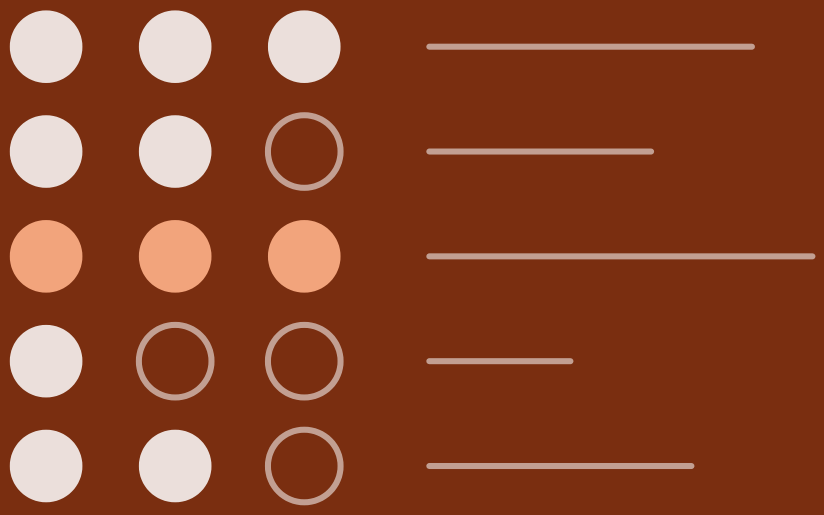
WORKSHEET FOR LEADERSHIP TEAMS



AI Readiness Assessment

25 questions to answer before you fund an AI project

Dr. Hitesh Sarda · AI advisor, Hyderabad

hitesh.in

A scored self-assessment for founders, CTOs and business heads who have one specific AI use case in mind and need to decide whether to fix the basics first, run a small pilot, or scale.

How this assessment works

Most readiness assessments grade your company in general: infrastructure, "data maturity", talent. You can score well on all of that and still fund a project that stalls after the demo. I find it more useful to assess one use case at a time, because readiness belongs to the task. The same company can be ready to let AI match invoices and nowhere near ready to let it reply to customers.

So this worksheet asks 25 questions about a single use case, in five areas: the business case, how cheaply the output can be checked, data and systems, people and ownership, and governance and risk. The second area carries the most weight in my own advice, because producing an answer is now cheap and deciding whether the answer is right is the expensive part. When Anthropic reported that its Claude Mythos Preview system had found a new weakness in HAWK, a post-quantum signature scheme, and a faster attack on a reduced version of AES, the AES run had taken roughly a week and about $100,000 in compute. Human cryptographers then spent close to a month confirming the result was real.

> Pick one use case. Fill in the block below. Then score each question 0, 1 or 2 using the descriptions in the table, add up each area, and carry the five totals to the scoring sheet. It takes about 40 minutes, and it works best with the business owner, an engineer and a finance person in the same room.

Score what is true today. Give a 2 only if you could show the evidence to someone outside the project this week. A plan to do it next quarter scores 0. If two people in the room disagree on a score, write down the lower one.

Start here: the use case on one page

If you cannot fill in these lines, stop and work on them before you score anything. An assessment of a vague idea produces a vague number.

The use case in one sentence (one task, with an input and a finished output) _____

The decision this assessment supports (fund, pilot, buy, drop) and the date it is due _____

The named owner of the outcome (one person, not a team) _____

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

Who pays the bill for running it _____

Who checks the output _____

Assessed by, and on what date _____

Area 1: The use case and its business case

Planning AI by job title ("AI can do 40% of customer support") mixes up exposure with evidence. A role is a bundle of tasks, and they score very differently. This area checks that you have a task, a count, a test of success and an honest cost.

#	Question	Score 0	Score 1	Score 2	Score
1.1	Is the use case one task, or a job title?	A function or a role ("AI for support").	A workflow that still bundles several tasks.	One task with a defined input and a finished output.	
1.2	Do you know how often this task repeats and how many hours it takes today?	Nobody has counted.	An estimate from the people who do it.	Counted from your own records for a recent period.	
1.3	Did you agree, before any pilot, what result would justify production?	No test of success.	A general aim ("save time", "improve quality").	A written result, measured against today's baseline on real work.	
1.4	Have you checked whether a tool you already run, or a workflow fix, solves this without AI?	Nobody asked.	Discussed, nothing written.	Checked, and you can say why the existing stack cannot do it.	
1.5	Does the business case price a finished unit of work, including the human checking?	Only seats or price per token.	Cost per unit, checking time left out.	Cost per finished unit with checking time, compared with today's cost.	

Area 1 total (out of 10) _____

On 1.4: a surprising amount of "we need AI here" turns out to be a workflow problem an existing tool already handles. On 1.5: price per token misleads, because a cheaper model that takes more steps can cost more per task. When Databricks graded coding agents on more than a hundred of its own merged pull requests, Sonnet 5 was cheaper per token than Opus and yet cost more per task, which is why I would count the finished unit.

