of responsibility.

Promotion to an AI Lead role follows, with responsibility for a portfolio of bets and the reuse of patterns across teams.

What to Change This Month

To begin aligning your hiring and development practices with AI-throughput:

1. Replace generic technical screens with a work-sample evaluation similar to the support-summarization exercise.
2. Adopt a hiring scorecard based on the aptitude signals you care about: systems thinking, problem slicing, evaluation mindset, bias awareness, writing, and asynchronous collaboration.
3. Launch a pilot 12-week upskilling cohort of two to four engineers with a named mentor and clear milestones.
4. Tie part of your recognition and promotion criteria to closed experiments and reusable patterns, rather than to raw output alone.

These changes do not require a complete overhaul of your organization. They require a shift in what you observe, what you reward, and how you help people grow into the roles AI-enabled teams actually need.

[10] Camille Fournier, *The Manager's Path: A Guide for Tech Leaders Navigating Growth and Change* (Sebastopol, CA: O'Reilly Media, 2017).

CHAPTER 5



Designing AI-Ready Teams

How to Organize the Work

5

Designing AI-Ready Teams How to Organize the Work

When Structure Becomes an Asset

Most teams do not stall because people are not capable enough. They stall because it is unclear who owns what and who decides what. Ambiguity about responsibilities and decision rights quietly erodes throughput.

- The remedy does not need to be complex. A simple, repeatable structure can do much of the heavy lifting:
- Three types of teams: Pods, Platform, and Enablement
- A clear two-speed delivery model
- A small set of rituals that make US–Latin America collaboration smooth and predictable

The aim of this chapter is to show how to organize the work so that AI experiments and productionization can move at an appropriate pace, without relying on heroics.[11]

The Three Team Types And What They Own

You can think of these as “hats” the organization wears, rather than permanent power centers. Smaller companies may combine hats; larger ones will separate them more clearly.

Pods (Product Squads) – Closest to Users

Pods are the teams closest to customer problems and workflows. They own experiments from start to finish, decision logs and demo cadence, and the core KPI for the workflow they serve (for example, time to resolve tickets). They do not own long-term infrastructure, company-wide data access rules, or shared templates and tools.

Example KPIs for pods:

- Closed experiments per week
- Cycle time to insight
- Percentage of decisions that used AI-assisted analysis
- Number of shipped, reversible wins

Platform (Enablers) – Removing Friction at Scale

Platform teams reduce friction and make good patterns reusable at scale. [12] They are accountable for safe data access, the shared knowledge hub, common evaluation templates, feature flags and rollbacks, and basic telemetry and monitoring across systems. They are not responsible for setting product priorities or deciding what any given pod ships in a particular week. Instead, they focus on enabling others.

Example KPIs for platform teams:

- Time-to-first-experiment for new pods
- Reuse rate of patterns and templates
- MTTR-AI (time to recover from AI issues) across features

[11] Matthew Skelton and Manuel Pais, *Team Topologies: Organizing Business and Technology Teams for Fast Flow* (Portland, OR: IT Revolution Press, 2019).

[12] Matthew Skelton and Manuel Pais, *Team Topologies: Organizing Business and Technology Teams for Fast Flow* (Portland, OR: IT Revolution Press, 2019); Nicole Forsgren, Jez Humble, and Gene Kim, *Accelerate: The Science of Lean Software and DevOps* (Portland, OR: IT Revolution Press, 2018).

5

Designing AI-Ready Teams How to Organize the Work

Enablement (Coaches) – Raising the Floor

Enablement functions raise the baseline capability of the organization. They design and deliver training, host office hours, run “red-team” reviews to stress-test higher-risk ideas, and curate templates and playbooks so that teams do not need to reinvent foundational practices. They are not the ones doing delivery work themselves; their role is to accelerate delivery and make it safer.

Example KPIs for enablement:

- Adoption of demos and Decision Logs
- Quality and consistency of rubrics
- Completion rates for the 12-week upskilling plan
- Consistent practices across pods

In combination:

- Pods create value.
- Platform makes that value repeatable.
- Enablement makes people and practices better.

Everyone can see whether they are succeeding by looking at a small set of shared metrics.

Making Decision Rights Simple

Unclear decision rights are one of the fastest ways to kill momentum. A one-page Decision Rights Map can remove a great deal of friction. It should name roles, not individuals, and indicate where their authority begins and ends.

Default Exposure Levels

You can define three standard exposure levels and who approves each:

- Internal only: Pod lead can start and stop.
- Pilot (limited users): Product owner approves; Platform ensures feature flags and basic telemetry are in place.
- General availability (GA): Director or general manager approves, after Enablement’s checklist has been met.

Rollback Authority

Rollback should be simple and well understood:

- Any on-call engineer or pod lead can roll back an AI-enabled feature that violates its acceptance checks or suddenly drives up cost or latency.
- They must log the rollback in the Decision Log and post a brief recap within 24 hours.

5

Designing AI-Ready Teams How to Organize the Work

Handling Exceptions

- If a test touches regulated content or highly sensitive data, involve Policy (typically a joint view from Enablement plus Platform) for a short review.
- Timebox this review to 48 hours; default to Internal or Pilot exposure unless a rule explicitly prohibits it.

Template: Decision Rights

A practical Decision Rights Map can fit on a single page. For each step, define:

Step → Responsible → Accountable → Consulted → Informed

Include key steps such as:

- Starting an experiment
- Defining acceptance checks
- Requesting and setting up data access
- Running the demo
- Changing exposure (internal → pilot → GA)
- Approving evaluation rubrics
- Productionization planning
- GA release
- Rollback if triggered

The goal is not to document every contingency, but to make the default path obvious and visible to all involved.

5

Designing AI-Ready Teams How to Organize the Work

Rituals That Make US–LATAM Collaboration Work

Most collaboration problems emerge at hand-offs, not because of geography. A small set of shared rituals creates predictable touchpoints where decisions are made and information is exchanged.

1. Overlap Window (2–3 Hours)

- Choose a fixed time window that both regions protect.
- Place the weekly demo inside this window and avoid moving it without strong reason.

2. Weekly Demo (45 Minutes)

- Structure the time into 5–7 minute segments per team; anything shown must actually run.
- End every segment with a decision—Ship, Shelve, or Iterate—and record it in the Decision Log.
- Rotate facilitators between US and Latin America each month to reinforce shared ownership.

3. Friday Decision-Log Roundup (15 Minutes, Asynchronous)

The pod lead posts a brief update in the knowledge hub:

- Links to that week’s decisions
- What shipped
- What was rolled back
- One pattern worth reusing

4. Monthly Cross-Pod Retrospective (60 Minutes)

Divide the session into three parts:

- a. Celebrate reuse.
- b. Surface blockers.
- c. Agree on one platform improvement.

Maintain a backlog of platform improvements, ranked by how much they reduce time-to-first-experiment.

5. Bilingual Documentation Norms

To keep everyone included without fragmenting the knowledge base:

- Headings and key fields are in English.
- Inline notes or examples may appear in Spanish or Portuguese where they aid understanding.
- Reviewer comments can be in either language.
- Final decisions are summarized in English in the Decision Log.

This allows teams to work naturally while maintaining a single, coherent record.

5

Designing AI-Ready Teams How to Organize the Work

Lightweight RACI for AI features

A simple RACI (Responsible, Accountable, Consulted, Informed) grid for AI features should fit on one page and can be attached to the Decision Rights Map.

An example structure:

Step	Responsible	Accountable	Consulted	Informed
Define experiment & acceptance checks	Pod	Pod Lead	Enablement	Stakeholders
Data access request & setup	Platform	Platform Lead	Pod	Security/Legal (if needed)
Run experiment & weekly demo	Pod	Pod Lead	Enablement	Platform
Exposure change (internal → pilot)	Pod	Product Owner	Platform, Enablement	Support, Sales
Productionization plan	Platform	Platform Lead	Pod, Enablement	Stakeholders
GA release	Platform	Director/GM	Pod, Enablement	All Hands
Rollback (if triggered)	On-Call/Pod Lead	Platform Lead	Product Owner	Stakeholders

The exact entries will vary by organization, but the principle remains: for each step, mark Responsible, Accountable, Consulted, and Informed, and make the grid visible where teams work.

5

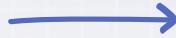
Designing AI-Ready Teams

How to Organize the Work

Common failure patterns (and quick fixes)

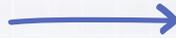
Even with a sound structure, certain patterns tend to recur. A few of the most common, with corresponding remedies:

Ownership drift: Pods expect Platform to decide product scope.



Fix: Restate KPIs clearly; Platform declines product decisions and offers patterns and tooling support instead.

Platform bottleneck: A single gatekeeper delays productionization.



Fix: Publish the productionization checklist and share authority across at least two platform leads per time zone.

Ritual decay: Demos devolve into slide presentations; Decision Logs fall out of date.



Fix: Refuse slide-only demos. Ask in every session, "Where is this decision recorded?"

Ping-pong hand-offs: Work bounces repeatedly between pods, platform, and enablement.



Fix: Enforce a simple hand-off checklist plus a brief, live review in the overlap window.

Silent cost creep: Features ship with no explicit cost ceiling or visibility.



Fix: Attach a cost ceiling to every experiment; Platform reviews and posts monthly spend deltas.

What to change this month

- A few concrete steps can bring this chapter's ideas to life without a large restructuring:
- Publish the Decision Rights Map and the RACI. Make them accessible where teams plan and review work.
- Stand up two-speed lanes and run one productionization hand-off end-to-end (from experiment to feature behind a flag).
- Lock a weekly demo time inside the overlap window and rotate facilitation across regions.
- Start tracking two organization-level signals:
- Closed experiments per week
- Time-to-first-experiment for new pods
- Maintain a platform improvement backlog ranked by its impact on those two signals.

