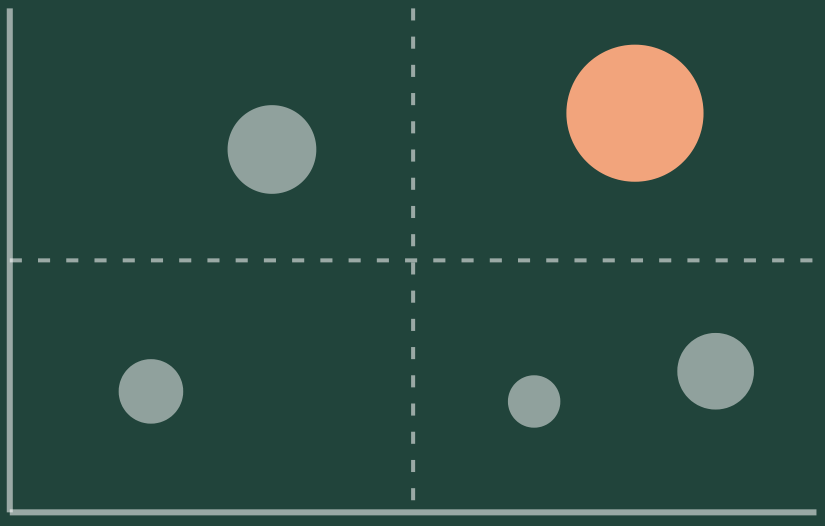
WORKSHEET FOR LEADERSHIP TEAMS



AI Use Case Prioritization Matrix and One-Page Business Case

Dr. Hitesh Sarda · AI advisor, Hyderabad

hitesh.in

A 90-minute working session for a leadership team that has more AI ideas than budget, ending with one use case to pilot, a list to buy, a list to drop, and a one-page business case for the winner.

Why this matrix has more than two axes

The usual matrix plots value against feasibility. That is a fine start, and it still sends most teams to the wrong first project, because "feasible" gets judged from a demo, and a demo only shows that a model can produce the output once with an expert steering. It tells you nothing about what it costs to trust that output ten thousand times.

Producing a candidate answer is now something you rent by the token, while deciding whether the answer is right is the constraint, and that part you have to build yourself. When Anthropic's agentic system found a faster attack on a reduced version of AES, the run took roughly a week and about $100,000 in compute, and then human cryptographers spent close to a month confirming the result was real. So this matrix keeps value and adds the criteria I find decide whether an AI project pays off:

- how cheap the output is to check
- what a wrong answer costs, and whether you can undo it
- whether the task is hard or merely messy
- whether a smaller model, or a product you can buy, would do the job
- who owns the outcome
- what it costs at production volume, with the checking included

Score tasks rather than job titles, because a role is a bundle of tasks that score very differently and the org chart will mislead you. In practice you will find automation candidates inside senior roles and stubbornly human work inside junior ones.

1. Run it as a 90-minute leadership session

> Invite five to eight people: the heads of the functions that proposed candidates, someone from finance, someone from engineering or IT, and one person who does the day-to-day work in each candidate area, and print parts 2 to 6. The 1 to 3 scale, the gates and the timings in this document are a starting point, not a standard, so change them to suit your team.

Before the session

- [ ] Each function head lists the workflows that consume the most hours in their function (aim for 20) and marks the three they would hand to AI first.
- [ ] Every candidate is written as a task with a start and an end ("match supplier invoices to purchase orders"), never as a role ("automate accounts payable").
- [ ] One intake form (part 2) is filled in for each candidate that will be discussed, up to a limit of eight.

- [ ] Someone brings last quarter's volume for each candidate: how many times the task was done, and roughly how long each one took.

Take the task list seriously, because public evidence will not cover most of it. When Mercor mapped 29 public AI benchmarks onto the US Department of Labor's catalog of 18,797 tasks, only 924 tasks were covered at all, and it estimated that about 90% of American working hours go to tasks with little or no benchmark coverage.

