GOLDEN SOURCE • TRAINING BIBLE

PROJECT MANAGEMENT IN THE WORLD OF AI

The AI Project Manager Playbook

How to frame, fund, build, evaluate, govern, launch and scale artificial intelligence work - without pretending it is ordinary software.

> **The golden rule:** No data, no model. No evaluation, no launch. No owner, no scale. No value hypothesis, no funding.

For Senior and Lead Project and Programme Managers, Delivery Leads, Product Managers, Scrum Masters and sponsors working on machine learning, generative AI and agentic systems.

Version 1.0 • October 2026 • A living reference - review after every major release and every incident. Keep this document as the single source your team argues from, then update it when evidence changes.

START HERE

How to use this bible - and what "golden source" means

A golden source is not the longest document in the room. It is the one document everyone agrees to trust when memories differ, vendors promise differently, and deadlines get loud.

Use it in four ways

1. Learn

Read Parts 1-3 end to end once. They give you the vocabulary, lifecycle and judgement the rest assumes.

2. Run a project

Work through the gates in order: Intake, Frame, Data, Experiment, Build, Evaluate, Govern, Launch, Operate. Do not skip a gate to save a week; you will pay for it for a quarter.

3. Decide in the room

Open the relevant checklist, RACI or scorecard during the meeting. Record the decision, owner and date in the artefact itself.

4. Teach others

Each chapter ends with a teach-back prompt and a red-flag list. Use them in team training, interviews and retrospectives.

Non-negotiables in every chapter: name a single accountable owner, write the value hypothesis in numbers, define how quality will be measured before building, and plan for failure before launch.

What this playbook is not

It is not a data science textbook, a prompt library, or legal advice. Where regulation is summarised, treat the summary as a prompt to involve Legal, Privacy and Security - never as a substitute for them. Model names, vendor prices and tool features change monthly; principles, gates and artefacts in this book are designed to survive those changes.

Conventions

GOLDEN RULE = a rule you break only with a written sponsor decision. **Gate** = a point where work may not proceed without evidence. **Artefact** = a document that must exist, have an owner and be kept current.

Contents

Part 1	Foundations every PM must own	4-7
1	AI in plain language: ML, GenAI, agents	4
2	The AI lifecycle is not the SDLC	5
3	The AI PM: role, skills and authority	6
4	Portfolio intake and prioritisation	7
Part 2	Framing, data and building	8-13
5	Discovery and problem framing	8
6	Data: readiness, quality, rights	9
7	Build, buy or partner	10
8	Experimentation and evaluation	11
9	Delivery operating model for AI teams	12-13
Part 3	Trust, launch and scale	14-21
10	Governance, risk and regulation	14
11	Privacy, security and safety	15
12	MLOps, LLMOps and deployment	16
13	Adoption, change and training	17
14	Stakeholders and communication	18
15	Money: budgets, vendors, unit economics	19
16	Metrics and value realisation	20
17	Programme management at scale	21
Part 4	The golden source toolkit	22-31
18	Failure modes and anti-patterns	22
19	Templates: charter, RACI, RAID, registers	23-26
20	Checklists: launch, eval, vendor, incident	27
21	Your first 90 days as an AI PM	28
22	Career, interviews and continuous learning	29
Appendices	Glossary, decision logs, prompt patterns for PMs	30-31

> **Reading time:** About 3 hours cover to cover. In a live programme, most PMs live in Chapters 5, 8, 10 and 19-20.

AI in plain language: what you are actually managing

Traditional software follows rules a person wrote. AI systems learn patterns from examples, or generate new content from patterns learned at scale. That single difference explains almost every delivery surprise that follows.

Three families - know which one is in your plan

Family	What it does	Typical uses	PM watch-out
Predictive ML	Predicts a value, class or ranking from inputs	Fraud scores, churn, forecasting, risk ratings	Needs labelled historical data; performance drifts as the world changes
Generative AI	Creates text, code, image, audio or structured output	Drafting, summarisation, search (RAG), copilots, coding assistants	Output varies run to run; fluent is not the same as correct
Agentic AI	Plans steps and calls tools/systems to complete a task	Workflow automation, research agents, service agents	Errors compound across steps; autonomy needs permissions, limits and audit trails

Vocabulary you must use precisely

- **Model:** the trained system that produces outputs. **Prompt:** the instruction and context you give it. **Context window:** how much it can consider at once.
- **Training vs inference:** training builds the model (expensive, occasional); inference is using it (every request, ongoing cost).
- **Fine-tuning vs RAG:** fine-tuning changes behaviour with examples; retrieval-augmented generation (RAG) supplies current, citable facts at query time. Most business use cases need RAG first, fine-tuning later, if ever.
- **Hallucination:** plausible output that is wrong or invented. Manage it with grounding, citations, evaluation and human review - never with hope.
- **Tokens:** the units GenAI is priced and limited by. Longer prompts and outputs cost more and run slower.

> **Red flag:** A plan that says "the AI will learn our business" with no named data source, no owner for that data, and no evaluation method. That is a wish, not a plan.

Teach-back: Explain to a sponsor, in 60 seconds and without jargon, which family your project uses, what it cannot do, and how you will know when it is wrong.