The Templates section [includes ready-to-use examples](#) for your Weekly Rhythm, Decision Rights Map, and RACI grid.

CHAPTER 6



Building an AI-First Culture

Courage With Controls

6

Building an AI-First Culture Courage with Controls

Culture Is What Gets Rewarded and Repeated

Posters that list company values do not ship features. Meaningful behavior change occurs when three elements are aligned:

- The stories leaders and teams tell
- The outcomes and actions the organization rewards
- The behaviors leaders consistently model in public

An AI-first culture is not defined by enthusiasm for particular models or tools. It is defined by making curiosity operational, through clear practices, and pairing it with lightweight controls, so that teams can run closed experiments without undue risk. In effect, courage is matched deliberately with mechanisms for safety. [13]

Choosing the Story Your Culture Repeats

Every organization runs on a small set of internal stories. Rather than allowing them to emerge by accident, it is useful to choose one or two core messages and repeat them until people can quote them back.

Examples include:

- “AI is a teammate, not a threat.”
- We use it to draft, compare options, and accelerate learning—not to replace human judgment.
- “Demos over decks.”
- If it does not run, it is not real.
- “Reversible by design.”
- Anything we ship can be rolled back quickly; that is part of our craft.

These statements only matter if they are backed by real examples. For instance:

“Here is a feature we rolled back in seven minutes, and here is what we learned.”

Credible stories will always outperform slogans.

Leader Behaviors That Set the Tone

People are influenced less by what leaders say than by what they consistently do in public.

A few visible behaviors are especially important:

- Saying “I do not know yet.”
- Leaders who acknowledge uncertainty signal that it is normal. The follow-up—“Let us design a two-week test”—shows what to do next.
- Praising reversible failure.
- When someone pulls the plug quickly and writes a clear Decision Log, thank them explicitly. Highlight the customer pain and cost they avoided.

[13] Amy C. Edmondson, *The Fearless Organization: Creating Psychological Safety in the Workplace for Learning, Innovation, and Growth* (Hoboken, NJ: Wiley, 2019).

6

Building an AI-First Culture Courage with Controls

- Asking for the Decision Log, not only slides.
- When a presentation arrives as a deck, ask, “Show me what actually ran and the decision we took.” This reinforces that experiments, not just narratives, move the roadmap.
- Narrating trade-offs aloud.
- Talk openly about cost, speed, and quality, how you limited risk, and why you chose a smaller slice. Make this trade-off language your default.

Over time, these behaviors make experimentation, transparency, and reversibility feel like the normal way of working.

Make the Safety Net Visible

Teams move quickly when the worst-case outcome is a controlled rollback and a short, blame-free write-up, not a public shaming.

Rollback Norms

Rollback expectations should be explicit and straightforward. Any on-call engineer or pod lead should have the authority to roll back an AI-assisted feature that fails its acceptance checks or unexpectedly drives up cost or latency. The obligation that comes with this authority is simple: log the event and add a short note to the knowledge hub within twenty-four hours describing what happened, what changed, and what the team intends to do next. This combination—clear authority and a light documentation requirement—helps protect customers and budgets without slowing teams down.

Blameless Post-Mortems (20 Minutes)

Post-mortems do not need to be long to be effective. A focused twenty-minute conversation can achieve a great deal. Begin by establishing the basic facts: the timeline of events and the impact on users or systems. Then identify the contributing conditions that set the stage—such as vague acceptance checks, unclear ownership, or missing data. Finally, agree on a single improvement to test in the coming week. The goal is to strengthen the system, not to assign blame.

A Badge of Craft

Within this operating model, the ability to ship with a feature flag and a clear rollback plan should be regarded as a professional skill, not as an afterthought. Engineers and teams who demonstrate disciplined reversibility—shipping carefully, monitoring thoughtfully, and rolling back decisively when needed—are practicing a form of craft that is characteristic of high-reliability organizations.

It is worth recognizing this explicitly, both to reinforce the behavior and to signal to others what “good” looks like in an AI-enabled environment.

6

Building an AI-First Culture Courage with Controls

Greenlight Rules (One Page, Always Visible)

Rules should enable action, not paralyze it. A concise Greenlight Rules document can provide a common frame of reference. It should address:

- **Data.** What is out of bounds (for example, unredacted personal data), what requires approval, and what is acceptable for experiments.
- **Labeling and disclosure.** When AI-assisted content must be labeled internally and for customers.
- **Logging.** The minimum record for every experiment (inputs, outputs, owner, decision link).
- **Human-in-the-loop moments.** Where a person must review output (for example, customer messages, legal documents, sensitive categories).
- **Exposure tiers.** The stages of exposure—internal, pilot, general availability—and who can authorize each step.
- **Rollback authority.** Who can “pull the plug.”

Post this one-pager wherever teams work and reference it in demos. When someone asks, “Are we allowed to...?”, the default response should be to point to the rule that says, “Yes, within these bounds.” [14]

Reward the Right Things

If you reward “slide theater,” you will get more slide theater. If you reward learning and reuse, you will get more learning and reuse.

Shift recognition towards:

- **Closed experiments.** Decision Logs that actually changed the roadmap—ship, shelve, or iterate.
- **Reuse created.** Templates, checklists, prompts, or patterns that at least two other pods adopt.
- **Mentorship outcomes.** Mentors whose mentees complete the 12-week plan and ship a win behind a feature flag.

Small, visible forms of recognition often have more impact than large but invisible ones:

- **Public acknowledgements** during demos
- A rotating “**Most Reused Pattern**” or “**Best Failure**” trophy
- **Occasional lunches** or office hours with an executive sponsor
- A quarterly “**Fastest Rollback**” recognition

These signals reinforce that the organization values learning throughput, not just volume of output.

[14] National Institute of Standards and Technology, *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*, NIST AI 100-1 (Gaithersburg, MD: U.S. Department of Commerce, 2023).

CHAPTER 7



Your First 6–12 Months

A Practical Roadmap

7

Your First 6–12 Months A Practical Roadmap

A Year That Trades AI Hype for Shipped Results

In twelve months, you can move from “we should use AI” to a steady rhythm in which teams ship real, reversible value. The goal is not to automate everything. The goal is to build learning throughput into how you work, so that each quarter:

- More work starts from proven patterns, and
- Fewer bets get stuck in limbo

What “Good” Looks Like by Month Twelve

By the end of the first year, you are aiming for a recognizable picture.

- **Cadence.** Weekly demos with clear decisions have become “just how we work.” Experiments close, and Decision Logs are searchable and used.
- **Shipped value.** Four to eight AI-driven improvements are live in production across two or more workflows, each reversible via feature flags.
- **Platform basics.** A simple knowledge hub is in place, with basic cost and speed (latency) limits and telemetry that indicates what is working.
- **Culture.** Greenlight Rules are pinned where people work; a regular Showcase runs; “best failure” and “most reused pattern” are real, used awards.
- **Metrics.** Time from idea to evidence has decreased; experiment velocity has increased; MTTR-AI (recovery time) is measured in minutes; reuse of patterns is rising.

The remainder of this chapter outlines how to reach that state in the first 90 days and then quarter by quarter.

Your First 90 Days – Getting Off the Ground

Days 0–30 · Ignite

In the first month, the priority is to establish basic guardrails and a working rhythm.

- Publish your Greenlight Rules and route AI-related questions through them.
- Start weekly demos and a Decision Log; close two or three small experiments end-to-end.
- Choose one workflow—such as support triage, coding assistance, or internal search—and write a clear acceptance checklist for it.

During this period, pay particular attention to demo participation rates and the number of closed experiments per week.

7

Your First 6–12 Months A Practical Roadmap

Days 31–60 · Stabilize

In the second month, focus on creating shared memory and validating reversibility.

- Stand up your knowledge hub—an /experiments folder with a simple index page is sufficient to start.
- Prove reversibility at least once: trigger a mock rollback on a low-risk internal feature and record your MTTR-AI (time to recover).
- Start a simple KPI dashboard that tracks time from idea to evidence (cycle time to insight) and experiment velocity.

Watch the time-to-first-experiment for a new pod and capture your first MTTR-AI number.

Days 61–90 · Ship

By the end of the first 90 days, you should have evidence of shipped, reversible value.

- Productionize one or two wins behind feature flags, with basic telemetry on usage, time saved, and cost.
- Hold your first Showcase with one win and one instructive failure; post clips and Decision Log links in the knowledge hub.
- Create a small platform backlog focused on reducing time-to-first-experiment (for example, access request forms, evaluation checklists, flag boilerplate).

At this stage, monitor the percentage of decisions that used AI assistance and how often other teams begin to reuse your evaluation rubrics.

Quarter 2 – Scaling the Wins

Theme: Spread the rhythm and prevent random tool adoption.

In the second quarter, the aim is to extend what works without losing coherence.

- **Formalize two-speed lanes:**
 - Pods own the Experiment Lane (rapid tests, demos, Decision Logs).
 - Platform supports the Productionization Lane with feature flags and telemetry.
- **Add a second workflow** (for example, QA assistance or another internal workflow such as document search).
- **Have Enablement run the first 12-week upskilling cohort**, with mentors explicitly assigned.

Targets for this quarter might include:

- Six to eight closed experiments per month across pods
 - At least one clean production rollback to validate MTTR-AI in a real incident
- It is also valuable to retire one bet publicly that did not pay off and explain why. This reinforces portfolio thinking: not every bet needs to win.

7

Your First 6–12 Months A Practical Roadmap

Quarter 3 – Turning Knowledge Into a Platform

Theme: Make reuse the default.

In the third quarter, the focus shifts from individual wins to institutionalizing what works.

