WHITE PAPER · PORTFOLIO & AI GOVERNANCE

The AI Value Test: Automate, Build, Buy, Hire, or Wait

A portfolio decision model for funding AI work only when the operating case is strong enough

Marco Policani

Enterprise Portfolio · PMO · AI Operating Governance

policani.net · linkedin.com/in/marcpolicani

July 2026

Executive premise

The AI value test is not whether AI can do the work. It is whether the portfolio should automate, build, buy, hire, or wait.

That question sounds broader than an AI adoption question because it is. Once AI enters a portfolio, the decision is no longer only technical. It touches funding, vendor strategy, operating design, human accountability, risk controls, data readiness, and the organization's appetite for change. A narrow automation frame can make the first demo look successful while hiding the operating cost of the choice.

The market now offers enough evidence to reject that narrow frame. McKinsey reported that AI use is widespread but enterprise-level impact is still uneven.1 Its 2026 value-realization framework argues that AI impact should be measured with the rigor of any capital investment, with a line from technical performance to adoption, operational KPIs, strategic outcomes, and financial impact.5 Gartner found in 2026 that at least half of GenAI projects had been abandoned after proof of concept by the end of the previous year, citing familiar failure points: lack of business value, poor data, escalating total cost, responsible-AI gaps, and change-management weakness.3 Gartner also predicted in 2025 that more than 40 percent of agentic AI projects would be canceled by the end of 2027 because of cost, unclear value, or inadequate controls.6

The practical lesson is direct. The portfolio should not ask, "Where can we use AI?" It should ask, "Which option creates the best value case for this constraint, given the evidence we have and the operating capacity we can sustain?"

This paper offers a decision model for that question. It does not treat waiting as failure, buying as weakness, hiring as old-fashioned, or building as prestige. Each is a legitimate answer when the constraint, value mechanism, risk, readiness, and cost point in that direction.

Why automation is only one answer

Automation is attractive because it promises a visible reduction in effort. The danger is that effort is often the easiest part of the work to see, not the most important part of the value system.

Some work should be automated because it is stable, repetitive, low ambiguity, and already governed. Some work should be augmented because human judgment remains the value-bearing component. Some work should be redesigned before any AI is added. Some work should be bought because a mature vendor capability already exists. Some work should be built because the capability is strategically differentiating or tightly bound to proprietary context. Some work should be handled by people because accountability, relationship, taste, or expert interpretation is the product. Some work should wait because the data, controls, or business case are not ready.

The wrong decision can raise cost while appearing to reduce labor. A custom agent may require integrations, monitoring, evaluation, human review, security controls, and support capacity that a simple workflow change would not. A vendor tool may speed deployment but create dependency, license cost, and data exposure that the business has not priced. A hiring or training decision may look slower but create durable internal capability. Waiting can protect the portfolio from funding a workflow that has not earned trust.

McKinsey's 2026 article on where AI will create value argues that productivity gains become table stakes and that deeper value comes from reshaping offerings, business models, and market structures.7 That does not make productivity unimportant. It means leaders need to know which economic game they are playing. A work-reduction automation, a strategic capability build, and a new business-model move should not be governed by the same funding logic.

The five options

Automate when the workflow is stable, the decision rules are known, the data is reliable, exceptions are manageable, and the value case depends on speed, throughput, consistency, or lower unit cost. Automation can use AI, deterministic rules, scripts, RPA, workflow tooling, or a combination. The portfolio should fund the simplest reliable automation.

Build when the capability is strategically important, tightly connected to proprietary context, or likely to become a repeatable advantage. Build decisions need stronger scrutiny because they create long-term obligations: architecture, data pipelines, evaluations, model routing, monitoring, support, change management, and future funding. Building is justified when control and learning matter enough to own those obligations.

Buy when the need is common, mature external capability exists, time-to-value matters, and differentiation will come from adoption or process fit rather than custom technology. Buying still needs governance. Vendor claims do not replace proof. Procurement should evaluate data rights, integration burden, model or agent behavior, security, cost scaling, portability, and the practical work required to make the tool useful.

Hire or train when the constraint is judgment, domain knowledge, relationship, adoption, change leadership, or operating capacity. AI can amplify expert workers, but it does not remove the need for people who know what good looks like, can challenge outputs, and can own decisions. Underinvesting in people often turns AI into a faster way to produce work nobody can evaluate.