The AI lifecycle is not the SDLC - plan for loops, not lines

AI delivery is experimental by nature. You do not know feasibility, quality or cost until you test with real data. A plan that promises a fixed date for an untested model is fiction with a Gantt chart.

The nine gates

Gate	Question it answers	Evidence to pass
1 Intake	Is this worth exploring?	Scored intake form, sponsor named
2 Frame	What problem, for whom, measured how?	Problem statement, baseline, success metric
3 Data	Do we have lawful, usable data?	Data readiness assessment, rights confirmed
4 Experiment	Can it work at all?	Time-boxed spike with results, even if negative
5 Build	Can it work reliably for real users?	Working system in staging, eval suite running
6 Evaluate	Is it good, safe and fair enough?	Eval scorecard vs agreed thresholds
7 Govern	Are risk, privacy and security signed off?	Risk tier, approvals, residual risk accepted
8 Launch	Can we release gradually and roll back?	Rollout plan, monitoring, support rota
9 Operate	Is it still good next quarter?	Drift monitoring, retraining and review cadence

How estimation changes

- Estimate discovery and data work in time-boxes with exit criteria, not fixed scope.
- Carry explicit contingency for data remediation - in most programmes it is the largest unknown.
- Separate *commit dates* (infrastructure, integrations, training, comms) from *discovery dates* (model quality milestones). Commit only to what you control.
- Report confidence, not just status: "Quality at 82% against a 90% gate, two data fixes in flight, confidence medium."

> **GOLDEN RULE:** Every AI plan has a kill / pivot point written in advance. Deciding in advance what evidence would stop the project is what makes stopping possible later.

PART 1 • FOUNDATIONS

The AI PM: role, skills and authority

The AI PM is not a note-taker between business and data science. You own the translation, the trade-offs and the evidence - and you need enough technical fluency to challenge both sides respectfully.

What you own

Value

The business case, baseline, benefits owner and the honest re-forecast when evidence changes.

Evidence

Evaluation design, thresholds, test data integrity and the go / no-go recommendation.

Risk

The risk register, governance route, incident readiness and the record of who accepted what risk.

Adoption

Workflow redesign, training, communications and the feedback loop after launch.

Competency map

Domain	Working level means you can...
AI / data literacy	Read a model card and an eval report, question a data pipeline, spot leakage and bias risks
Delivery	Run iterative delivery, manage dependencies, protect a team from thrash
Commercial	Model unit economics, negotiate vendor limits, write a build-buy case
Risk and governance	Tier a use case, run a DPIA conversation, prepare an audit trail
Change	Map affected roles, design training, measure adoption honestly

Authority you need in writing

- Authority to pause a launch that fails an agreed quality or safety gate.
- A named sponsor who decides trade-offs within one business day.
- Access to real users and real (or realistically masked) data early - not in UAT week.

Teach-back: Name the last decision you escalated that should have been yours. Fix the delegation, not just the decision.

Portfolio intake and prioritisation: fund fewer, better bets

AI portfolios fail by starting everything. Intake is where a PM earns their salary: turning enthusiasm into a ranked, resourced, killable pipeline.

The intake form - no form, no meeting

- Problem and affected users, in one paragraph. Current process and its cost or pain, with a baseline number.
- Proposed AI approach and family (predictive / generative / agentic). What happens if the model is wrong?
- Data needed, its owner, and whether rights to use it are confirmed or assumed.
- Sponsor, benefits owner, and the team or vendor expected to build it. Rough cost range and ongoing run cost.

Scoring model (score 1-5 each, discuss, do not average away disagreement)

Criterion	5 looks like	1 looks like
Value	Large, measurable, owned benefit	"Efficiency" with no baseline
Feasibility	Proven pattern, data ready	Research problem, data absent
Data readiness	Clean, lawful, accessible now	Unknown owner, rights unclear
Risk (reverse)	Internal, reversible, low harm	Customer-facing, regulated, irreversible
Adoption ease	Users asking for it, workflow simple	Role redesign, resistance expected

> **Portfolio rule:** Limit work in progress. One team finishing one use case teaches the organisation more than five teams demoing five pilots. Pilot purgatory - endless pilots, no production - is a portfolio management failure, not a technology failure.

Intake outcomes

Only four are allowed: **Fund** (with gates and a kill point), **Spike** (time-boxed feasibility, typically 2-6 weeks), **Queue** (ranked, with the reason), or **Decline** (with the reason recorded). "Maybe later" without a rank and a reason is a hidden yes that consumes attention.

Discovery and problem framing: solve the workflow, not the demo

Most AI projects that disappoint were framed around a technology demo instead of a job in a real workflow. Discovery is where you earn the right to build.

Discovery sprint plan (2-4 weeks)

1. **Observe:** shadow 5-10 real users doing the task today. Time it. Note workarounds and exceptions.
2. **Baseline:** capture current volume, cycle time, error rate and cost. Without a baseline, benefits are theatre.
3. **Frame:** write the problem statement (template in Chapter 19). Define the human decision the AI informs or makes.
4. **Bound:** list explicitly what is out of scope, which cases always go to a human, and what "good enough" means numerically.
5. **Prototype the riskiest assumption first:** usually data quality or output quality on hard cases, not the happy path.