- Standardize rubrics and acceptance checklists, and promote them to templates.
- Expand the knowledge hub with a simple index that organizes patterns by domain (triage, summarization, routing, generation) and links to prompts, evaluation sets, and runbooks.
- Introduce a Build-versus-Buy Decision Tree and a Vendor Diligence Checklist into intake, so tool choices stop being ad-hoc.

Aim to cut time-to-first-experiment for a new pod by roughly 50 percent (from initial request to first demo). Success here looks like a reuse rate that is trending upward and experiment velocity that remains steady even as more teams participate.

Quarter 4 – Managing a Small Portfolio and Edge Cases

Theme: Add sophistication where it matters; keep reversibility first.

By the fourth quarter, you will have enough experience to manage a small portfolio of AI-related bets.

- Run a portfolio of three to five initiatives, balancing quick wins with one longer “edge” or on-premises pilot if you genuinely need it (for reasons such as privacy or latency).
- Tighten FinOps for AI by setting cost ceilings per feature and adding caching or batching where it does not damage the user experience.
- Sunset at least one older approach: write a short migration note and remove legacy approaches from documentation so new team members do not copy them by default.

Targets for this phase can include:

- Another meaningful reduction in cycle time to insight (for example, a 20–30 percent drop)
- MTTR-AI consistently below fifteen minutes
- Cost per AI interaction that is stable or declining

Targets for this phase can include:

- Another meaningful reduction in cycle time to insight (for example, a 20–30 percent drop)
- MTTR-AI consistently below fifteen minutes
- Cost per AI interaction that is stable or declining



Your First 6–12 Months

A Practical Roadmap

KPIs and Review Rhythm

A small set of metrics and a consistent review cadence help keep the roadmap grounded.

Key metrics:

- Cycle time to insight. Aim for seven days or less from idea to evidence by Quarter 2, and five days or less for mature pods by Quarter 4.
- Experiment velocity. Target four to eight closed experiments per month across active pods.
- Percentage of decisions augmented by AI. Aim for 30–50 percent on targeted workflows by Quarter 4.
- MTTR-AI. Prove less than thirty minutes by Quarter 2; less than fifteen minutes by Quarter 4 through drills and real incidents.
- Reuse rate. By Quarter 3, aim for at least one pattern per month that is used by two or more pods.

Review rhythm:

- Monthly reviews between pods and leadership, focusing on these metrics and on Decision Logs and runbooks rather than on slides.
- Quarterly retrospectives at the leadership level, with selected highlights shared company-wide.

Make these meetings artifact-first: start with Decision Logs, telemetry, and runbooks; use slide decks only to support or summarize.

Risk Log and Rollback Calendar

Managing AI-related risk is not about eliminating it; it is about ensuring that you can reverse course quickly, calmly, and deliberately.

- Pre-commit to a rollback drill window each quarter, and avoid major launches during explicit freeze periods.
- Practice at least one live rollback per quarter.
- Maintain a short risk log in the knowledge hub that lists key dependencies, data-sensitivity notes, known failure modes, and who is on-call.

The goal is not zero risk. The goal is to ensure that when issues arise, the organization knows how to respond without panic.

7

Your First 6–12 Months A Practical Roadmap

Case Example: Putting This Roadmap to Work

A mid-market SaaS company began at Level 2 (“Pilot-Curious”) with scattered demos and no consistent decisions.

- Days 0–30.** → Greenlight Rules were introduced, weekly demos began, and a Decision Log was created. Two experiments were closed.
- Days 31–60.** → The knowledge hub went online; a mock rollback established MTTR-AI at twenty-two minutes.
- Days 61–90.** → Two wins were shipped behind feature flags, each with telemetry.

During Quarter 2, two-speed lanes were formalized. The company averaged seven closed experiments per month and executed one clean production rollback during a traffic spike. In Quarter 3, rubrics were standardized, a Build-versus-Buy Decision Tree reduced tool churn, and time-to-first-experiment dropped from twenty-one days to seven. By Quarter 4, the team was managing a small portfolio that included one on-premises pilot; cycle time to insight fell by an additional thirty percent and pattern reuse doubled.

They entered the following year at Level 4 (“Platformized”) with a one-page playbook they could explain to any new team.

What to Do Today

To begin using this roadmap immediately:

1. Block ninety minutes to plan your first thirty days using this chapter as a guide.
2. Choose one workflow and write its acceptance checklist.
3. Schedule your first Showcase and your first rollback drill.
4. Print the quarterly planner and KPI worksheet and post them where demos occur.

In the next section, we will focus on how to keep these practices evolving over time, so that the momentum you build in the first year becomes part of how the organization operates, rather than a short-lived initiative.

CHAPTER 8



Keep **Evolving**

Sustaining AI-Readiness



Keep **Evolving** Staying AI-Ready Over Time

AI-Readiness as a Deliberate Practice

Organizations do not “stay AI-ready” because they care deeply or talk about AI often. They stay ready because they place a few small, repeatable routines on the calendar—and keep them. The aim is not bureaucracy for its own sake, but a steady rhythm that helps you:

- Continue shipping useful changes
- Learn from your own work, and
- Retire what no longer helps

This chapter describes a set of lightweight practices that keep AI-throughput healthy over time.

Quarterly Architecture and Portfolio Review (90 Minutes)

Once per quarter, schedule a focused architecture and portfolio review. The purpose is to understand what you have actually shipped, what it implies for your systems and costs, and what should be retired or adjusted.

Inputs typically include:

- The last quarter’s Decision Logs
- A snapshot of key telemetry (usage, cost, latency)
- MTTR-AI incidents and drills
- The current KPI dashboard

A practical agenda can be organized around five questions:

1. What decisions did we make, and what did they imply for architecture and cost?
2. What should we sunset? Which features, prompts, or patterns can be retired?
Aim to retire at least one item per quarter.
3. What are the cost and performance trends? Look at token or API spend, opportunities for caching or batching, and latency outliers.
4. What new risks have appeared? Consider changes in data residency requirements, evaluation drift, or vendor behavior.
5. What two small nudges to architecture or platform would reduce time-to-first-experiment?

The output is a single page that summarizes what will be retired, the two architectural or platform adjustments to make, and one new bet to explore. Post this note in the knowledge hub so that teams can see the direction and avoid resurrecting retired approaches.



Keep **Evolving** Staying AI-Ready Over Time

Prompt, Tool, and Data Debt Ledger (Monthly)

Fast-moving teams inevitably create a certain amount of “debt”: forked prompts, abandoned evaluation sets, and one-off scripts that no one fully understands. A simple debt ledger keeps this visible and normalizes cleanup rather than letting it accumulate silently.

A straightforward table is sufficient, with columns such as:

- Item
- Type (prompt, tool, evaluation, data)
- Owner
- Estimated pain (for example, hours wasted per month)
- Proposed fix
- Estimated completion date
- Status

Review this ledger monthly in a Platform–Enablement sync and commit to closing at least two items per month. Over time, this keeps the environment understandable and reduces the risk of “zombie” artifacts influencing production work.

Model Sunset and Migration Notes

Retiring models, prompts, and associated artifacts well is as important as introducing new ones. Poorly retired components turn into “zombie systems” that confuse new team members and complicate audits.

A lightweight approach can include:

- **Archive**. Move old prompts, runbooks, and evaluation sets into a /retired area with a short paragraph explaining why they were retired.
- **Redirect**. Update links in the knowledge hub so newcomers arrive at current artifacts rather than dead ends.
- **Communicate**. Post a short note in the demo or AI channel summarizing what replaced the retired item and why.
- **Refresh cadence**. For critical models and prompts, schedule a 6–12 month review to re-score cost, speed, and quality against contemporary options.

These steps prevent drift and help ensure the system reflects current best practice rather than historical accidents.

8

Keep Evolving Staying AI-Ready Over Time

The Community Flywheel: Keeping Ideas Moving

Healthy cultures share patterns without needing constant prompting. You can encourage this by designing a few simple community mechanisms.

- Guilds. Self-organized groups around themes such as triage, search, or coding assistance can meet monthly, with each guild sharing at least one reusable artifact per session.
- Lightning talks. Short, twenty-minute internal demos between pods can be recorded and indexed in the hub.
- US–Latin America exchanges. Pair pods across regions once a quarter to co-present a pattern. This spreads context and normalizes bilingual documentation.
- Requested patterns list. Maintain a small, visible backlog of desired patterns (for example, “evaluation rubric for financial summaries”). Anyone can volunteer to create one.

Over time, these practices create a “community flywheel” in which useful ideas circulate, are refined, and are reused without needing heavy coordination.

Checklist: “Are We Still AI-Ready?”

Twice a year, it is helpful to run a quick self-check. If you cannot answer “yes” to at least eight of the following ten statements, it is likely time to rerun your maturity scorecard and adjust your roadmap. We:

- 1. Run weekly demos and close experiments with a Decision Log.
- 2. Can roll back AI features within minutes, and we practice doing so.
- 3. Have Greenlight Rules pinned and used in demos.
- 4. Maintain a knowledge hub where patterns and post-mortems live and are reused.
- 5. Use a two-speed model (experiments versus productionization) in our pods.
- 6. Track cycle time to insight, experiment velocity, MTTR-AI, and reuse rate.
- 7. Align hiring and promotion with throughput behaviors, not lines of code.
- 8. Run a monthly Showcase that includes both wins and instructive failures.
- 9. Retire at least one pattern or dependency per quarter, with a brief note.
- 10. Refresh our build-versus-buy and vendor diligence decisions at least annually.

If your “no” answers cluster in one area (for example, telemetry, rollback, or reuse), that cluster suggests your next 30-day focus.

