

The PROVE Framework

PRACTICAL ENTERPRISE AI IMPLEMENTATION GUIDE

PROVE guides enterprise AI implementation through five practical stages, from choosing a specific workflow to deciding whether the evidence supports expansion.

P PINPOINT

Choose the workflow

R RUN

Test useful work

O OPERATIONALIZE

Make it dependable

V VERIFY

Establish use and value

E EXPAND

Broaden selectively

GOVERNING PRINCIPLE Expand only what has earned the right to scale.

P Pinpoint

QUESTION What specific workflow, burden, and baseline are we addressing?

IN PRACTICE Select a bounded workflow and define what better performance would mean.

- Map the current workflow: trigger, users, inputs, decisions, handoffs, output, and known pain points.
- Establish a baseline using measures that matter in this workflow, such as cycle time, quality, rework, cost, risk, or user burden.
- Define the target outcome, the intended users, and the work that must remain under human judgment or approval.
- Name the business sponsor, workflow owner, technical owner, data owner, and affected users.
- Set proof thresholds, non-goals, and stop conditions before implementation begins.

EVIDENCE TO RETAIN Use-case brief, workflow map, baseline snapshot, ownership map, authority boundary, and proof criteria.

ADVANCE WHEN The problem is specific and measurable, the evidence is accessible, and a bounded proof is authorized.

R Run

QUESTION Does it work on real data with its failures visible?

IN PRACTICE Test the smallest useful implementation under representative real conditions.

- Build the thinnest end-to-end slice that can produce a useful work product. Avoid building a broad platform around an unproven use case.
- Test with approved, representative data and include normal cases, edge cases, and likely failure conditions.
- Define expected outputs and human review steps before running the evaluation.
- Compare results with the current baseline or manual process. Record quality, correction burden, latency, cost, overrides, and failures.
- Review results with the people who perform and receive the work. Capture where the implementation helps, creates friction, or fails.

EVIDENCE TO RETAIN Reproducible test record, evaluation results, failure log, cost and latency data, user feedback, and a go, revise, hold, or stop recommendation.

ADVANCE WHEN The smallest useful version meets its proof threshold on representative data, and its known limits justify hardening work.

O Operationalize

QUESTION Can we responsibly operate this as a production capability?

IN PRACTICE Turn a useful proof into a dependable capability with clear limits and owners.

- Define what the capability may retrieve, recommend, or act on. Specify approvals, prohibited actions, and sensitive-data rules.
- Add reliability measures such as fallbacks, safe retries, recovery procedures, version control, and rollback paths.
- Make performance visible through logs, health measures, error categories, alerts, and a repeatable evaluation set.
- Apply least-privilege access, retention rules, security review, incident handling, and vendor or procurement requirements.
- Measure operating cost per task and expected volume. Set budget thresholds and capacity limits.
- Assign lifecycle ownership and document the runbook, change process, evaluation cadence, and retirement conditions.

EVIDENCE TO RETAIN Readiness checklist, control and risk register, runbook, monitoring plan, cost model, owner record, and release plan.

ADVANCE WHEN Named owners can operate the capability safely, reliably, observably, and affordably within its approved scope.

V Verify

QUESTION Are people using it, and is it producing business value?

IN PRACTICE Establish sustained adoption and measurable value in the target workflow.

- Place the capability at the point of work in the tools, roles, and handoffs people already use. Remove unnecessary duplicate entry.
- Train users when to use it, when not to use it, how to review its work, and how to correct or escalate problems.
- Start with a bounded cohort and provide an accountable support and feedback route.
- Track eligible users or tasks, active use, repeat use, completion, correction, override, abandonment, and exception patterns.
- Compare operational outcomes with the baseline and investigate nonuse, workarounds, downstream burden, and uneven effects.

EVIDENCE TO RETAIN Updated workflow procedure, enablement materials, adoption measures, outcome comparison, feedback and issue log, and owner review.

ADVANCE WHEN People use the capability consistently in the real workflow, and the measured value remains positive after correction and exception costs.

E Expand

QUESTION Has it earned the right to scale?

IN PRACTICE Decide whether the evidence supports broader investment, reuse, or scale.

- Choose among expand, revise, hold, and retire. Record the decision and the evidence behind it.
- Separate reusable components from workflow-specific assumptions. Do not copy the entire solution by default.
- Assess the next workflow's users, data, risk, integration, governance, and local ownership as a new context.
- Update the business case with observed costs, benefits, support load, and capacity needs. Avoid extrapolating beyond the evidence.
- Expand in bounded increments. Start each new context at Pinpoint and preserve monitoring, version history, and stop conditions.

EVIDENCE TO RETAIN Expansion decision, reuse map, next-use-case brief, updated cost and risk view, capacity plan, and measurement plan.

EXPAND WHEN The expected marginal benefit exceeds the added cost and risk, and the next context is specific enough to begin a new PROVE cycle.