Questions that change projects

- What decision improves if this works? Who makes it today, and will they trust a machine's input?
- What is the cost of a false positive versus a false negative? They are rarely equal, and the threshold should reflect that.
- Which 10% of cases cause 90% of the difficulty? Evaluate on those, not just the average.
- If the model disappeared tomorrow after launch, what breaks? That answer is your dependency and fallback design.

> **Anti-pattern:** "Human in the loop" as a slogan. Specify the loop: who reviews, what percentage, with what time budget, what they can override, and how overrides are logged and fed back.

Discovery exit criteria

Signed problem statement, baseline pack, user list for pilot, data readiness verdict, eval approach drafted, risk tier proposed, and a costed next phase with a kill point. If any item is missing, discovery is not done - it is paused.

Data is the product: readiness, quality and rights

Ask any experienced AI PM where the time went and the answer is data: finding it, accessing it, cleaning it, labelling it, and proving you may use it. Plan for data as a workstream with its own lead, milestones and risks.

Data readiness assessment - score before you commit

Dimension	Check
Availability	Does the data exist, where, in what system, at what volume and history?
Access	Who approves access, how long does approval take, dev / test / prod differences?
Quality	Completeness, duplication, consistency, known errors, how labels were produced
Representativeness	Does it cover all user groups, regions, edge cases and time periods in scope?
Rights and consent	Lawful basis, licence, retention, whether vendor training on your data is excluded contractually
Freshness	How current must it be, how is it refreshed, who owns the pipeline when it breaks?

Data work to schedule explicitly

- Discovery and profiling, access approvals, extraction and pipeline build, cleaning and de-duplication, labelling and expert review, train / validation / test splits with leakage checks, a frozen golden test set that the build team cannot tune against.
- Synthetic data may supplement training or testing, but never replaces evaluation on real cases.
- Data versioning: every eval result names the data version, model version and prompt version that produced it. Results without versions cannot be reproduced or trusted.

> **GOLDEN RULE:** The golden test set is governed like money: controlled access, change log, independent owner. If the team can quietly edit the exam, every grade is meaningless.

Build, buy or partner: decide with total cost and exit in mind

Speed to demo is the worst basis for this decision. Decide on fit, total cost over three years, control of data and roadmap, and how you leave if it fails.

Decision matrix

Route	Choose when	Hidden cost to price in
Buy (SaaS / platform)	Commodity capability, speed matters, internal skills scarce	Per-seat / per-call growth, customisation limits, exit and data export
Build on foundation models / APIs	Differentiating workflow, you own data and UX	Eval, monitoring, prompt and model upgrades, run cost per transaction
Build own models	Unique data advantage, scale justifies MLOps investment	Talent, labelling, infrastructure, retraining forever
Partner / consultancy	Capacity or specialist skill gap, with knowledge transfer contracted	Dependency, IP terms, handover quality

Vendor evaluation essentials

- Evaluate on *your* golden test set, in your environment, with your hard cases - never on the vendor's demo data alone.
- Contract for: no training on your data (or explicit opt-in), data residency and retention, security evidence, model-change notice periods, usage caps and price-change limits, export of your data and prompts, and termination assistance.
- Test the exit during selection: can you export everything, in usable form, within a stated period?

Proof of concept discipline

Every PoC has a hypothesis, a fixed time-box, success thresholds agreed *before* results, a named decision-maker, and a decision date. A PoC without a decision date becomes a permanent pilot.

Experimentation and evaluation: no eval, no launch

A demo shows the model can work once. Evaluation shows how often it works, where it fails, for whom it fails, and whether it is better than what you have today.

Build the eval before the feature

1. Define metrics: task quality (accuracy, completeness, groundedness), safety (harmful or policy-violating output rate), fairness (performance by relevant group), operational (latency, cost per successful task, failure rate).
2. Build the golden set: representative real cases plus edge, adversarial and previously failed cases. Size it so results are meaningful, and refresh it as real usage teaches you.
3. Set thresholds with the sponsor and risk owner before seeing results. Record them in the eval scorecard (Chapter 19).
4. Combine automated scoring, model-assisted scoring (with human spot-checks - a model marking its own homework needs supervision) and structured human review.
5. Evaluate the *system* - model, prompts, retrieval, guardrails and UX together - not the model alone.

Reading an eval report like a PM

- Ask for performance by segment and by case difficulty, not only the headline average.
- Read 20 raw failures yourself. Categories in a chart hide tone, harm and absurdity that one page of examples reveals.
- Compare against the human baseline and the current process, not against perfection.
- Track regression on every change: model upgrade, prompt edit, data refresh. No change ships without re-running the suite.

GOLDEN RULE: Thresholds are set before results, signed by the sponsor, and changed only through the decision log - never quietly after a miss.

Delivery operating model: how AI teams actually work

AI teams need agile delivery with experimental honesty: short cycles, visible evidence, and ceremonies that discuss failures as data rather than blame.