Wait when the business case is weak, the workflow is unstable, the data is not ready, the risk boundary is unclear, the vendor market is immature, or the organization cannot absorb the change. Waiting should be an explicit decision with review triggers, not a vague delay. A governed wait names what must become true before the idea returns.

The five options at a glance

The five options are funding choices, and they read most clearly side by side. Each fits a different condition, each creates a different obligation after approval, and each fails in a characteristic way when it is chosen for the wrong reason.

Option	When it fits	What it demands after approval	Dominant failure mode
Automate	Workflow is stable, rules are known, data is reliable, and value comes from speed, throughput, or consistency	Monitoring for drift, exceptions, cost, and changing rules	Automation theater: effort falls, no outcome moves
Build	Capability is differentiating or tightly bound to proprietary context	A roadmap, evaluation method, support model, and future funding you now own	Prestige building: a custom asset where a feature would do
Buy	Need is common, mature capability exists, and time-to-value matters	Configuration, integration, adoption, data-rights, and exit management	Procurement substitution: the tool arrives, the constraint stays
Hire or train	Constraint is judgment, relationship, adoption, or capacity	Role clarity, learning paths, and evidence the human system is changing the work	Headcount avoidance: AI covers a gap only people can close
Wait	Case is weak, workflow unstable, or data and controls are not ready	A named trigger and review date, not an open-ended delay	Passive waiting: deferral with no condition to bring it back

Decision criteria that keep the test honest

The five options become useful only when the portfolio evaluates them against common criteria.

Start with strategic value. Is the constraint tied to cost, revenue, customer experience, risk, regulatory exposure, employee capacity, decision quality, or a future differentiator? A low-value constraint should not receive a high-complexity answer.

Then test workflow readiness. Is the current process stable enough to improve? Are handoffs, owners, exception paths, and acceptance criteria clear? If the workflow is messy, AI may amplify ambiguity rather than remove it.

Data and context readiness come next. Does the system have the structured or unstructured information needed for reliable output? Is the data current, governed, accessible, and appropriate to use? RAND's failure analysis and Gartner's GenAI failure points both put problem understanding and data readiness near the center of project risk.[2] [3]

Risk and trust decide how much reliance is acceptable. NIST's AI RMF frames risk management as govern, map, measure, and manage across the lifecycle.[4] That framing matters because reliance is not a yes-or-no switch. A tool can assist a draft, propose a classification, prepare evidence, or execute a step. Each reliance level needs different controls. McKinsey’s 2026 read on AI trust describes a shift into an agentic era in which trust and control, not model access, increasingly gate adoption.[8]

Total cost of ownership should include licensing, token or inference cost, integration, security review, evaluation, monitoring, human review, training, support, vendor management, and future maintenance. Gartner's agentic AI warning is useful here because it links cancellation risk to escalating cost, unclear business value, and inadequate controls.[6]

Finally, test reversibility. If the choice is wrong, how hard is it to stop, unwind, switch vendors, rebuild trust, or recover the workflow? The more irreversible the choice, the stronger the proof gate should be before scale.

Data readiness deserves particular scrutiny because it fails quietly. A workflow can look ready in a demo built on curated inputs, then break on the uneven, permission-restricted, partially-owned data the same process runs on in production. The test before funding is not whether clean data exists somewhere, but whether the specific data this workflow depends on is current, governed, accessible under the right permissions, and appropriate to use for the purpose. When that answer is no, the first fundable move is usually the data work, not the model.

A funding-gate model

The AI value test should run before a project receives serious funding, then repeat at each gate. A good first answer can become a bad scale answer when evidence changes.

At idea intake, the portfolio asks for the constraint, option set, value hypothesis, owner, and evidence plan. It does not need a polished deck. It needs enough clarity to decide whether the idea deserves discovery.

At discovery, the team compares automate, build, buy, hire, and wait. The output is a recommended path and the rejected alternatives. Rejected alternatives matter because they show that the team considered cheaper, safer, or more durable options.