Agenda

Minutes	What you do	Output
0 to 10	Read the eight criteria aloud (part 3). Agree that scores are about the task as it runs today.	Shared definitions
10 to 30	Each sponsor presents one intake form in two minutes. Questions of fact only.	Up to eight candidates on the wall
30 to 40	Clear the dominated options: any candidate that is worse than another on every criterion the room cares about. Say so and remove it.	A shorter list
40 to 65	Score what is left (part 4). Everyone writes a score privately, then you compare. Argue only where scores differ by two.	Completed matrix
65 to 80	Read the result (part 5). Sort each candidate into pilot, buy, build the check first, or drop.	Four lists
80 to 90	For the pilot winner: name the owner and write the go or no-go test in one sentence.	First two lines of the business case

The person who does the work gets a veto on two scores. Checking cost and specification are judged best by whoever handles the exceptions every week, so if their score is lower than the sponsor's, use theirs.

2. Use-case intake form

> One copy per candidate, filled in by the sponsor before the session. If a line cannot be answered, leave it blank and say so in the room, since blanks show where scoring will be hardest. The undo line matters. In Project Vend, an AI ran a small office shop for a month and slowly lost money through a hundred small, unverified decisions it could not take back.

Use case name _____

Sponsor, and function _____

The task in one sentence, from trigger to finished output _____

Who does it today, and how many people _____

Volume per month _____

Time per unit today _____

What a finished unit of work is (a matched invoice, a resolved case, a reviewed contract) _____

How you know today that one unit was done correctly _____

Who is qualified to check it, and how long one check takes _____

Worst realistic result of one wrong output going unnoticed _____

Can a wrong output be undone? How, and by when? _____

Number of steps from start to finish, and which step is irreversible _____

Share of cases that are exceptions needing context nobody has written down _____

Where past cases with known right answers are stored, and how many exist _____

How fast the person or customer on the other end needs a response _____

Is there a product on the market that already does this? Name it _____

The one person who would own the result in production _____

Whose budget pays for it, and for the checking _____

3. The scoring criteria

> Score each candidate 1, 2 or 3 on every criterion, where a 3 always means "favorable for a first pilot". Score the task as it runs today.

Criterion	1 (low)	2 (medium)	3 (high)
A. Value Hours and volume the task consumes	Occasional task, a few hours a month	Regular task for a few people	Among the top hour-consumers in the function, repeats daily
B. Cheap to check What one verification costs, and who can pay it	Needs a scarce expert to read and judge each output	A trained person can spot-check a sample in minutes	Automatic check exists: it reconciles, the test passes, the field matches the source

Criterion	1 (low)	2 (medium)	3 (high)
C. Cost of being wrong Blast radius and undo	Moves money, messages customers or touches production, and cannot be taken back	Error is caught downstream at some cost and can be corrected	Output is a draft, a staged change or a recommendation a human approves
D. Hard, not messy How crisply the task is specified	Ambiguous, depends on context held in people's heads, no answer key	Mostly rule-based, with a known set of exceptions	Crisp success criterion and everything needed is on the page
E. Proof on hand Your own past cases with known answers	No record of past cases, or nobody agrees what "correct" was	Cases exist but need cleaning or labeling	Fifty or more closed cases with the right answer recorded
F. Duller tool would do Smaller model, classifier or bought product	Varied, language-heavy work whose categories change every few weeks	A mid-tier model clears the bar for the common case	High-volume, narrow, stable decision, or a proven product you can buy
G. Owner One named person accountable for the outcome	"The AI team" or a committee	A named owner without budget or authority over the workflow	A named owner who runs the workflow and sees the bill
H. Cost at production volume Per finished unit, checking included	Nobody can estimate it, or only the demo cost is known	Model cost estimated, checking time not yet counted	Cost per finished unit estimated with checking and rework, and it beats today's cost