Team shape (adapt to scale, keep the roles)

Role	Accountable for
Sponsor	Value, funding, trade-off decisions, risk acceptance
AI PM / Programme Mgr	Plan, gates, evidence, stakeholders, benefits
Product Manager	User outcomes, workflow, prioritisation
Data / ML Lead	Data pipelines, model approach, eval integrity
Engineering Lead	Architecture, integration, reliability, security build
Design / Research	Workflow UX, trust cues, user testing
Risk partners	Legal, Privacy, Security, Compliance - engaged from framing, not at the end
Change / Training Lead	Adoption plan, training, support model

Cadence that works

- **Weekly:** delivery stand-up plus a 30-minute evidence review - eval results, failures read aloud, decisions logged.
- **Fortnightly:** demo to real users on real cases, including one failure case by design.
- **Monthly:** steering with value, quality, risk and cost on one page (template in Chapter 19).
- **Per gate:** formal go / no-go with the scorecard, not a verbal assurance.

Backlog hygiene for AI work

Label items by type: data, experiment, build, eval, governance, adoption. A backlog that is 90% build items is a warning sign - the other types are where AI projects actually succeed or fail. Cap experiment work in progress; finish and learn, or stop and record why.

Estimation, planning and the honest roadmap

AI roadmaps earn trust by separating what is known from what is being discovered, and by updating visibly when evidence arrives.

Plan in horizons

Horizon	Commitment level	What belongs there
Now (0-6 weeks)	Committed	Named work, owners, dates for controllable items
Next (6-12 weeks)	Planned, confidence-rated	Gate outcomes, ranges, dependencies
Later	Directional	Outcomes and hypotheses, no false dates

Estimation techniques that survive contact with reality

- Reference-class forecasting: how long did data access, labelling and integration take on the last comparable project - inside or outside your organisation?
- Three-point estimates for unknowns, with the pessimistic case discussed out loud and the assumption behind it recorded.
- Buffer where the uncertainty is (usually data and evaluation), not spread thinly everywhere where no one can see it.
- Re-plan triggers written in advance: for example, "if data readiness is still red at week 4, we re-scope to the smaller data set and re-baseline."

Status reporting that helps

Report four numbers every cycle: value progress (against baseline), quality (eval vs threshold), cost (spend and unit cost trend), and risk (top risk, its owner, its date). Add one paragraph: what we learned, what changed, what we need decided. RAG status without those numbers is decoration.

> **Stakeholder sentence to practise:** "We are green on delivery, amber on quality, and here is the evidence and the decision I need from you by Friday." That sentence builds more credibility than a year of green dashboards.

Governance, risk and regulation: build trust into the gates

Governance done at the end is a delay. Governance built into intake, framing and evaluation is how AI work moves fast without becoming the incident everyone remembers.

Risk-tier every use case at intake

Tier	Examples of characteristics	Governance
Low	Internal, reversible, human decides, low sensitivity data	Team review, standard checklist, log in AI inventory
Medium	Customer-visible, personal data, material decisions supported	Privacy and security review, eval thresholds signed, monitoring plan
High	Decisions affecting rights, money, health, safety or employment; regulated activity; vulnerable users	Formal risk assessment, independent review, sponsor risk acceptance, audit trail, periodic re-approval

Core governance artefacts

- **AI inventory:** every system, its owner, tier, data, vendor/model, and review date. If it is not in the inventory, it does not launch.
- **Model / system card:** purpose, data, performance by segment, limits, known failure modes, human oversight design.
- **Decision log and risk acceptance:** who decided, on what evidence, what residual risk they accepted, and when it is reviewed.

Regulatory landscape - know the shape, get advice on the detail

Frameworks such as the NIST AI Risk Management Framework (Govern, Map, Measure, Manage) give a voluntary structure many organisations adopt. The EU AI Act introduces risk-based obligations phased in over time, with the heaviest duties on high-risk systems and transparency duties for generative AI. Sector regulators (financial services, health, employment) apply existing duties - fairness, explainability, record-keeping, model risk management - to AI systems. Your job as PM is to tier early, involve Legal and Compliance at framing, and keep the evidence trail those regimes expect: data provenance, test results, human oversight, and change history.

> **Never do:** launch a high-tier system on a low-tier approval because the deadline moved. Escalate the tier question; it is exactly the decision governance exists for.

Privacy, security and safety: the PM's working checklist

You do not need to be a lawyer or a security engineer. You need to ask the right questions early, record the answers, and refuse to let them be deferred past the gate where they matter.

Privacy questions

- What personal data is used, why is each element necessary, and what is the lawful basis? Can we minimise, mask or aggregate instead?
- Where does data go - which vendors, which regions, for how long, and is it used to train anyone else's model?
- How are data-subject rights handled: access, correction, deletion - including in logs, prompts and vector stores?
- Is a privacy impact assessment required, who owns it, and what is its gate date?

Security questions

- Threat model for AI-specific attacks: prompt injection, data exfiltration through outputs, poisoned documents in retrieval, and excessive agent permissions.
- Identity and access: least-privilege tool permissions for agents, secrets handling, environment separation, audit logging of prompts, outputs and tool calls.
- Output handling: how are secrets, personal data and harmful content detected and blocked before reaching users or downstream systems?

Safety and human oversight design

- Define prohibited outputs and uses, escalation routes for harmful content, and the kill switch: who can suspend the system, how fast, and how users are told.
- Design review queues humans can actually handle: volume, time per case, training, and what happens when the queue overflows.
- Test with adversarial cases before launch (red-teaming) and log every safety incident with a severity, owner and fix date.

> **GOLDEN RULE:** An agent gets the minimum permissions for the task, for the minimum time, with every action logged. Convenience is not a permission model.

