

AI ROI Framework

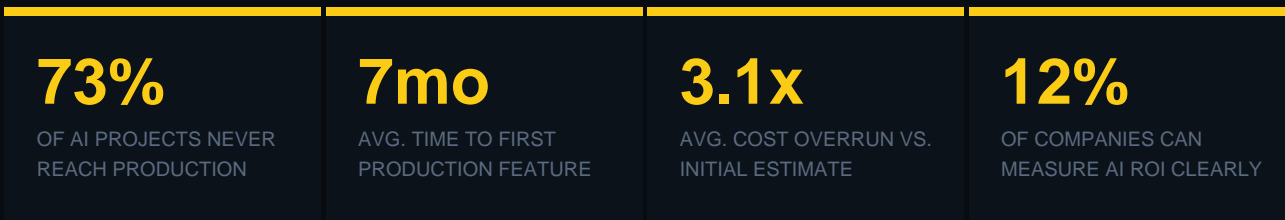
How to measure, justify, and maximise the return on AI investment — before you build a single model.

| | |
|-------------|-----------------------------------|
| Prepared by | Dwayo.ai · Devji Chhanga |
| Version | 1.0 · 2026 |
| Audience | CTOs, Engineering Leads, Founders |
| Format | 5-phase framework + worksheets |

Why most AI projects fail to show ROI

Most companies approach AI investment backwards. They identify a technology, build a system, ship it, and then try to measure whether it worked. By that point, the engineering investment is sunk, the scope has drifted, and "did this work?" is impossible to answer cleanly.

This framework reverses the order. Before any architecture decision, any model selection, any line of code — you define exactly what success looks like, how you will measure it, and what the financial upside is if you hit it. Everything downstream follows from that foundation.



The five phases in this framework address each failure mode systematically. Work through them in order. Each phase has a worksheet section — fill it in for your specific context before moving to the next.

| | | |
|----|---------------------------------|---|
| 01 | DEFINE THE PROBLEM | What exactly are you trying to improve? In numbers. |
| 02 | QUANTIFY THE OPPORTUNITY | How much is this problem costing you right now? |
| 03 | MODEL THE RETURN | What does the upside look like if AI solves it? |
| 04 | ESTIMATE THE INVESTMENT | What will this realistically cost to build and run? |
| 05 | DECIDE & DE-RISK | When does it make sense to proceed, and how? |

Define the Problem

The most common mistake: "We want to use AI for customer support." That is not a problem definition — it is a technology preference. A real problem definition is measurable, bounded, and tied to a business outcome.

The three questions that define a real AI problem:

- Q 1** **What specific process or decision are you trying to improve?** *e.g. "First-pass triage of inbound support tickets" — not "customer support generally"*
- Q 2** **How is it currently performed, and by whom?** *e.g. "4 agents, manually reading and tagging 800 tickets/day across 3 queues"*
- Q 3** **What does a successful outcome look like — in numbers?** *e.g. "Correct triage tag applied >85% of the time, response SLA under 2 hours"*

Worksheet 1 — Fill in before proceeding

Our target process: _____

Currently performed by: _____ Volume: _____ per _____

Success metric #1: _____

Success metric #2: _____

How we will measure it: _____

Quantify the Opportunity

Before modelling upside, calculate the cost of the status quo. This number anchors every ROI conversation — with your board, your team, or a partner.

| Cost type | Formula | Example |
|-------------|---|-----------------------------------|
| Labour cost | Headcount x avg. loaded salary x % time on task | 4 agents x \$60K x 35% = \$84K/yr |

| | | |
|-------------------------|---|--|
| Error cost | Error rate x volume x cost per error | <i>12% misroutes x 800/day x \$18 remediation = \$63K/yr</i> |
| Opportunity cost | Hours lost x value of engineer time | <i>2 hrs/day x 4 people x \$75/hr x 250 days = \$150K/yr</i> |
| Delay cost | Avg. delay per decision x decisions/yr x \$ impact per hr | <i>4hr avg delay x 1,200 cases x \$22/hr = \$105K/yr</i> |

Model the Return

AI ROI is not one number — it is a range across three scenarios. Model all three. The conservative case should be the one you present to your board. The optimistic case is your internal target.