Notes on scoring honestly

- **B, cheap to check.** A better model does not raise this score. A weak model produces obvious garbage that is cheap to reject, while a strong one produces output that is plausible and wrong in ways that look right, which needs full scrutiny before anyone can dismiss it. Score the check you have today.
- **C, cost of being wrong.** Count the steps, because errors compound: at 90% reliability per step, a five-step task finishes correctly about 59% of the time, and ten steps drops it to 35%. A long chain that ends in an irreversible action is a 1, however good the model looks.
- **D, hard, not messy.** Difficulty is the wrong test. OpenAI published what it says is a model-produced proof of the Cycle Double Cover Conjecture, open since the 1970s, while the same class of model still needs a human watching every move on a month-end close. A problem that is hard but self-contained, with a crisp answer, suits a model; a task a new hire learns in a month can stump it when the work is soaked in unwritten context. So score how messy the task is, whatever the seniority of the person who does it.
- **F, duller tool.** "Which LLM?" is the third or fourth question on a roadmap, so ask first what the decision is and how often it repeats. Capability you do not need on this task is still a cost.

- **H, cost at volume.** Price per token is the wrong number. Use the cost of one finished unit of work, and include the minutes of human judgment each unit still consumes.

4. The scoring matrix

> Use one row per candidate and write 1, 2 or 3 in each cell, then the total out of 24. Before you add anything up, circle any 1 in columns B, C or G.

Use case	A Value	B Check	C Wrong	D Spec	E Proof	F Duller	G Owner	H Cost	Total
1.									
2.									
3.									
4.									
5.									
6.									
7.									
8.									

Three scores act as gates. A 1 on B (checking), C (cost of being wrong) or G (owner) takes a candidate out of the running for first pilot, whatever its total, because strong scores elsewhere do nothing for an output nobody can check, a mistake nobody can undo, or a result nobody owns. Money does not move these gates. Meta plans to spend around $145 billion on AI infrastructure in 2026, and Mark Zuckerberg still told his staff that agentic AI development "hasn't really accelerated in the way that we expected."

Candidates removed as dominated before scoring _____

Scores the room disagreed on, and what would settle them _____

5. How to read the result

The total ranks the candidates, but it does not make the decision for you. Read the pattern of scores in each row and sort every candidate into one of four lists.

Pattern	Decision	What to do next
No circled 1s, and B and C are both 3	**Pilot first**	Take the highest total in this group. Fill in the business case in part 6.

Pattern	Decision	What to do next
F is 3 and a proven product exists	**Buy**	Ask the vendor for a trial on your data, scored against your own cases. Do not build.
A is 3 but B or E is 1	**Build the check first**	The first investment is the test set, the reconciliation or the review rule. Rescore once it exists.
A is 3 but C is 1	**Narrow the scope**	Let the model run the reversible stretch and put a human checkpoint before the costly step. Draft, stage, recommend. Rescore the narrower task.
D is 1 and most cases are exceptions	**Drop for now**	Leave it with your people. This is the durable human work. Write the exceptions down and look again in six months.
G is 1	**Park**	No pilot starts until one person with authority over the workflow accepts ownership.

Pilot first where checking is cheap

Your first pilot will probably be boring, something like a workflow where an invoice either matches or it doesn't, and that is fine, because a tight feedback loop can be tested in weeks and reduces uncertainty cheaply. Teams keep pointing AI at the task where it looks most impressive, which is usually the one where checking costs the most.