MLOps and LLMOps: launch is the start, not the finish

Models degrade, prompts drift, vendors update models underneath you, and usage grows into cases you never tested. Operations is where AI value is protected - budget and staff it in the original business case.

What must exist before launch

- **Versioning:** model, prompt, retrieval index, data and guardrail versions recorded for every output in logs.
- **Monitoring:** quality sampling on live traffic, drift indicators, latency, error and cost alerts, with thresholds and named responders.
- **Feedback loop:** user corrections and overrides captured, triaged weekly, and fed into the eval set and backlog.
- **Change control:** vendor model changes treated as releases - notice, re-evaluation on the golden set, staged rollout, rollback plan.
- **Fallback:** the manual or previous process kept warm for a defined period, with a rehearsed switch.

Release patterns

Pattern	Use when
Shadow mode (runs, not shown to users)	Comparing against human decisions safely before exposure
Pilot cohort (small, supported, opted-in users)	First real exposure, high learning per user
Canary / phased rollout	Scaling with quality gates between phases
Full release with review queue	Only after eval, safety and support are proven at pilot scale

Incident management for AI

Define AI incident severities in advance (wrong outputs at scale, harmful content, data leakage, runaway agent actions, cost spikes). The playbook is: detect, contain (throttle, disable feature, switch to fallback), communicate, preserve logs, root-cause, add the failure to the golden set, and re-evaluate before restoring. Practise it once before you need it.

Adoption, change and training: where ROI is decided

Users do not adopt models; they adopt new workflows that make their day better and their judgement safer. A technically successful launch that nobody uses is a failed project with good metrics.

Adoption plan essentials

1. **Impact map:** whose job changes, what stops, what starts, what gets harder before it gets easier. Name fears honestly, including job-security fears - unspoken, they become resistance; spoken, they become design inputs.
2. **Workflow redesign:** rewrite the procedure, not just the tool guide. Where does AI output enter, who checks it, where does accountability sit?
3. **Training by role:** users (use and verify), reviewers (judge and override), managers (interpret metrics, coach quality), support (triage AI incidents).
4. **Trust calibration:** teach known limits with real failure examples. Over-trust and under-trust are both adoption failures.
5. **Champions and feedback:** early users with a direct line to the team, visible fixes from their feedback in the first month.

Measure adoption honestly

- Active use by target users, task completion through the new workflow, override and correction rates, time saved measured (not surveyed once and assumed forever), and quality of outcomes downstream.
- Watch for shadow workarounds: exports, copy-paste to personal tools, and parallel spreadsheets are data about your workflow design.

> **Launch rule:** No launch without trained reviewers, a staffed support route, and a named adoption owner for the first 90 days after go-live.

Stakeholders and communication: translate in both directions

AI programmes have more stakeholder types, more jargon and more fear than ordinary programmes. Your communication plan is a delivery tool, not an afterthought.

Stakeholder map and what each needs

Audience	They need	Cadence / format
Sponsor / steering	Value, quality, risk, cost, decisions	Monthly one-pager + gate papers
Business users	What changes for me, training, support	Demos, office hours, short guides
Technical teams	Decisions, priorities, dependencies	Weekly evidence review
Risk partners	Early sight, evidence, honest gaps	From framing; gate reviews
Customers / public (if exposed)	Clear disclosure that AI is used, how to reach a human	In-product disclosure, help content

Communication principles

- Say what the system cannot do, plainly, before someone discovers it publicly.
- Use the same four numbers (value, quality, cost, risk) in every update so trends are visible.
- Write the difficult update early: a quality miss explained with a plan builds more trust than a surprise at the gate.
- Keep a single decision log and link every update to it. Memory is not a governance system.

Difficult conversations - scripts that work

To a sponsor pressing for a date: "I can commit to the integration and training dates today. The launch date depends on the quality gate on [date]; here is the current evidence and what would have to be true to commit." **To a vendor:** "We will evaluate on our test set, in our environment. If the results differ from the demo, we will work from our results." **To a worried team:** "Here is exactly which tasks change, which do not, and how we will measure whether this helps or harms your work."

Money: budgets, vendors and unit economics

AI costs behave differently: they scale with usage, spike with experiments, and continue forever in operations. A PM who cannot explain cost per successful outcome will lose control of the programme.

Budget structure

Line	Include
Discovery and data	Access, cleaning, labelling, expert review time
Build	Team, platform, integration, eval tooling
Run (per period)	Inference / licence per volume, monitoring, support, retraining, re-evaluation
Change	Training, comms, backfill for reviewers and champions
Risk and exit	Assessments, audits, contract exit and migration reserve

Unit economics to track from pilot

- Cost per request, cost per *successful* task (divide by the success rate - a cheap model that fails half the time is expensive), human review cost per case, and total cost per business outcome (per resolved case, per approved document, per saved hour verified).
- Model and route optimisation is a backlog item with an owner: caching, smaller models for easy cases, prompt discipline, and volume tiers - reviewed monthly, never by silently lowering quality.
- Set cost alerts and usage caps before launch. Runaway agents and retry loops are budget incidents, not curiosities.