At pilot, the work is tested against technical performance, adoption, and early workflow indicators. McKinsey's five-layer framework is especially useful because it prevents teams from jumping straight from model quality to financial claims.5 The pilot should prove that the system works, that users can put it into real work, and that a meaningful operating signal is emerging.

At scale, the portfolio reviews the full operating case: financial impact, total cost, governance controls, support model, change readiness, and ownership. A project that cannot explain who will maintain evidence, handle exceptions, and review drift is not ready to become business as usual.

At run, the decision remains alive. The organization should continue asking whether the original answer is still right. A bought capability may need to be replaced by a build. A custom build may become a vendor commodity. An automation may expose a need for training. A wait decision may become fundable after the data foundation improves.

How the answers differ in practice

Consider a recurring executive reporting workflow. If the constraint is that leaders receive too much unstructured status language, automation alone may generate prettier noise. The better move may be to redesign the intake fields, define decision categories, and use AI only to prepare exceptions for human review.

Consider a service operation with high first-contact resolution pressure. If the knowledge base is current and the resolution paths are stable, buying or configuring an AI-assisted tool may create value quickly. If the knowledge base is stale, the first investment may be content governance and ownership. If the real constraint is expert escalation capacity, hiring or training may carry more value than another model layer.

Consider a product team exploring agentic workflows. If the use case is strategically differentiating and depends on proprietary context, a build path may make sense. If the use case is a common horizontal task, buying may be faster and safer. If the agent would act without a clear review boundary, waiting is governance.

The same logic applies to portfolio operations. AI can scan inconsistent tracking fields, propose classifications, summarize dependency risk, and prepare scenario inputs. Those uses create value only when connected to named human owners, evidence review, and decisions the cadence actually needs to make.

How to compare the five choices

The value test becomes practical when every option is compared against the same eight questions: value, readiness, data, risk, cost, capacity, reversibility, and strategic control.

Value asks what business condition changes if the choice works. Readiness asks whether the workflow and owner can absorb the change. Data asks whether the information needed for reliable operation is available and governed. Risk asks what happens if the system is wrong, misused, overtrusted, or unavailable.

Cost includes more than the first invoice. It includes licenses, tokens, integrations, evaluations, monitoring, support, security review, vendor management, human review, and future change. Capacity asks whether the organization has the people and attention to implement the choice without starving higher-value work.

Reversibility matters because AI markets move quickly. A choice that is cheap to test but hard to unwind deserves stronger proof before scale. Strategic control asks whether the organization needs to own the capability, the data, the workflow logic, or the learning loop.

These questions keep the five options from becoming slogans. Automate, build, buy, hire, and wait are funding choices. They should be compared like funding choices.

Not every criterion carries equal weight on every decision. On a reversible, low-stakes choice, readiness and cost dominate and the proof gate can be light. On an irreversible or high-exposure choice — a multi-year build, a deep vendor dependency, a workflow that touches regulated decisions — reversibility and strategic control move to the front and the evidence bar rises with them. The eight questions are a checklist for what to examine, not a scorecard to be averaged. A single failing answer on an irreversible choice should outweigh five comfortable answers elsewhere.

What each answer demands after approval

Automation demands maintenance. Someone must monitor whether the rules, model behavior, integrations, exceptions, and cost still match the value case. A workflow that changes every month may turn a promising automation into an expensive upkeep obligation.

A build decision demands product discipline. The organization owns the roadmap, evaluation method, support model, data dependency, security posture, and future funding. Build decisions work when the capability is important enough to justify ownership.

A buy decision demands adoption discipline. Vendor capability does not equal organizational value. The organization still has to configure the tool, integrate it into work, train users, review outputs, manage data exposure, and track whether the operational measure moves.

A hire or train decision demands clarity about judgment. If the constraint is expertise, trust, relationship, or decision ownership, people are not a residual cost after automation. They are the capability. AI may expand their reach, but the portfolio should fund the human system deliberately.

A wait decision demands a trigger. Waiting is governable only when the organization names what must change: data readiness, vendor maturity, risk tolerance, workflow stability, budget capacity, or strategic urgency. Without a trigger, waiting becomes drift.

A worked example

The following is an illustration, not a client account. It runs one constraint through the five options.

A proposal team is missing deadlines because first drafts take too long to assemble. The obvious request is to buy an AI drafting tool. The value test asks a wider question before funding anything.