Keeping the Loop Tight

In the second year, a common mistake is to add more process instead of tightening the loop between a few essential steps:

- Demos that show real behavior
- Decisions recorded in a Decision Log
- Telemetry that indicates impact and cost
- Retirement of what no longer serves

The aim is to keep artifacts light, cadence steady, and stories honest. That is how “AI-ready” evolves from a one-time initiative into a durable advantage

CHAPTER 9



Bringing It All Together

And How **Better**Engineer Can Help

9

Bringing It All Together And How BetterEngineer Can Help

From AI Hype to a Working System

Across these chapters, we have looked at what it actually takes to build AI-ready teams and organizations:

- Throughput over theater. AI-ready teams do not simply talk about AI; they turn ideas into small, testable bets, evaluate them, and close the loop with decisions.
- Clear levels of maturity. From ad-hoc tinkering to perpetual learning, you have seen how to score where you are today and identify which dimensions are holding you back.
- The core building blocks. People, Process, Platform, Policy, and Culture working together so that experiments are safe, repeatable, and reversible.
- Hiring and upskilling for behaviors, not buzzwords. Screening and developing for systems thinking, problem slicing, evaluation mindset, bias awareness, clear writing, and asynchronous collaboration.
- Team design and collaboration. Pods, Platform, and Enablement; two-speed delivery; and simple rituals that make US–Latin America collaboration predictable.
- A 6–12 month roadmap. A practical sequence that trades hype for shipped value and keeps risk reversible.
- Staying ready. Lightweight reviews, a shared knowledge hub, and a “ship–measure–retire” rhythm so you do not slide backward in year two.

If there is one idea to carry forward, it is this:



AI-readiness is not a tool you buy. **It's a system you build.**
And you don't have to build that system alone.

Where BetterEngineer Fits In

AI transformation is rarely linear and companies usually encounter two major gaps:

- Strategic clarity – knowing where AI can drive real value.
- Executional capability – having the right people to turn those priorities into shipped, reliable systems.

Most consulting firms focus on the first gap. Most staffing firms address the second. Our view is that real change requires both.



Bringing It All Together And How BetterEngineer Can Help

This book was written by [Marc Boudria](#), Chief Innovation Officer at BetterEngineer, based on what we have seen work inside real teams. At BetterEngineer, we help you turn this playbook into practice in two primary ways:

- An AI Readiness Assessment that gives you a clear, practical view of where you are today and what to do next.
- Team augmentation that connects you with senior engineers who already work in the ways this book describes.

Through the AI Readiness Assessment, we help you:

- Pinpoint the real problems AI can solve by analyzing user journeys, data ecosystems, and operational pain points.
- Break free from the “pilot trap” by identifying projects that can scale and deliver measurable ROI.
- Rethink workflows, not just automate existing steps, so you build lasting advantage rather than short-term wins.
- Evaluate feasibility and risk across business units, data, and change management.

With that clarity in place, our augmentation teams step in to execute by:

- Embedding with your pods and platform teams.
- Working with your product, data, and operations leaders.
- Applying the practices described in this book—Decision Logs, weekly demos, Greenlight Rules, two-speed delivery, and clear rollback paths.

You are not left choosing between “strategy” and “implementation.” You gain a partner that helps you uncover what matters most and then provides the talent to build it.

The AI Readiness Assessment: Maximizing AI’s Impact Through Purposeful Discovery

We believe AI’s true value lies not in chasing trends, but in solving the right problems, intelligently and intentionally. AI is not simply another layer on top of outdated processes; it often requires rethinking how the business operates. Our AI Readiness Assessment is designed to help you do exactly that.

Rather than stopping at surface-level experimentation, **we work closely with your team to:**

- Analyze user journeys, data realities, and operational bottlenecks.
- Identify the specific places where AI can genuinely reduce friction or elevate customer and employee experience.
- Validate feasibility and risk across business units, data structures, and change management constraints.

From these insights, we co-create a pragmatic, data-driven roadmap you can actually execute, not one that lives only in a slide deck.



Bringing It All Together And How BetterEngineer Can Help

What you get and how we differ

- A real path forward. A compelling AI roadmap is not enough if you cannot execute it. Our assessment bridges the gap between planning and action, so your initiatives do not remain theoretical.
- Practical, business-aligned AI. We focus on AI applications that deliver measurable impact—improved operations, reduced costs, or better customer experiences. No gimmicks, no unnecessary complexity.
- Scalability built in. We help design AI initiatives with real-world constraints in mind, so they work in production, integrate with your existing systems, and can evolve as the business grows.

Schedule an AI Readiness Assessment

If you are ready to move from scattered experiments to a focused, executable roadmap, your next step is to schedule an AI Readiness Assessment. Together, we will clarify where AI can create real value in your workflows and what to prioritize in the next 6–12 months.

Team Augmentation – Building the Right Team to Make AI Happen

A roadmap only creates value when the right people can bring it to life. Unlike traditional consulting firms that stop at strategy, or staffing firms that start with generic hiring, we do both. We help define what to build and then help you secure the team that can actually build it.

Through our staff augmentation service, we connect you with senior engineers from Latin America who:

- Are strong in systems thinking, problem slicing, and evaluation.
- Are comfortable using Decision Logs, Greenlight Rules, and two-speed delivery.
- Work effectively in your time zones and are accustomed to asynchronous, documentation-first collaboration.

Our teams are:

- Embedded with your organization. Alignment is based on shared work, not just written requirements.
- Multidisciplinary. Capable of spanning machine learning engineering, backend development, DevOps, and AI infrastructure.
- Culturally aligned. Fluent in English, familiar with U.S. business norms, and experienced with agile delivery.
- Mission-aware. Selected not only for technical skill, but for how they think, communicate, and grow with you.

We do not hand over a strategy and depart. We co-build the future with you.



Bringing It All **Together** And How BetterEngineer Can Help

Taking the Next Step

If this book has resonated with the challenges and opportunities you see in your own organization, the most important step is simply not to let the ideas remain abstract.

You can:

1. Schedule an AI Readiness Assessment to understand where you stand today and what to prioritize over the next 6–12 months.
2. Begin building your AI-ready team with senior engineers who can apply these practices in your real environment.



To explore either—or both—options:

- Visit [Betterengineer.com](https://betterengineer.com)
- Contact our team directly to discuss your context and goals

The work of becoming AI-ready is ongoing. With deliberate structure, clear practices, and the right partners, it is also entirely achievable.

CHAPTER 9



AI-Ready **Templates**

The practical kit for shipping AI safely

Decision Log

A print-ready page record of an experiment or feature decision. Captures context, evidence, the decision (Ship / Shelve / Iterate), and what happens next.

How to use: Fill this out at the end of a demo or review. Share the link in your channel and file it in the knowledge hub. Keep it concise—this is a record, not a memo.

1. Header

What this is: Fast scan info so anyone can find, filter, and reference this decision.

How to use: Fill all fields; use precise titles (verb + object).

Title:.....

Decision ID:.....

Date:.....

Owner(s):.....

Pod / Workflow:.....

Related Ticket / PR:.....

2. Context (3–5 bullets)

What this is: Why we looked at this now.

How to use: State the user/problem, scope/slice, and any constraints. Avoid history lessons.

-
-
-

3. Options Considered

What this is: The real alternatives, not straw men.

How to use: List 2–3 options with one line each: approach + key tradeoff.

1.....

2.....

3.....

4. Acceptance Checks & Evidence

What this is: The pass/fail criteria and what actually happened.

How to use: Copy the checks from your Experiment Worksheet. Summarize measured results

1. Checks (pre-set):.....

2. Quality:.....

3. Safety:.....

4. Latency:.....

5. Cost:.....

6. Results (measured)

a.....

b.....

5. Decision

What this is: The call.

How to use: Check one and write 2–4 bullets explaining why—tie back to evidence.

Outcome: Ship Shelve Iterate

Rationale:

-
-

Decision Log

6. Exposure & Rollback Plan (if Shipping)

What this is: How we'll safely expose the change and unwind if needed.

How to use: Name the feature flag and steps (shadow → canary % → GA). List the rollback trigger thresholds.

- Flag / Route:.....
- Exposure steps:.....

Rollback triggers:

- Error rate >
- Latency p95 >
- Cost/1k calls >
- Quality drop (metric) >

7. Follow-Up (if Iterating)

What this is: The next concrete move.

How to use: Write the new hypothesis or slice, who owns it, and the target date.

- New hypothesis/slice:.....
- Owner:.....
- Target date:.....

8. Owners & Approvals

What this is: Who's accountable and who signed off.

How to use: List the accountable owner and any required approvers (e.g., Platform, Security, PMM).

- !:.....
-
-

9. Links & Artifacts

What this is: Where to see details or rerun the tests.

How to use: Link the Experiment Worksheet, eval delta report, telemetry dashboard, and runbook.

- Link.....

10. Review Date

What this is: A reminder to check back.

How to use: Set a date to review this decision (e.g., after canary week or quarterly re-eval). Add it to the calendar.

AI Experiment Worksheet

Standard, print-ready form to kick off and run any AI experiment. Fits on one page for speed and consistency. Aligns with the main guide (S7) but stands alone for design/production.

How to use: Fill before work starts. Attach to demo invites. At close, link the Decision Log and archive in the knowledge hub. [Editable Here](#)

Experiment Summary

What this is: A quick header for discoverability—so ops, leads, and reviewers can find and reference the experiment fast.

How to use: Keep titles specific (verb + object). Link to the ticket/PR you'll demo. If multiple pods are involved, list all owners.

Title:.....
Owner(s):.....
Pod / Workflow:.....
Date Opened:.....
Target Demo Date:.....
Related Ticket / PR:.....

Hypothesis & Slice

What this is: The bet you're making and the smallest test that could prove you right or wrong in ≤ 2 weeks.

How to use: Write one sentence in falsifiable form. Force the slice to the tiniest viable unit of value. Name what you are not doing yet to prevent scope creep.

Data & Constraints

Hypothesis (one sentence): We believe that _____ will _____ for _____ as measured by _____.