Benefits discipline

Separate hard savings (budget actually removed), capacity release (time redirected, named to what), revenue effects (with the business owner's method), and risk reduction (valued by the risk owner). Report them separately forever; blending them is how business cases become unfalsifiable.

Metrics and value realisation: prove it, then improve it

If you cannot show the before, the after and the method, you have an anecdote. Value realisation is a workstream that runs from baseline to benefits review, months after launch.

The metric tree

Level	Examples	Owner
Outcome	Cycle time, cost per case, error rate, revenue, risk events	Benefits owner (business)
Adoption	Active users, workflow completion, override rate	Change lead
Quality	Eval scores by segment, live sampled quality, safety rate	Data / ML lead
Operational	Latency, uptime, cost per successful task, incidents	Engineering lead

Value realisation rhythm

1. Baseline signed before build. Method agreed with Finance or the benefits owner.
2. Pilot readout: outcome movement in the pilot cohort vs a sensible comparison, with limitations stated.
3. Scale readout at 90 days: adoption-adjusted benefit (benefit only counts for work actually flowing through the system).
4. Benefits review at 6 and 12 months: continue, optimise, expand or retire. Retiring a system that no longer earns its run cost is good management, not failure.

> **Vanity metrics to refuse:** prompts sent, licences distributed, demo applause, and "hours saved" multiplied from a single self-reported estimate. Replace each with a measured workflow metric.

Teach-back: Present your programme's value in three sentences: baseline, current measured result, and the method. If the third sentence is missing, fix the measurement before the next steering meeting.

Programme management at scale: many use cases, one system

Scaling from one use case to a programme adds platform choices, shared data, competing priorities and cumulative risk. The PM's job shifts from delivering a system to running a system of delivery.

What changes at programme level

- **Platform thinking:** shared components (identity, logging, eval tooling, guardrails, model gateway) built once, governed once, reused - with a platform roadmap and internal "customers."
- **Portfolio cadence:** quarterly intake and reprioritisation, WIP limits per team, and a visible kill list. Stopping work publicly and kindly is a programme skill.
- **Dependency management:** shared data owners, shared reviewers (Legal, Privacy, Security are scarce - schedule them like a critical resource), and shared vendor contracts.
- **Cumulative risk:** ten medium-tier systems touching the same customers can equal one high-tier exposure. Review the portfolio's combined footprint, not only each system alone.

Programme artefacts

Portfolio dashboard (value, quality, cost, risk per use case), platform roadmap, shared risk and decision logs, vendor and model inventory, and a benefits roll-up that never double-counts the same hour saved by two systems.

Operating rhythm

Weekly delivery sync across workstreams (dependencies and blockers only), monthly portfolio steering (four numbers per use case, decisions on funding and kills), quarterly strategy review (are we solving the right problems, is the platform earning its cost, what does the risk picture look like as a whole).

> **Scale rule:** Standardise the gates, the artefacts and the metrics. Let teams vary the methods inside them. Uniform process with local judgement scales; uniform process without judgement becomes theatre.

Failure modes and anti-patterns: the field guide to what goes wrong

Most AI failures are not exotic. They are the same dozen patterns repeating. Learn to name them in the room, early, while they are still cheap.

Pattern	Symptom	PM intervention
Demo-driven development	Happy-path demos, no hard cases	Insist demos include failures; build golden set now
Data surprise	"We just need the data" for months	Data workstream, readiness gate, re-scope trigger
Pilot purgatory	Pilots multiply, nothing scales	Decision dates, scale criteria at pilot approval
Eval theatre	High scores nobody can explain	Independent golden set, read raw failures
Vendor lock by accident	No export, no exit terms, price shock	Exit tested at selection, contract terms in Ch. 7
Automation without oversight	Review queue overwhelmed or rubber-stamping	Redesign sampling, time budgets, override logging
Shadow AI	Staff using personal tools with company data	Provide a sanctioned route fast; policy plus enablement
Model-change surprise	Quality drops after vendor update	Change control, re-eval on every model change
Benefits fiction	Savings claimed, budgets unchanged	Separate benefit types; Finance-agreed method
Agent overreach	Agent acts beyond intent	Least privilege, step limits, approval for irreversible actions

Using this table: In retrospectives, ask the team to pick the pattern closest to what happened - without blame - and name the gate or artefact that would have caught it earlier. Then update that artefact. That loop is how a playbook stays golden instead of going stale.

Template A: AI project charter and problem statement

One page, signed, and re-read at every gate. If the charter and the work disagree, stop and fix one of them.

Field	Write this
Problem statement	[Users] struggle to [task] because [cause], costing [baseline metric] per [period].
Decision improved	The specific decision or output, and who owns it today
Success metric + target	Primary outcome metric, baseline, target, measurement method, review date
Quality threshold	Eval metrics and minimums required to launch (from scorecard)
Users and scope	In scope, out of scope, cases that always go to a human
Data	Sources, owners, rights status, golden set owner
Risk tier	Tier, assessor, approvals needed, residual risk acceptor
Team and sponsor	Names, not just roles; sponsor decision SLA
Gates and kill point	Gate dates, evidence required, the condition that stops the project
Costs	Build cost, run cost per period, unit cost assumption

Template B: Decision log (one row per decision)

Date	Decision	Options considered	Decider	Evidence	Review date

Keep both on one page each. A charter nobody can find, or a log nobody updates, does not exist for governance purposes.