Automate: drafting is not stable, rule-based work; it depends on judgment about what each opportunity needs. Full automation would produce fast drafts the team rewrites anyway.

Build: nothing in generic proposal drafting is differentiating enough to justify owning a custom system, its evaluations, and its upkeep.

Buy: a mature AI-assisted drafting tool could help, but only if the underlying content — past proposals, approved language, current pricing — is organized and trustworthy. On a disorganized content base, the tool accelerates the wrong starting point.

Hire or train: if the real bottleneck is a shortage of people who can shape a winning argument, tooling will not fix it; capacity and skill will.

Wait: if the content base is stale, the first investment is content governance, and the buy decision waits behind a named trigger — the library reaching a defined coverage and freshness bar.

The honest reading is that the strongest first move is often not the tool the request asked for. It is the content and capacity work that makes a later tool dependable. The value test does not reject AI; it sequences it behind the conditions that let it hold. Recorded as a decision, the outcome is not "buy a drafting assistant" but "govern the proposal library now, protect senior proposal capacity, and reassess the buy at the coverage threshold."

Failure patterns the value test prevents

The first pattern is automation theater. The team automates a visible task because it can, then struggles to show that a customer, process, or financial outcome improved.

The second pattern is prestige building. A custom solution is funded because it feels strategically serious, even though a mature vendor feature or workflow change would have solved the constraint faster.

The third pattern is procurement substitution. A tool is bought to avoid a harder operating decision about ownership, process design, data quality, or human review. The software arrives; the constraint remains.

The fourth pattern is headcount avoidance. AI is asked to cover a judgment, relationship, or change-management gap that actually requires skilled people. The organization saves a role on paper and pays for it later in quality, trust, or adoption.

The fifth pattern is passive waiting. Leaders defer a decision without naming what evidence would bring it back. A governed wait is active. It protects scarce capacity while preserving the option to move when readiness improves.

The evidence pack for the value test

A value-test evidence pack should show how the portfolio chose among automate, build, buy, hire, and wait. It should not only defend the chosen answer. It should make the rejected answers visible enough that leaders can see the tradeoff.

The first page names the constraint, the value mechanism, and the current baseline. The second page compares the five options across value, readiness, data, risk, total cost, capacity, reversibility, and strategic control. The third page defines the proof needed for the next gate. The fourth page records the decision and the trigger for review.

This structure protects against one of the most common AI funding mistakes: confusing a plausible option with the best option. A build can be plausible and still inferior to a vendor feature. A vendor feature can be fast and still inferior to training or data cleanup. Automation can reduce manual effort and still fail if the workflow should have been redesigned first.

The evidence pack also helps finance and operating leaders participate without becoming model experts. They can challenge the value mechanism, the total cost, the capacity assumption, the proof plan, and the reversibility of the choice. That is where many AI decisions become clearer.

Governance after the choice

The value test does not end when the portfolio chooses an option. Every choice creates a different run-state obligation.

Automated work needs monitoring for drift, exceptions, cost, and quality. Built capability needs backlog ownership, support funding, evaluation methods, and lifecycle management. Bought capability needs vendor governance, adoption review, data-rights monitoring, and exit options. Hiring or training needs role clarity, learning paths, and evidence that the human capability is changing the work. Waiting needs named triggers and a review date.

This run-state view is the part of AI governance that often gets missed. Leaders can fund a pilot with enthusiasm and still underfund the operating model that makes the pilot durable. The result is a familiar pattern: impressive proof of concept, fragile production path, unclear ownership, and a quiet return to old work.

A portfolio decision is only complete when the organization knows what it will have to own after approval. That is the real value of comparing automate, build, buy, hire, and wait. It turns the decision from a technology preference into an operating commitment.

What the review should decide

A value-test review should end with one of six decisions: explore, pilot, scale, switch option, wait, or stop. Each decision should be tied to evidence, not to enthusiasm or fatigue.

Explore means the constraint may be valuable, but the team has not earned a pilot. The next work is discovery: clearer baseline, user research, data review, vendor scan, risk boundary, or intervention comparison. Pilot means the team has enough clarity to test a defined option with limited exposure and explicit proof.