Smallest viable slice: What's the tiniest test that proves or disproves the hypothesis in ≤ 2 weeks?

Out of scope (now): What are we explicitly not trying to do yet?

What this is: The inputs and guardrails that bound your test.

How to use: List exact sources (links). Capture privacy/residency notes and any rate limits/tool caps that may affect results. If any PII is present, state the handling.

Acceptance Checks (write before you run)

What this is: The pass/fail criteria that decide Ship/Shelve/Iterate without debate.

How to use: Set numeric thresholds where possible. Keep 3–5 checks total across quality/safety/latency/cost/UX. If you can't measure it, rephrase it until you can.

Quality: (e.g., " $\geq 85\%$ correct labels on golden set; $\leq 2\%$ false positives")

Safety: (e.g., "No claims without citation; pass injection probe pack")

Latency: (e.g., " $p_{95} \leq 700$ ms for route X")

Cost: (e.g., " $\leq \$0.02$ /interaction avg")

UX: (e.g., "Inline provenance shown; clear edit/override affordance")

Method & Evaluation

What this is: The retrieval/model route you'll run and how you'll score it.

How to use: Specify the retrieval recipe (hybrid, reranker, k). Name each route under test and the metrics. Include a small human spot-check plan (sample size + who).

Retrieval recipe: (lexical/dense/hybrid; reranker; k; citation policy)

Model route(s) under test: (e.g., SLM→Frontier fallback)

Evaluation plan: (golden set size; metrics: Precision@k, Faithfulness, task success)

Human spot-check: (sample size; reviewers; schedule)

AI Experiment Worksheet

Guardrails & Exposure Plan

What this is: The safety rails and rollout plan that make speed reversible.

How to use: Reference Greenlight Rule IDs. Define any HITL gates. Name the feature flag and exposure steps (shadow → canary % → GA). State the exact rollback triggers.

Greenlight Rules referenced: (IDs/links)

Feature flag: (name; shadow → canary % → GA plan)

HITL gates (if any): (criteria & approver)

Rollback trigger: (thresholds for error/latency/cost/quality)

Telemetry Fields (must be logged)

What this is: The minimum data needed to understand performance and cost.

How to use: Ensure your service logs these fields for every run. If your stack adds more, list them in the PR but keep this baseline intact.

timestamp, feature_name, route_id, prompt_hash, retrieval_config_hash,
tokens_in, tokens_out, model_id, latency_ms, cost_usd, cache_hit,
citations_count, error_code

Results (fill at demo/close)

What this is: The evidence you'll show in the demo, in one place.

How to use: Paste metric summaries and any key screenshots. If results are mixed, add one sentence per metric explaining the variance.

Retrieval metrics: (Precision@k, Recall@k, MRR, Freshness %)

Answer metrics: (Faithfulness, Groundedness, Utility rubric)

Operational: (p95 latency, cost/1k, error rate, cache rate)

Notes & screenshots:.....

Decision

What this is: The close of the loop—commit to Ship, Shelve, or Iterate and name the next owner.

How to use: Check one outcome. Add 2–4 bullets on why. Link the Decision Log. If Iterate, write the new hypothesis or slice.

Outcome: Ship Shelve Iterate

Decision rationale (2–4 bullets):

Next step & owner:

Decision Log link:

Weekly Rhythm Template

(Weekly Rhythm Agenda)

A print-ready agenda that keeps your AI program moving: demo → decision → recap. Establishes a light, repeatable weekly cadence that ships value without meetings sprawl. Standalone for design/production; aligned with S7 and the Showcase/Postmortem templates.

How to use: Schedule 45–60 minutes on a consistent day. In the case there are remote teams, be sure to choose a time that overlaps time zones. The host enforces time boxes, captures decisions, and posts a recap with links. Quick note, the “Links” area for each section is for you to link to those documents within your organization.

<p>Pre-Reads & Setup (host, day-before) What this is: Prep that makes the meeting fast. How to use: Send a single message with links and the plan.</p> <ul style="list-style-type: none">• Agenda + demo order• Links to each AI Experiment Worksheet and Decision Log• Metrics snapshot (latency p95, cost/1k, cache rate, task success)• Any Greenlight Rules updates or policy notes <p>Links: Agenda doc · Metrics snapshot · Demos list</p>	<p>Opening (5 minutes) What this is: Set context and spotlight momentum. How to use: Host shares wins + top metrics.</p> <ul style="list-style-type: none">• Quick wins since last week (2–3 bullets)• Metrics snapshot: latency p95 __ ; cost/1k __ ; error rate __ ; cache __ ; task success __• Carry-over items from last week (owners/dates) <p>Links: Dashboard · Last recap</p>	<p>Demos & Decisions Block (≈ 30 minutes) What this is: Show it running, show the evidence, make the call. How to use: 5–7 minutes per item; keep Q&A to clarifiers. For each demo (script):</p> <ol style="list-style-type: none">1.Run the feature (live or replay).2.Evidence: acceptance checks + deltas vs golden set.3.Ask: Ship / Shelve / Iterate.4.Decision (host confirms outcome + owner + date). <p>Timekeeper cues: green 4:00 · yellow 6:00 · red 7:00</p>
<p>Backlog & Flags Triage (10 minutes) What this is: Choose the next experiments and remove blockers. How to use: Review 5–10 tickets; confirm owners and dates.</p> <ul style="list-style-type: none">• Next 1–2 experiments per pod (two-week slices)• Feature flag changes needed (shadow → canary → GA)• Cross-pod collisions or dependencies <p>Links: Backlog board · Flag console</p>	<p>Risks & Incidents (5 minutes) What this is: Keep safety and reliability visible. How to use: Note any incidents and drills.</p> <ul style="list-style-type: none">• Incidents since last week (MTTA/MTTR-AI, status)• Upcoming drills (rollback rehearsal date)• Security/privacy updates (DPA, sub-processors) <p>Links: Postmortem(s) · Incident log</p>	<p>Learning Share (5 minutes) What this is: Institutionalize reuse and craft. How to use: One short teach-back per week.</p> <ul style="list-style-type: none">• Pattern card or eval tip of the week• UX/provenance pattern in the wild• Tooling shortcut (prompt library, caching, reranker tweak) <p>Links: Pattern cards · Prompt library</p>
<p>Commitments & Recap (5 minutes) What this is: Close the loop and set the next step. How to use: Host reads out owners/dates; scribe posts recap.</p> <ul style="list-style-type: none">• Owners/dates confirmed for all decisions• Next week's demo candidates named• Recap slated for posting within 24 hours <p>Links: Live notes · Recap post template</p>	<p>Roles & Rotations What this is: Who does what to keep cadence healthy. How to use: Fill once; rotate monthly. Host: Timekeeper: Scribe: Platform rep: Security contact:</p>	<p>Artifact Capture Checklist (scribe) What this is: Ensure the paper trail exists. How to use: Tick before ending the call.</p> <ul style="list-style-type: none">• Decision Log links pasted for each demo• Backlog tickets updated (owners/dates)• Flag/Runbook links added where shipping• Recap draft started (with recording link)

Weekly Rhythm Template

Copy-Paste Agenda (for calendar invites)

What this is: A one-block agenda string.

How to use: Paste into the calendar description.

WEEKLY RHYTHM – Demo → Decision → Recap

1. Opening (wins + metrics) – 5m
2. Demos (5–7m each) – decisions recorded live
3. Backlog & Flags triage – 10m
4. Risks & Incidents – 5m
5. Learning Share – 5m
6. Commitments & Recap – 5m

In this section we make reference of two terms MTTA and MTTR-AI

MTTA (Mean Time To Acknowledge)

What it means: The average time between when an incident is first detected and when a human acknowledges it.

How to calculate (in words):

- a. For each incident, note the time it was detected and the time it was acknowledged.
- b. For that incident, compute the minutes between those two times.
- c. Do this for all incidents in your measurement window (e.g., last 30 days).
- d. Add up all those minutes.
- e. Divide by the number of incidents you included.

Example: If three incidents took 4 minutes, 6 minutes, and 5 minutes to acknowledge, then MTTA is $(4 + 6 + 5) \div 3 = 5$ minutes.

MTTR-AI (Mean Time To Restore – AI)

What it means: The average time between when an AI-related incident is first detected and when service is restored to an acceptable state (either the feature is rolled back/disabled, or metrics are back within agreed envelopes).

How to calculate (in words):

- a. For each AI incident, note the time it was detected and the time it was restored.
 - b. For that incident, compute the minutes between those two times.
 - c. Do this for all AI incidents in your measurement window.
 - d. Add up all those minutes.
 - e. Divide by the number of AI incidents you included.
- Example: If three AI incidents took 12 minutes, 18 minutes, and 15 minutes to restore, then MTTR-AI is $(12 + 18 + 15) \div 3 = 15$ minutes.

Implementation tips:

- Log three timestamps per incident: detected, acknowledged, restored.
- Set a starter target such as $MTTA \leq 5$ minutes and $MTTR-AI \leq 15-30$ minutes; tune by severity and team capacity.

Decision Rights Map

(Exposure & Rollback Authority)

A print-ready map of who can change exposure (shadow → canary → GA) and who can roll back AI features –without waiting. Clarifies decision speed, safety, and comms. Standalone for design/production; aligned with Greenlight Rules, Weekly Rhythm, and Postmortem templates.

How to use: Fill Sections A–G for your org. Post in the hub. Link this in PRs and runbooks. Keep it to one page front-and-back.

Roles & Contacts

What this is: The people who can act.

How to use: Fill names and on-call paths.

Pod Lead:.....
 Platform Rep (flags/telemetry):.....
 Product Manager:.....
 Security/Privacy Contact:.....
 On-Call Engineer (24/7):.....
 Approver on Duty (exec or delegate):.....
 Comms/Support Contact:.....