Template C: RACI for an AI programme

Agree this before work starts. Most AI friction is unowned work - especially evaluation, data rights and post-launch monitoring.

Activity	Sponsor	AI PM	Data/ML	Eng	Risk partners
Business case and benefits	A	R	C	C	C
Problem framing	A	R	C	C	C
Data access and rights	I	A	R	C	A (privacy)
Golden test set	I	A	R	C	C
Eval thresholds	A	R	R	C	C
Go / no-go at launch gate	A	R (recommendation)	C	C	A (their domains)
Incident response	I/A by severity	R (coordination)	C	R (technical)	C
Benefits realisation	A	R	C	I	I

R = Responsible (does it), A = Accountable (owns the outcome, one per row), C = Consulted, I = Informed.

Template D: RAID entries that actually help

- **Risk:** "If [cause], then [event], causing [impact on value/quality/date]. Owner, mitigation, contingency, trigger date."
- **Assumption:** state it, name who confirmed it, and the date it must be re-checked. Unchecked assumptions become issues.
- **Issue:** impact in numbers, decision needed, decider, deadline. An issue without a decider is a worry, not an issue.
- **Dependency:** who provides what, by when, with the escalation path agreed in advance.

Template E: Evaluation scorecard and gate paper

The single most important template in this book. Complete it before the gate meeting; the meeting decides, it does not discover.

Measure	Threshold (pre-agreed)	Result	Evidence / version
Primary task quality			Eval run ID, data / model / prompt versions
Quality - lowest segment			Segment breakdown attached
Safety violation rate			Red-team and live-sample method
Groundedness / citation accuracy (if RAG)			Sample size and reviewer
Latency (p95)			Load conditions
Cost per successful task			Volume assumption
Human baseline comparison			Baseline pack reference

Gate paper structure (2 pages maximum)

1. Recommendation: go / conditional go / no-go, in the first line.
2. Scorecard results against thresholds, with failures named.
3. Top residual risks and who is asked to accept each.
4. Rollout, monitoring, support and rollback readiness.
5. Conditions if conditional: owner and date for each, and what happens if a condition slips.

> A gate paper that hides a failed threshold in an appendix is a governance failure. Put misses in the summary; sponsors respect candour and punish surprises.

Template F: Risk register for AI systems

Start every register from this list, then add what is specific to your system. Review at every gate and monthly in operation.

Risk	Typical mitigation
Wrong outputs cause business or customer harm	Thresholds, human review design, phased rollout, fallback
Bias / unfair outcomes by group	Segment evaluation, representative data, fairness review, monitoring
Personal or confidential data leakage	Minimisation, masking, vendor terms, output filtering, logging controls
Prompt injection / adversarial manipulation	Input controls, least-privilege tools, red-teaming, output validation
Model or vendor change degrades quality	Change notice terms, re-eval gate, version pinning where possible
Drift as data and behaviour change	Drift monitoring, scheduled re-evaluation, retraining plan
Runaway cost (usage, loops, retries)	Caps, alerts, rate limits, cost per outcome dashboard
Over-trust by users	Training with failure examples, confidence cues, verification steps
Regulatory or policy breach	Early tiering, Legal review, inventory, audit trail
Key-person / vendor dependency	Documentation, knowledge transfer terms, exit plan tested

For each risk record: likelihood, impact, owner, mitigation status, residual risk, acceptor and review date. A register without acceptors and dates is a list of anxieties.

Checklists: launch, vendor and incident

Print these. Check them in the meeting, with evidence, not from memory.

Launch readiness checklist

[ ] Eval scorecard passed or residual risk formally accepted

[ ] Privacy, security and governance approvals recorded

[ ] Monitoring, alerts and dashboards live and tested

[ ] Support rota, incident severities and kill switch rehearsed

[ ] Reviewers trained, queue capacity tested at expected volume

[ ] Rollback / fallback process staffed and timed

[ ] User comms, disclosure and help content published

[ ] Cost caps and usage alerts set

[ ] AI inventory and model card updated

[ ] 30 / 60 / 90-day adoption and benefits reviews scheduled

Vendor selection checklist

[ ] Evaluated on our golden set, our hard cases, our environment

[ ] No-training-on-our-data, residency, retention and deletion terms written

[ ] Model-change notice and price-change limits agreed

[ ] Security evidence reviewed by Security, not filed unread

[ ] Export and exit tested; termination assistance defined

[ ] Total 3-year cost modelled at realistic volume growth

AI incident first-hour checklist

[ ] Severity declared, incident owner named, clock started

[ ] Contain: throttle, disable, or fall back - decide within the hour

[ ] Logs, prompts, outputs and versions preserved

[ ] Affected users / customers identified; comms owner assigned

[ ] Safety / privacy leads notified if harm or data is involved

CAREER • GOLDEN SOURCE

Your first 90 days as an AI PM

Whether the role is new or the organisation is, the first 90 days decide whether you are seen as the person who brings evidence or the person who brings status slides.

Days 1-30: Learn and inventory

- Map the portfolio: every AI system and pilot, its owner, tier, stage and spend. Expect surprises; shadow AI and forgotten pilots are normal findings.
- Meet the data owners, risk partners and real users before the executives. Learn where data lives and where work actually happens.
- Read the last eval reports, incident logs and benefits claims. Note what is measured and what is asserted.
- Deliverable: a one-page portfolio picture with the three biggest risks and three quick wins.