| Scenario | AI Performance Assumption | Expected ROI | Proceed if... |
|---------------------|---|---------------------------------|---------------------------------------|
| Conservative | 50-60% task automation Error reduction: 30% | 1.4x - 1.8x Payback: 14-18mo | Cost savings alone justify investment |
| Realistic | 70-80% task automation Error reduction: 55% | 2.2x - 3.1x Payback: 8-11mo | Primary scenario for planning |
| Optimistic | 85-95% task automation Error reduction: 75%+ | 3.5x - 5.0x Payback: 4-7mo | Use as upside case only |

The ROI formula

$$ROI = (Annual\ value\ delivered - Annual\ cost\ to\ run) / Total\ build\ cost$$

Annual value delivered = Labour savings + Error cost reduction + Revenue uplift

Annual cost to run = Inference cost + Maintenance + Monitoring

Total build cost = Design + Engineering + Integration + Testing

Worksheet 3 — ROI Model

Annual value delivered:

Labour savings: \$ _____

Error cost reduction: \$ _____

Revenue uplift: \$ _____

Total: \$ _____

Annual cost to run:

Inference / API cost: \$ _____

Maintenance est.: \$ _____

Total: \$ _____

Total build cost: \$ _____

ROI (conservative): ____ x Payback period: ____ months

Estimate the Investment

Most AI cost estimates are wrong because they only count build costs and ignore the full lifecycle. Use this breakdown as your starting point — then get a firm quote from your engineering partner.

| | | |
|---|------------|------------------------|
| <p>Discovery & Architecture</p> <ul style="list-style-type: none"> → Stack audit and integration risk assessment → Trust & Safety Spec (privacy, guardrails, compliance) → Technical blueprint and data flow diagrams | 1-2 weeks | \$5K - \$15K |
| <p>Build & Integration</p> <ul style="list-style-type: none"> → Model selection, fine-tuning or RAG pipeline → Integration with existing stack (API, DB, auth) → Eval framework, guardrails, CI/CD | 4-10 weeks | \$30K - \$120K |
| <p>Inference & Running Costs</p> <ul style="list-style-type: none"> → API token usage (model-dependent) → Vector DB / embedding storage → Monitoring and alerting infrastructure | Ongoing | \$200 - \$8K/mo |
| <p>Maintenance & Evolution</p> <ul style="list-style-type: none"> → Model drift monitoring and retraining → Eval suite updates as product evolves → Security patches and dependency updates | Ongoing | \$2K - \$8K/mo |

Decide & De-risk

Use this decision matrix to determine whether to proceed, delay, or restructure the investment.

| Signal | What it means | Recommended action |
|--|--|--|
| Conservative ROI > 1.5x Payback < 18 months | Strong case. Cost savings alone justify investment even if upside is limited. | Proceed. Start with a 2-week discovery sprint. |
| Conservative ROI 1.0x-1.5x Payback 18-30 months | Marginal. Acceptable only if strategic value is high or competitive pressure is acute. | Proceed with reduced scope. Run a 4-week prototype first. |
| Conservative ROI < 1.0x Payback > 30 months | Does not justify investment on current numbers. Problem may be under-scoped or data unready. | Do not proceed yet. Revisit data readiness and problem definition. |

Cannot calculate ROI | Data not available

High risk. You cannot measure what you cannot define.

Stop. Complete Phase 01-02 before any engineering work.

How Dwayo applies this framework

Every Dwayo engagement begins with a completed ROI framework — before any architecture work. This is not a sales exercise. It is how we ensure the project is worth doing, and how we define success in a way that is measurable at handoff.

| | |
|------------------|---|
| Week 1 | Stack Audit + ROI Workshop We work through Phases 01-03 with your team. You leave with a completed ROI model and a go/no-go recommendation. |
| Week 1-2 | Trust & Safety Spec Data flow diagrams, privacy controls, guardrail design, and compliance checklist — written before code. |
| Week 2-4 | Prototype against your real data A working system your engineers can review. Eval baseline established. ROI projections stress-tested. |
| Week 4-10 | Production build with eval gates Every milestone has a quality benchmark. Nothing ships without passing a defined performance threshold. |
| Handoff | Full IP transfer + runbooks You own everything. Source code, infra configs, documentation, and the knowledge to run it independently. |

Ready to run this framework on your problem?

Book a free 30-minute call with Devji. We will work through Phase 01 and 02 with you on the call — no pitch deck, no retainer proposal. You will leave with a concrete problem definition and a rough opportunity size, regardless of whether we work together.

calendly.com/dev-dwayo/30min hello@dwayo.ai