Buy before you build

Most companies should buy before they build, and that includes buying the boring specialist. Building the first version was never the expensive part; keeping it alive for five years is. An agent that forks an open-source tool and runs a nightly rebase against upstream makes the writing nearly free, but someone still has to know whether last night's rebase broke something. Before you approve a build, answer these:

- [ ] Is there a product that already does this task for companies like yours? If yes, what exactly would your own build do that it cannot?
- [ ] If you build or fork, who reviews the changes every month, and is that person's time in the budget?
- [ ] Is the thing you would own boring, slow-moving and well-scoped? Those are worth owning, while security-sensitive, fast-moving or regulated work is better bought.
- [ ] Whichever you choose, will your test cases, your corrections and your review rules be stored somewhere you control, so you can change vendor or model later?
- [ ] If you start with a frontier model as a fast first version, are you keeping every human correction as a labeled example? Those corrections are what later let a smaller, cheaper specialist take over the high-volume part.
- [ ] Does the result live inside a tool your people already open every day? A standalone assistant has to win attention on its own, and usually loses.

Name the axis you will lose on. If two candidates survive and trade against each other, stop asking which is best and say the trade aloud: "we will be slower so we can be cheaper", or "we will spend more so it feels instant". If nobody in the room can name the axis, the decision has not been made yet.

Pilot first _____

Buy _____

Build the check first, or narrow _____

Drop or park _____

6. One-page business case for the winner

> Filled in by the owner within a week of the session. If the case does not fit on this page, the scope is too wide.

Use case, and owner (one name) _____

Problem

The task, and what is wrong with how it runs today _____

What a finished unit of work is _____

Baseline today

Units per month _____

Cost per finished unit today (people time and tools) _____

Error or rework rate today, and how it was measured _____

Expected change

The one number that should move, from what to what _____

The axis you accept losing on (cost, speed, or quality) _____

How the result will be checked

The check for one output, and who or what performs it _____

Test set: how many real past cases, and who holds the unseen slice _____

Where the human checkpoint sits, and the last step the model may take alone _____

Cost at production volume

Model or license cost per finished unit at full volume _____

Checking cost per finished unit (minutes of whose time) _____

Integration and upkeep cost per year, and who does the upkeep _____

Total cost per finished unit, against today's _____

The budget that pays for the checking _____

Decision

Buy or build, and why _____

The test that decides go or no-go, written before the pilot starts _____

Date the decision will be made, and who makes it _____

What happens if it fails: what is switched off, what you keep _____

Name the budget the saving moves into. When a bill falls, the cost often moves into engineering time, testing effort or an expert's afternoon, where nothing measures it. Unmanaged budgets get cut by decree: Microsoft had projected at least $1 billion of internal spending on Anthropic in 2026, and The Information reported that the estimate has since been cut by more than a third. If nobody pays for the checking yet, write that down as an open item.

7. Pilot design checklist

A convincing pilot often measures the expert running it more than the tool. A 2023 essay, The LLMentalist Effect, compares chatting with a model to a psychic's cold reading, and that bias works better on smart people, who correct the drift without noticing. Tick every item before the pilot starts, so the result tells you about the tool.

Who runs it

- [ ] The pilot is run by a competent non-expert, the kind of person who will use the tool in production. Your strongest person is not on it.
- [ ] After each task, the operator records one line: did the model do this, or did my prompting and correction do it?
- [ ] You are counting the minutes of human judgment each finished unit still consumes.

What it is scored against

- [ ] The test set is built from your own closed cases with known right answers: at least fifty.

- [ ] "A good outcome" is written down in a sentence two reviewers would apply the same way.
- [ ] You have a human baseline on the same cases: accuracy, time and cost.
- [ ] The acceptance test was written before the build and is owned by someone who is not building or tuning.
- [ ] Part of the test set is held back from whoever tunes the prompts, and from the vendor.
- [ ] If a vendor is involved, the trial runs on your data. A demo on the vendor's data does not count as evidence.

What it measures

- [ ] Cost per finished unit of work, with checking and rework included. Price per token is not reported.
- [ ] The share of cases the model hands back to a human or refuses to decide, alongside how often it is right. A model that escalates everything looks safe and removes no work.
- [ ] If people or customers interact with it live: how often they interrupt, abandon or stop responding.
- [ ] Response time against the limit set by the person waiting, written down before the model was chosen.
- [ ] Seats and prompts used are not reported as a success measure, because usage is not an outcome.