Days 31-60: Establish the gates

- Introduce intake, the golden set principle and the eval scorecard on one live project first - prove the pattern before mandating it.
- Agree risk tiering and the AI inventory with Legal, Privacy and Security.
- Fix one reporting rhythm: the four numbers, monthly, same format.

Days 61-90: Prove value

- Take one use case through a proper gate with a written paper and a clear decision.
- Publish a baseline-and-benefits pack for the flagship project.
- Run the first retrospective against the failure-mode table (Chapter 18) and update this playbook with what you learned.

> **90-day test:** Can any stakeholder, asking you any week, get the same four numbers, the current gate status, and the next decision with its date? If yes, you have built the operating system. Everything else is scale.

CAREER • GOLDEN SOURCE

Career, interviews and continuous learning

AI PM is a judgement profession built on evidence habits. Careers compound the same way programmes do: measured, honest, and updated when the evidence changes.

Build your evidence portfolio

- Keep (appropriately anonymised) examples of your artefacts: a charter, a gate paper, an eval readout, a benefits pack, an incident review. Interviewers and sponsors believe artefacts over adjectives.
- Be able to tell three stories in STAR form: a project you stopped and why, a quality problem you caught before launch, and a benefit you proved - or honestly disproved.

Interview readiness - questions to expect and to ask

- Expect: "How do you estimate AI work?" "Tell me about a model that failed evaluation." "How do you handle a sponsor pressing for a launch date?" Answer with gates, evidence and decision logs.
- Ask them: "Where does your AI portfolio stall - data, evaluation, governance or adoption?" "Who owns benefits?" "What happened at your last failed gate?" Their answers tell you the real job.

Stay current without chasing hype

- Monthly: one model or platform change that affects your systems, reviewed for eval and cost impact.
- Quarterly: re-read your risk register against new guidance from your regulators and standards bodies.
- Per project: one retrospective lesson written into this playbook. A golden source that never changes is a museum piece.

Final word

The PMs who thrive in AI are not the ones who predict the technology best. They are the ones whose projects produce evidence early, whose sponsors always know the four numbers, and whose systems earn trust because the limits were stated before the failures were found. Be that PM, and keep the receipts.

Appendix A: Glossary

Plain definitions, consistent with Chapter 1. Use these exact meanings in your artefacts so the whole programme argues about evidence, not words.

Term	Meaning in this playbook
Agent	A system that plans steps and calls tools to complete a task, within set permissions
Drift	Degradation as real-world data or behaviour moves away from training / test data
Evaluation (eval)	Structured measurement of system quality, safety and fairness against thresholds
Fine-tuning	Further training a model on examples to change its behaviour
Golden set	Governed test set of real and edge cases used to judge quality and regressions
Grounding	Anchoring outputs in supplied, verifiable sources, ideally with citations
Guardrails	Controls that constrain inputs, outputs and actions to safe, allowed bounds
Hallucination	Plausible output that is incorrect or invented
Human in the loop	A specified human review step with owner, sampling, time budget and override rights
Inference	Using a trained model to produce an output; the ongoing per-use cost
LLMOps / MLOps	The operational discipline of deploying, monitoring and improving AI systems
Prompt injection	Adversarial input designed to make a system ignore instructions or leak data
RAG	Retrieval-augmented generation: supplying relevant source material at query time
Red-teaming	Deliberate adversarial testing to find safety and quality failures before launch
Token	The unit of text GenAI processes, prices and limits by

Appendix B: PM prompt patterns and meeting templates

Prompts are tools for the PM's own work - drafting, summarising and preparing - always with human verification. Never paste confidential or personal data into an unapproved tool.

Five PM prompt patterns (adapt, verify every output)

1. **Meeting to decisions:** "From these notes, extract decisions, owners and dates only. List anything ambiguous as a question, do not resolve it."
2. **Risk first draft:** "Given this project summary, list risks in the form: if [cause], then [event], causing [impact]. Flag which need a named owner."
3. **Steering one-pager:** "Rewrite this update under four headings - Value, Quality, Cost, Risk - one short paragraph each, keeping every number exactly as written."
4. **Failure reading:** "Group these 30 failed cases into categories, quote one example per category, and say what test would have caught each group."
5. **Plain-language check:** "Rewrite this for a non-technical sponsor. Keep all numbers and limits. Flag any claim that needs evidence."

Weekly evidence review - agenda (30 minutes)

1. Eval results since last week, including regressions (10 min)
2. Three raw failures read aloud (5 min)
3. Decisions needed, with deciders named (10 min)
4. Risks whose trigger dates fall this week (5 min)

Steering one-pager - structure

Line 1: recommendation or decision needed. Then the four numbers - Value, Quality, Cost, Risk - each with trend since last month. Then: what changed, what we learned, next gate and its date. One page. If it needs a second page, the meeting is the second page.

Closing golden rule: Write it down, measure it, name the owner, set the date. In AI programmes, that sentence is the difference between a demo and a system people can trust.

End of Playbook • Version 1.0, October 2026 • Review after every major release, incident and gate failure, and at least quarterly.