Scale means the proof is strong enough to expand reliance and operating commitment. Switch option means the team learned that the original path was wrong but the constraint still matters. Wait means the idea should return when a named condition changes. Stop means the value case no longer deserves attention.

That vocabulary matters because it gives leaders more than yes or no. Many AI ideas should not be killed, but they also should not be funded as if the answer is already known. The value test creates a middle path: keep the option alive while protecting the portfolio from premature commitment.

Evidence gaps this paper cannot close

The cited research does not give a universal formula for when to automate, build, buy, hire, or wait. It provides signals about where AI programs stumble: unclear value, poor problem framing, weak data readiness, cost growth, adoption gaps, and insufficient controls.

The decision model in this paper is therefore a governance standard, not a statistical prediction. It gives portfolio leaders a way to compare options before the organization overcommits. It should be adapted to the risk profile, regulatory environment, architecture, workforce model, and business strategy of the organization using it.

A final limitation is timing. Vendor maturity, model capability, pricing, and enterprise controls change quickly. That volatility strengthens the case for the value test. The portfolio should revisit decisions as the market changes rather than locking a 2026 assumption into a multi-year operating model.

The shift

AI gives organizations more possible moves. It does not make the choice for them.

The mature portfolio will not ask every team to automate by default. It will ask which constraint matters, which option fits, what evidence earns reliance, and whether the total operating case is stronger than the alternatives.

Automate, build, buy, hire, or wait. The value is in making the choice before the tool makes it for you.

Notes and sources

1. Alex Singla, Alexander Sukharevsky, Bryce Hall, Lareina Yee, Michael Chui, and Tara Balakrishnan, "The state of AI in 2025: Agents, innovation, and transformation," McKinsey & Company, November 5, 2025. Verified July 7, 2026. https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai

2. James Ryseff, "The Root Causes of Failure for Artificial Intelligence Projects and How They Can Succeed: Avoiding the Anti-Patterns of AI," RAND Corporation, 2024. Verified July 7, 2026. https://www.rand.org/pubs/research_reports/RRA2680-1.html

3. Arun Chandrasekaran, "Why 50% of GenAI Projects Fail - And How to Beat the Odds," Gartner, January 26, 2026. Verified July 7, 2026. https://www.gartner.com/en/articles/genai-project-failure

4. NIST, "AI Risk Management Framework Core," excerpt from AI RMF 1.0, 2023. Verified July 7, 2026. https://airc.nist.gov/airmf-resources/airmf/5-sec-core/

5. Johannes-Tobias Lorenz, Joshan Cherian Abraham, Robert Levin, and Douglas Ziman, "From promise to impact: How companies can measure and realize the full value of AI," McKinsey & Company, April 24, 2026. Verified July 7, 2026. https://www.mckinsey.com/capabilities/quantumblack/our-insights/from-promise-to-impact-how-companies-can-measure-and-realize-the-full-value-of-ai

6. Gartner, "Gartner Predicts Over 40% of Agentic AI Projects Will Be Canceled by End of 2027," June 25, 2025. Verified July 7, 2026. https://www.gartner.com/en/newsroom/press-releases/2025-06-25-gartner-predicts-over-40-percent-of-agentic-ai-projects-will-be-canceled-by-end-of-2027

7. Antoine Montard, Dago Diedrich, and Tanguy Catlin, "Where AI will create value - and where it will not," McKinsey Quarterly, April 29, 2026. Verified July 7, 2026. https://www.mckinsey.com/capabilities/strategy-and-corporate-finance/our-insights/where-ai-will-create-value-and-where-it-wont

8. McKinsey, "State of AI trust in 2026: Shifting to the agentic era," 2026. Verified July 7, 2026. https://www.mckinsey.com/capabilities/tech-and-ai/our-insights/tech-forward/state-of-ai-trust-in-2026-shifting-to-the-agentic-era

About the author

Marco Policani is an enterprise portfolio, PMO, and AI operating-governance leader. He builds portfolio governance systems where AI carries the analytical load and named humans own every decision — an approach documented across case studies, walkthroughs, and working governance guides on his portfolio site.

Portfolio & governance library: policani.net/governance · Full portfolio: policani.net · LinkedIn: linkedin.com/in/marcpolicani