How it is contained

- [ ] The steps in the workflow are counted, and a human or hard guardrail sits before the first step that is expensive or permanent.
- [ ] During the pilot the tool drafts, stages and recommends, and does not send, commit or execute.
- [ ] Every human correction is saved as a labeled example somewhere you control.
- [ ] The go or no-go test, the decision date and the decider are the ones written in the business case, unchanged.

8. Common ways the exercise goes wrong

- **You score job titles.** "AI can do 40% of customer support" mixes exposure with evidence, so break the role into tasks and score those.
- **The most impressive candidate wins.** The room falls for the demo of the strategy memo and ignores the reconciliation, so check column B again.
- **The model is chosen first and the task second.** When dermatologist Magnus Lynch showed 200 biopsy-proven skin lesions to two frontier models and a ten-year-old image network, the old network got 62% right and both frontier models got 42%. If the first slide names a model, send it back.
- **Feasibility is taken from a public benchmark.** Benchmarks exist where grading is cheap, and most of your work is not on any of them. A vendor's score tells you about the benchmark's neighborhood, and only your own cases tell you about yours.
- **Savings are budgeted before checking.** A team that automates generation and forgets verification has moved the risk somewhere it cannot see, and the cost went with it.
- **The business case uses the demo's cost.** One run with a cheap prompt says little about ten thousand runs with retries, long context and a reviewer. Cost it per finished unit at full volume.

- **"Human in the loop" is treated as a safety answer.** It is really a pricing question: if the loop is a senior person fixing the model's mistakes, you have bought a power tool for experts, and you should price and staff it as one.
- **A dominated option gets equal airtime.** It has a champion in the room, so the debate feels balanced. If it loses on every axis you care about, say so in the first five minutes.
- **Nobody owns the winner.** Pilots that stall usually had no owner and no agreed test of success. Leave the room with both written down.
- **Everything on the list gets built.** A fork you can generate for free is still a fork you own and have to review, so choose what to own by what you are willing to govern.
- **The session is run once.** Models get cheaper and the trade-offs shift, so a candidate you dropped this quarter may be worth a pilot next year, and one you built may now be worth buying. Rescore the list every six months.

If you finish the session with one owned pilot, a written test for it, and a short list of things you have decided not to do, the 90 minutes were well spent.

Further reading

- Point your AI where the checking is cheap, the argument behind criterion B and the rule of piloting where a check already exists.
- AI does your hardest work, not your messiest, on scoring tasks by specification and verifiability instead of seniority.
- The Autonomy Budget: Deploying AI Agents Before They're Reliable, for the compounding-error arithmetic (the 59% figure) and where to put the human checkpoint.
- Your best people are the worst judges of your AI pilot, the source of the pilot checklist's staffing and scoring rules.
- Nobody has benchmarked AI on 90% of the work your company does, on the task inventory and the fifty-case test built before rollout.
- Your best AI use case may not need a frontier model, on small specialists, buying before building, and what the model does when unsure.
- Cheap to fork, costly to keep: AI redraws build versus buy, on choosing what to own by governance cost.
- The engineer who picks the model never sees the bill, on cost per finished unit of work and giving the bill an owner.

About the author

Dr. Hitesh Sarda is an AI advisor based in Hyderabad and the founder of 256 Technologies, an applied AI lab. He has spent more than 25 years building production software and now helps leadership teams decide what AI to build, buy or skip. He is the author of *Implementing AI Agents in Your Company* and writes at hitesh.in.

> **Want a second pair of eyes on what you filled in?**
>
> Bring this document to an advisory hour. We go through your answers, and you leave with the simplest next step, or an honest view that AI is not the answer yet.
>
> hitesh.in/work-with-me · hitesh@256tec.com

You are welcome to use and adapt this document inside your own organisation. Please do not republish or resell it.