Exposure Tiers & Default Gates

What this is: The allowed rollout stages and minimum gates per tier.

How to use: Circle the tier you're changing to, verify the gates are met.

- Tier 0 – Shadow (internal traffic only)
- Gates: Worksheet complete · Acceptance checks set · Telemetry live · Rollback written
- Tier 1 – Canary (1–10% real users)
- Gates: Tier-0 gates + alerts configured · Provenance on · HITL if side-effects
- Tier 2 – GA (>10% users)
- Gates: Canary stable ≥ 1 week · Postmortem ready · Decision Log filed · Runbook published

Decision Rights Matrix (RACI-style)

What this is: Who is Responsible, Accountable, Consulted, Informed for common exposure/rollback decisions.

How to use: Mark one letter per role per decision. Add names in Section A.

Decision	Pod Lead	Platform Rep	Product Manager	Security/ Privacy	On-Call Eng	Approver on Duty	Comms/ Support
Move to Tier 0 (Shadow)	R	R	C	C	C	I	I
Move to Tier 1 (Canary ≤10%)	R	R	A	C	C	I	I
Increase Canary %	R	R	A	C	C	I	I
Promote to Tier 2 (GA)	C	R	A	C	C	A	I
Emergency rollback to Shadow/Off	I	C	I	I	R/A	I	I
Pause new exposure (freeze)	C	R	A	C	C	A	I
Change Feature Flag defaults	C	R	A	C	R	I	I
Approve exception to Greenlight Rules	C	C	A	A	I	A	I
Trigger Incident Comms (customer)	I	I	A	C	I	A	R

Legend: **R = Responsible (does the work)** · **A = Accountable (final call)** · **C = Consulted** · **I = Informed**

Decision Rights Map (Exposure & Rollback Authority)

Rollback Authority & Triggers

What this is: Clear thresholds that give people permission to act now.

How to use: Copy into runbooks and feature PRs.

Any on-call engineer may rollback immediately if any of the below breach for ≥ 2 consecutive minutes (or 50 consecutive events), no prior approval required:

- Error rate $> _ _ \%$
- Latency p95 $> _ _ \text{ms}$
- Cost per 1k calls $> \$ _ _$
- Faithfulness/quality metric $< _ _$
- Security/privacy concern raised by Security (verbal is enough)
- Post-action: Notify #ai-ops; file a Decision Log within 24 hours.

Exposure Change Workflow (3 steps)

What this is: The minimal, repeatable flow for changing exposure.

How to use: Follow exactly; link artifacts.

1. Propose: Post in #ai-ops with target tier, gates checklist, and link to Worksheet/Decision Log.
2. Approve: Product Manager (A) confirms; Platform flips flag; Security adds conditions if needed.
3. Announce: Scribe updates the notes doc; timebox monitoring window (e.g., first 60 minutes at new %).

Exception Handling (when rules don't fit)

What this is: How to move fast safely when you must deviate.

How to use: All exceptions must be documented and time-boxed.

- Requestor posts: what/why, risk, compensating controls, time limit.
- Approvals required: Product Manager (A) + Security (A) + Approver on Duty (A).
- Expiry: Exception auto-expires in $_ _ \text{days}$; re-approval required to extend.

Communications Tree

What this is: Who tells whom, in what order.

How to use: Use for both promotions and rollbacks.

- #ai-ops (internal)
- #product and #support (internal)
- Status page (if user-impacting)
- Customer comms via Support (template link)

Templates: Status update · Customer email · Release notes blurb

Sign-off & Review

What this is: Make it real and keep it fresh.

How to use: Get signatures; review quarterly.

- Signatures: Pod Lead $_ _ _ _$ Product Manager $_ _ _ _$ Platform Rep $_ _ _ _$ Security $_ _ _ _$ Approver on Duty $_ _ _ _$
- Effective date:
- Quarterly review date:

AI-Feature RACI Grid

A print-ready matrix that clarifies who is Responsible, Accountable, Consulted, and Informed across the lifecycle of an AI feature (from scoping to GA and operations). Reduces decision friction and speeds safe execution. Standalone for design/production; aligned with Decision Rights Map, Greenlight Rules, Weekly Rhythm, and Postmortem templates.

How to use: Fill the role names up top, then mark one letter (R/A/C/I) per row. Use exactly one A (Accountable) per row and at least one R (Responsible). Keep to one page; review quarterly.

Definitions:

- R – Responsible: Does the work and delivers the artifact/outcome. There can be multiple Rs.
- A – Accountable: Single owner of the decision/result. Must sign off; only one A per row.
- C – Consulted: Gives input before the decision; two-way communication. Keep Cs lean.
- I – Informed: Notified after the decision/change; one-way communication.

Roles (fill names)

What this is: The teams/people in the room.

How to use: Replace with your titles; add initials in the grid.

- Product Manager (PM):
- Pod Lead / Engineering Lead:
- Engineer(s):
- Designer (UX/Content):
- Data/ML (eval/retrieval):
- Platform (flags/telemetry/gateway):
- Security/Privacy:
- Support/Comms:
- Legal (as needed):

Feature Lifecycle RACI

What this is: Who does what at each step.

How to use: Mark exactly one primary A per row; at least one R. Keep C lean.

AI-Feature RACI Grid

Decision / Activity	PM	Pod Lead	Eng	Design	Data/ML	Platform	Sec/Privacy	Support/Comms	Legal
Define user problem & success metrics	A	R	C	C	C	I	I	I	I
Write acceptance checks (quality/safety/latency/cost)	A	R	R	C	R	C	C	I	I
Choose retrieval domains & recipe (chunking/reranker)	C	R	C	I	A	C	C	I	I
Select model route(s) (SLM/frontier; fallback policy)	C	R	C	I	A	C	C	I	I
Draft prompt(s) + few-shots (versioned)	C	R	R	C	C	I	I	I	I
Instrument telemetry (min fields + dashboards)	I	R	C	I	C	A	I	I	I
Configure feature flags & exposure plan	I	R	C	I	C	A	C	I	I
Security & privacy review (DPA, PII handling)	I	C	I	I	I	C	A	I	C
Demo at Weekly Rhythm (evidence shown)	A	R	R	R	R	C	I	I	I
Ship decision (Ship/Shelve/Iterate)	A	R	C	C	C	C	I	I	I
Promote Shadow → Canary	I	R	C	I	C	A	C	I	I
Promote Canary → GA	A	R	C	C	C	C	C	I	I
Emergency rollback	I	A	R	I	I	R	C	I	I
Postmortem ownership	A	R	R	C	C	C	C	I	I
Update Runbook & Version manifest	I	R	R	I	C	A	I	I	I
Publish release notes / comms	C	I	I	C	I	I	I	A/R	C
Pattern card / reuse extraction	C	R	R	R	R	C	I	I	I

Legend: **R = Responsible (does the work)** - **A = Accountable (final call)** - **C = Consulted** - **I = Informed**

AI-Feature RACI Grid

Feature Snapshot (fill for this grid)

What this is: The context this RACI applies to.

How to use: Fill once per feature; attach to Worksheet and Decision Log.

- Feature name:
- Workflow/Pod:
- Links: Worksheet · Decision Log · Runbook · Dashboards

Exceptions & Notes

What this is: Where you record deviations from the default RACI.

How to use: Keep it short, time-bounded, and add owners.

-
-
-

Review & Sign-off

What this is: Make the RACI official.

How to use: Get signatures, then revisit quarterly or after major org changes.

Signatures: PM ____ Pod Lead ____ Platform ____ Security ____ Date ____

Build-vs-Buy Decision Tree

A print-ready, vendor-neutral framework to choose Build, Buy, or Blend for an AI capability. Makes tradeoffs explicit (quality, latency, privacy, cost, ops), ties to acceptance checks, and enforces reversibility. Standalone for design/production.

How to use: Fill Sections A–F in order. Circle Build/Buy/Blend at the bottom. Attach to any proposal that requests budget, staff, or a new platform. Keep it to 1–2 pages.

Quick Triage (2-minute gate)

What this is: A fast filter to avoid platform tourism.

How to use: If any "No," default to Blend or a pilot behind flags.

- Acceptance checks defined? (quality, safety, latency, cost) Yes No
- Data residency/privacy constraint? Strong Moderate None
- Time horizon? < 4 weeks (pilot) 1–2 quarters Long-term
- Reversibility in scope? (feature flags, prompts versioned, exit plan) Yes No

Decision Factors (score 0–3)

What this is: A head-to-head scoring across three concrete options.

How to use: Define each option, then score every factor 0–3 (0 = not met, 1 = weak, 2 = adequate, 3 = strong). Multiply by weight; sum the column totals.

Option definitions (fill first):

- Option 1: Name/description _____
- Mode (circle): Hosted API / Open-Source Hosted (self-hosted) / Edge-On-Device
- Option 2: Name/description _____
- Mode (circle): Hosted API / Open-Source Hosted (self-hosted) / Edge-On-Device
- Option 3: Name/description _____
- Mode (circle): Hosted API / Open-Source Hosted (self-hosted) / Edge-On-Device

Feel free to add other factors that might be unique or important to your specific use case.

Factor	Weight	Option 1	Option 2	Option 3
Quality vs checks (task fit)	3			
Latency envelope (p95)	2			
Privacy/residency fit	3			
Unit cost (per 1k calls)	2			
Ops complexity (team fit)	2			
Roadmap/change risk	2			
Integration speed (≤ 2 wks)	2			
Footprint/Model class fit (frontier/medium/small viable)	2			
Offline/Intermittent connectivity need	3			
Exit posture (export/portability)	3			
Weighted score (\sum factor \times weight)				

Build-vs-Buy Decision Tree

Top 3 risks (one-liners):

- 1).....
- 2).....
- 3).....

Decision Modes (definitions)

What this is: The plain-English meanings of each path.

How to use: Use these to interpret scores; you're not following branches.

- Build: You host/run the capability (open-source model or SLM, your infra). Higher control/residency; more ops.
- Buy: You consume a hosted API for the core capability. Fastest to learn; lower ops; vendor risk; higher unit cost.
- Blend: Combine Buy for the quality-critical path with a Build track for control/cost (e.g., SLM or OSS in parallel). Revisit after a 2-week bake-off.

Scoring Interpretation & Thresholds

What this is: How to turn Section B's weighted totals into a recommendation.

How to use: Rank Option 1/2/3 by total score. Then apply these thresholds and override rules using each option's Mode (selected above).

Thresholds (guide, not law):

- Choose the top-ranked option if it is $\geq 15\%$ higher than the runner-up and its factors for Integration speed ≥ 2 and Ops capacity ≥ 2 (or are acceptable for the team).
- If the runner-up is within 10%, prefer Blend (run both in parallel with a reversible pilot in Section F) and decide after the delta report.
- If all three are within 10%, run a two-week bake-off across the top 2 and default to Blend until results diverge.

Mode-based override rules (apply before the final call):

- If any hard privacy/residency requirement exists, options with Mode = Open-Source Hosted or Edge-On-Device get priority unless a contractual Hosted API satisfies it.
- If Interactive p95 ≤ 500 ms is mandatory, options with Mode = Edge-On-Device (or Open-Source using a lightweight/distilled model) get priority unless quality deltas are proven in favor of Hosted.
- If Ops capacity is very low (factor score ≤ 1), short-term Hosted API is acceptable only with an explicit exit plan (exportable prompts/indices, quarterly re-eval).

Evidence Pack (attach or link)

What this is: The minimally sufficient proof.

How to use: Paste links; no slides required.

- Acceptance checks (copied from Worksheet)
- Golden set & delta report (Appx-D)
- Cost/latency envelope with thresholds (Appx-B)
- Security posture (DPAs, retention, export)
- Exit plan (prompts/indices export; rollback steps)

Pilot Plan (2 weeks, reversible)

What this is: A safe way to learn before committing.

How to use: Fill even if choosing Build or Buy—pilot proves/invalidates assumptions.

- Flag/Route name:
- Shadow \rightarrow Canary % \rightarrow GA:
- Rollback triggers: error rate $> _ _$; p95 latency $> _ _$; cost/1k $> _ _$; quality drop $> _ _$
- Owner(s):
- Demo date:

Decision & Sign-offs

What this is: The call and who's accountable.

How to use: Circle one; gather signatures.

Decision: Build Buy Blend

Rationale (bullets):

-

Accountable owner:

Approver(s):

Review date: (quarterly re-eval)

12-Week Upskilling Plan

A print-ready learning path that turns a capable teammate into an AI-ready contributor who ships value safely. Structured in three phases with weekly objectives, exercises, and checkpoints. Standalone for design/production; aligned with the main guide's S7.

How to use: Manager + mentor tailor the plan on Day 0. Keep the weekly cadence lightweight (≤ 2 hrs/week of formal work). Each week ends with a check-in and a signature. Artifacts live in the knowledge hub.

Setup & Roles

What this is: Who's involved and what success looks like.

How to use: Fill at kickoff; keep goals measurable and tied to a workflow.

- Participant:
- Manager:
- Mentor:
- Workflow/Pod:
- Target outcomes by Week 12 (3 bullets):

Phase Overview (0–2, 3–6, 7–12)

What this is: The arc of the program.

How to use: Skim weekly, but commit to the phase gates and artifacts.

- Phase 1 (Weeks 0–2): Literacy & Safety – foundations, Greenlight Rules, evaluation basics.
- Phase 2 (Weeks 3–6): Guided Experiments – run 2–3 small experiments end-to-end.
- Phase 3 (Weeks 7–12): Productionize & Reuse – ship behind a flag with telemetry; add one reusable pattern to the hub.

Week-by-Week Plan

What this is: Concrete weekly goals, exercises, and checklists.

How to use: Check off items, paste links, and capture notes. Keep to one page if printed.

<p>Week 0 – Kickoff & Access Goals: Align on outcomes; ensure tools access; schedule demos. Exercises: Read Preface + S1/S2 summaries; review Greenlight Rules. Checklist:</p> <ul style="list-style-type: none">• Access: repos · eval harness · retrieval store (read) · model gateway (dev)• Add to channels (#ai-demos, #ai-ops)• Book weekly 20-min mentor sync	<p>Week 1 – Evaluation Basics Goals: Understand golden sets, acceptance checks, and Decision Logs. Exercises: Fill a sample AI Experiment Worksheet; review a Decision Log exemplar. Checklist:</p> <ul style="list-style-type: none">• Create a tiny golden set (10 items)• Draft acceptance checks for a known workflow• Run the eval harness once• Artifacts/Links:• Sign-off (Mentor):
<p>Week 2 – Safety & Guardrails Goals: Apply Greenlight Rules and HITL gates; understand rollback. Exercises: Walk through a mock rollback; label a HITL moment. Checklist:</p> <ul style="list-style-type: none">• Artifacts/Links:• Sign-off (Mgr/Mentor):	<p>Week 3 – Experiment 1 (Scoping) Goals: Frame a two-week slice and start work. Exercises: Complete Worksheet; write hypothesis; set metrics. Checklist:</p> <ul style="list-style-type: none">• Worksheet approved by mentor• Demo date on calendar• Telemetry fields confirmed• Artifacts/Links:• Sign-off (Mentor):
<p>Week 4 – Experiment 1 (Run & Close) Goals: Run retrieval/model route; collect metrics; make a decision. Exercises: Human spot check; compile delta report. Checklist:</p> <ul style="list-style-type: none">• Demo performed• Decision Log filed (Ship/Shelve/Iterate)• Post to hub with links• Artifacts/Links:• Sign-off (Mentor):	<p>Week 5 – Experiment 2 (Scoping) Goals: Define a second slice (new domain or constraint). Exercises: Worksheet; acceptance checks tuned by Week 4 learnings. Checklist:</p> <ul style="list-style-type: none">• Worksheet approved• Demo scheduled• Risk notes added (cost/latency caps)• Artifacts/Links:• Sign-off (Mentor):
<p>Week 6 – Experiment 2 (Run & Close) Goals: Execute and close the loop faster; compare to Week 4. Exercises: Evaluate; practice presenting tradeoffs. Checklist:</p> <ul style="list-style-type: none">• Demo performed• Decision Log filed• Lessons captured in hub• Artifacts/Links:• Sign-off (Mentor):	<p>Week 7 – Productionization Prep Goals: Select one shipped/iterated idea for a small production pilot. Exercises: Write runbook; define flags; outline rollback triggers. Checklist:</p> <ul style="list-style-type: none">• Runbook drafted (version manifest included)• Feature flag configured• Canary thresholds defined• Artifacts/Links:• Sign-off (Pod Lead):

12-Week Upskilling Plan

A print-ready learning path that turns a capable teammate into an AI-ready contributor who ships value safely. Structured in three phases with weekly objectives, exercises, and checkpoints. Standalone for design/production; aligned with the main guide's S7.

How to use: Manager + mentor tailor the plan on Day 0. Keep the weekly cadence lightweight (≤ 2 hrs/week of formal work). Each week ends with a check-in and a signature. Artifacts live in the knowledge hub.

Setup & Roles

What this is: Who's involved and what success looks like.

How to use: Fill at kickoff; keep goals measurable and tied to a workflow.

- Participant:
- Manager:
- Mentor:
- Workflow/Pod:
- Target outcomes by Week 12 (3 bullets):

Phase Overview (0–2, 3–6, 7–12)

What this is: The arc of the program.

How to use: Skim weekly, but commit to the phase gates and artifacts.

- Phase 1 (Weeks 0–2): Literacy & Safety — foundations, Greenlight Rules, evaluation basics.
- Phase 2 (Weeks 3–6): Guided Experiments — run 2–3 small experiments end-to-end.
- Phase 3 (Weeks 7–12): Productionize & Reuse — ship behind a flag with telemetry; add one reusable pattern to the hub.

Week-by-Week Plan

What this is: Concrete weekly goals, exercises, and checklists.

How to use: Check off items, paste links, and capture notes. Keep to one page if printed.

Week 0 — Kickoff & Access

Goals: Align on outcomes; ensure tools access; schedule demos.

Exercises: Read Preface + S1/S2 summaries; review Greenlight Rules.

Checklist:

- Access: repos · eval harness · retrieval store (read) · model gateway (dev)
- Add to channels (#ai-demos, #ai-ops)
- Book weekly 20-min mentor sync
- Artifacts/Links:
- Sign-off (Mgr/Mentor):

Week 1 — Evaluation Basics

Goals: Understand golden sets, acceptance checks, and Decision Logs.

Exercises: Fill a sample AI Experiment Worksheet; review a Decision Log exemplar.

Checklist:

- Create a tiny golden set (10 items)
- Draft acceptance checks for a known workflow
- Run the eval harness once
- Artifacts/Links:
- Sign-off (Mentor):

Week 2 — Safety & Guardrails

Goals: Apply Greenlight Rules and HITL gates; understand rollback.

Exercises: Walk through a mock rollback; label a HITL moment.

Checklist:

- List data/PII handling for the workflow
- Define a feature flag + shadow→canary steps
- Identify at least one HITL gate
- Artifacts/Links:
- Sign-off (Mentor):

12-Week Upskilling Plan

Week 3 – Experiment 1 (Scoping)

Goals: Frame a two-week slice and start work.

Exercises: Complete Worksheet; write hypothesis; set metrics.

Checklist:

- Worksheet approved by mentor
- Demo date on calendar
- Telemetry fields confirmed
- Artifacts/Links:
- Sign-off (Mentor):

Week 4 – Experiment 1 (Run & Close)

Goals: Run retrieval/model route; collect metrics; make a decision.

Exercises: Human spot-check; compile delta report.

Checklist:

- Demo performed
- Decision Log filed (Ship/Shelve/Iterate)
- Post to hub with links
- Artifacts/Links:
- Sign-off (Mentor):

Week 5 – Experiment 2 (Scoping)

Goals: Define a second slice (new domain or constraint).

Exercises: Worksheet; acceptance checks tuned by Week 4 learnings.

Checklist:

- Worksheet approved
- Demo scheduled
- Risk notes added (cost/latency caps)
- Artifacts/Links:
- Sign-off (Mentor):

Week 6 – Experiment 2 (Run & Close)

Goals: Execute and close the loop faster; compare to Week 4.

Exercises: Evaluate; practice presenting tradeoffs.

Checklist:

- **Demo performed**
- **Decision Log filed**
- **Lessons captured in hub**
- **Artifacts/Links:**
- **Sign-off (Mentor):**

Week 7 – Productionization Prep

Goals: Select one shipped/iterated idea for a small production pilot.

Exercises: Write runbook; define flags; outline rollback triggers.

Checklist:

- Runbook drafted (version manifest included)
- Feature flag configured
- Canary thresholds defined
- Artifacts/Links:
- Sign-off (Pod Lead):

12-Week Upskilling Plan

Week 8 — Pilot Behind a Flag (Shadow)

Goals: Run in shadow; verify telemetry; harden UX affordances.

Exercises: Probe for failure modes (injection, latency spikes).

Checklist:

- Shadow live
- Telemetry dashboard visible
- Issues tracked with owners
- Artifacts/Links:
- Sign-off (Pod Lead):

Week 9 — Pilot (Canary)

Goals: Expose to 1–10% traffic; watch alerts; refine.

Exercises: Daily check of latency/cost/quality; tune cache/reranker.

Checklist:

- Canary live
- Alerts green or mitigated
- Customer feedback captured
- Artifacts/Links:
- Sign-off (Pod Lead):

Week 10 — GA Decision & Reuse

Goals: Make the GA call; extract a reusable pattern.

Exercises: Write “pattern card” (what/when/how/limits).

Checklist:

- Decision Log for GA vs. rollback
- Pattern card published in hub
- Present at Weekly Demo
- Artifacts/Links:
- Sign-off (Mgr/Lead):

Week 11 — Teach & Document

Goals: Share what you learned; improve a template.

Exercises: Lightning talk; PR to /Templates (rubric/checklist).

Checklist:

- 10-minute talk delivered
- Template improvement merged
- Mentor review complete
- Artifacts/Links:
- Sign-off (Mentor):

Week 12 — Capstone & Next Steps

Goals: Own a small win (or rollback) end-to-end and propose next quarter goals.

Exercises: Write a one-page “What I’d change next” note.

Checklist:

- Capstone complete
- Next-quarter goals set
- Promotion rubric discussion
- Artifacts/Links:
- Sign-off (Manager):

Metrics & Review

What this is: How we know the program worked.

How to use: Record before/after and review with manager.

- Cycle-time-to-insight (baseline → Week 12):
- Closed experiments (count):
- MTTR-AI (drill result):
- Reuse created (patterns adopted by ≥1 other pod):
- Self-assessment (confidence across S1–S7 topics):

KPI/OKR Worksheet

(Definitions + Targets)

A print-ready worksheet to define clear Objectives, attach measurable Key Results, and standardize KPI definitions for AI programs. Keeps goals legible from exec → pod → platform. Standalone for design/production; aligned with Weekly Rhythm, Showcase, Decision Log, and Postmortem templates.

How to use: Complete top-down each quarter. For each Objective, write 3–5 measurable Key Results with owners and target dates. Use the KPI Definitions section to lock wording and calculation rules (copy/paste-friendly, no formulas). Review monthly.

Org Snapshot (this quarter)

What this is: Context for the goals.

How to use: Fill before drafting Objectives.

Quarter:.....

Exec sponsor:.....

Product/POD(s):.....

Platform lead:.....

Security lead:.....

Themes: (e.g., reduce cost/1k, raise provenance, speed safe exposure)

Objectives (qualitative) – Company / Product / Platform

What this is: The 3–5 narrative outcomes you want, not tasks.

How to use: Write them short and human; no numbers in the Objective line.

O1 – Company-level

Objective

Why it matters: Key Results (3–5):

1.KR1 –

Owner:

Target date:

2.KR2 –

Owner:

Target date:

3.KR3 –

Owner:

Target date:

4.KR4 –

Owner:

Target date:

O2 – Product/POD

Objective

Why it matters: Key Results (3–5):

1.KR1 –

Owner:

Target date:

2.KR2 –

Owner:

Target date:

3.KR3 –

Owner:

Target date:

4.KR4 –

Owner:

Target date:

O3 – Platform/Operations

Objective

Why it matters: Key Results (3–5):

1.KR1 –

Owner:

Target date:

2.KR2 –

Owner:

Target date:

3.KR3 –

Owner:

Target date:

4.KR4 –

Owner:

Target date:

KPI/OKR Worksheet (Definitions + Targets)

KPI Targets (fill baseline → target)

What this is: The quarterly numbers you'll move.

How to use: Record last quarter's baseline; set a realistic target; assign an owner.

KPI	Baseline (last Q)	Target (this Q)	Owner	Review cadence
Cycle-time-to-insight (days to first result)				Weekly
Closed experiments (count)				Weekly
MTTA (acknowledge, minutes)				Weekly
MTTR-AI (restore, minutes)				Weekly
Cost per 1k calls (USD)				Weekly
Latency p95 (ms)				Weekly
Provenance rate (% answers with citations)				Weekly
AI adoption (active users or tasks/week)				Monthly
Reuse created (new patterns adopted by ≥1 other pod)				Monthly
Training/Enablement engagement (% completion)				Monthly

KPI Definitions (copy/paste friendly)

What this is: Plain-English rules so everyone calculates the same way.

How to use: Keep wording tight; avoid math notation.

Cycle-time-to-insight

- What it means: The average days from project kickoff to the first demo that shows real evidence (acceptance checks and a Decision Log filed).
- How to calculate: For each project this quarter, count the days between kickoff and the first evidence-based demo. Add all those days together. Divide by the number of projects.
- Scope: Only projects that reached a first demo this quarter.

Closed experiments

- What it means: The number of experiments that reached a decision (Ship, Shelve, or Iterate) and have a Decision Log on file.
- How to calculate: Count Decision Logs created this quarter that record one of those three outcomes.

KPI/OKR Worksheet (Definitions + Targets)

MTTA (Mean Time To Acknowledge)

- What it means: The average minutes between when an incident is detected and when a person acknowledges it.
- How to calculate: For each incident, subtract the detection time from the acknowledgment time to get minutes. Add up the minutes for all incidents this quarter. Divide by the number of incidents included.
- Scope: Include incidents at Medium severity or higher.

MTTR-AI (Mean Time To Restore – AI)

- What it means: The average minutes between when an AI incident is detected and when the feature is restored to an acceptable state (either rolled back or back within envelopes).
- How to calculate: For each AI incident, subtract the detection time from the restore time to get minutes. Add up the minutes for all AI incidents this quarter. Divide by the number of AI incidents included.
- "Restored" means: Quality/faithfulness, latency p95, cost per 1k, and error rate are back within agreed limits, or the feature is safely disabled or shadowed.

Cost per 1k calls

- What it means: The average dollars spent for every 1,000 model calls on this feature.
- How to calculate: Sum the total dollars spent on model calls for the feature this quarter. Divide by the total number of calls. Multiply by 1,000.

Latency p95

- What it means: The time in milliseconds that 95 out of 100 requests finish under.
- How to calculate: Sort all request latencies for the feature this quarter. Find the value that 95% of requests are faster than. Use that value as p95.

Provenance rate

- What it means: The percentage of answers that include citations to sources and timestamps where required.
- How to calculate: Count the number of user-visible answers this quarter that include proper citations. Divide by the total number of user-visible answers this quarter. Multiply by 100.

AI adoption

- What it means: How many people or workflows actively use the AI feature.
- How to calculate: Choose a consistent measure (active users per week or tasks per week). Count the unique users (or tasks) that used the feature in a typical week this quarter. Report the weekly average.

Reuse created

- What it means: Reusable patterns extracted that other pods adopted.
- How to calculate: Count the number of pattern cards published this quarter that were used by at least one other pod.

Training/Enablement engagement

- What it means: The percentage of required modules completed by the target audience.
- How to calculate: Count the total required modules completed by all people in the target group this quarter. Divide by the total required modules assigned. Multiply by 100.

KR → KPI Mapping

What this is: Connect each Key Result to the KPI(s) it moves.

How to use: For every KR above, list the KPI name(s) and update frequency.

Key Result	Related KPI(s)	Update cadence

KPI/OKR Worksheet (Definitions + Targets)

Review Cadence & Ownership

What this is: Who reviews what, and when.

How to use: Decide how often to check progress and where to report.

- Weekly Rhythm: PM reviews Closed experiments, Cycle-time-to-insight.
- Showcase (Monthly): Exec reviews AI adoption and Reuse created.
- Platform Review (Monthly): Platform reviews Latency p95, Cost/1k, MTTA/MTTR-AI.
- Quarterly Business Review: Full OKR readout and next-quarter targets.

Report location: Dashboard URL

Notes:

Sign-off

What this is: Make the quarter's plan official.

How to use: Collect signatures; post in the hub.

Signatures: Exec ____ PM ____ Platform ____ Security ____ Date ____