Area 2: How cheaply the output can be checked

Every deployment needs something that tells you whether a given output is right. If that is a test, a reconciliation or a match against a source document, checking is close to free and you can deploy hard. If it is a scarce expert reading every output, no model quality removes that cost. Ask what one check costs and who can pay it.

#	Question	Score 0	Score 1	Score 2	Score
2.1	What tells you a single output is right?	An expert reads it and judges.	A partial automatic check, then a person.	An automatic check: a test, a reconciliation, a match to source.	

#	Question	Score 0	Score 1	Score 2	Score
2.2	Do you know what one check costs?	No. Checking is left to "the team".	You know who checks, not how long it takes.	Timed: minutes per check, and whose minutes.	
2.3	What evidence do you have that AI can do this task?	A vendor demo or a public benchmark.	A handful of examples you tried yourself.	About fifty real closed cases with known answers, plus a human baseline.	
2.4	Who operated the tool when it was tested?	The vendor, or your strongest expert.	Your own team, with an expert steering.	A competent non-expert, and you noted per task who solved it.	
2.5	Do you know what the model does when it is unsure?	Nobody has looked.	You know accuracy only.	You know the share of cases it hands back to a person, and can live with it.	

Area 2 total (out of 10) _____

On 2.4: your best people are the worst judges of a pilot. They correct the model in one line and route around the dead ends without registering it, so the result measures them. On 2.5: a model that escalates nearly everything scores well on anything that sounds like safety and removes none of the work you bought it to remove. When the dermatologist Magnus Lynch showed 200 biopsy-proven skin lesions to GPT-6 and Claude, they kept 97 and 96 of the 100 dangerous lesions but discharged only 32 and 22 of the 100 harmless ones, so people were told to see a doctor for nearly every lesion.

> **A better model raises the checking bill.** A weak model produces obvious garbage, which is cheap to reject. A strong one produces output that is plausible and wrong in ways that look right, and that needs full scrutiny. Do not plan for checking time to shrink as models improve.

Area 3: Data and systems

This area is deliberately narrow. I am asking whether this task has what it needs, and whether you are adding more new machinery than the project can carry. Dan McKinley's 2015 essay Choose Boring Technology gives a company roughly three innovation tokens to spend, and an AI component holds one of them permanently, because it fails in ways you cannot fully list in advance. Everything around it should be as dull and well understood as you can make it.

#	Question	Score 0	Score 1	Score 2	Score
3.1	Can you name the system that holds the source of truth for this task, and the person who can grant read access?	Neither.	The system is known, access is not arranged.	Both named, access granted.	
3.2	Is the knowledge the task needs written down?	Mostly in people's heads: exceptions nobody documented.	Partly documented, with known gaps.	A capable new hire could do the task from the written material.	
3.3	What happens to a human correction of the AI's output?	It is discarded.	It stays inside the vendor's product.	Stored with the reason, in your own storage, in a plain format.	

#	Question	Score 0	Score 1	Score 2	Score
3.4	How many new components does this need besides the model?	Several at once: new database, framework, orchestration.	One new piece.	None. The rest runs on systems you already operate.	
3.5	Could you change the model or the vendor?	Your rules and workflow logic live in one vendor's product.	Possible, with a rewrite.	The model sits behind a wrapper, and your own test set tells you within a week whether a replacement is good.	

Area 3 total (out of 10) _____

On 3.2: the work that stumps AI inside a company is often work a new hire picks up in a month. It is ambiguous, soaked in context, and comes with no answer key. If the context is not on the page, the model does not have it. On 3.3: every correction is a labeled example of what "good" means in your business. Keep them and you may later be able to hand the high-volume, stable part of the job to a smaller, cheaper model. Throw them away and you have paid for a prototype you can never graduate from.

Area 4: People and ownership

Three seats have to be filled by named people: the one who owns the outcome, the one who sees the bill, and the one who reviews the output. Pilots that stall usually had no owner and no agreed test of success. The last two questions are about the reviewer, because "we keep a human in the loop" only helps if that human still brings doubt to the job.

#	Question	Score 0	Score 1	Score 2	Score
4.1	Who owns the outcome?	Nobody, or "the AI team".	A committee, or a sponsor who cannot stop it.	One named person, accountable for the result, who can stop it.	
4.2	Does the person who chooses the model see what it costs?	The spend sits in a central budget.	Finance sees a total after the fact.	The team sees its own spend weekly, with a default model for this workflow.	
4.3	Who reviews the output, and is their time in the plan?	"The team" will check.	A named reviewer, with no hours set aside.	A named reviewer who can judge the work, with hours budgeted.	
4.4	Can the reviewer catch the model being wrong?	They depend on the tool to understand the domain.	They know the domain. Nobody has tested it.	They show what they verified, and have caught real model errors in the trial.	
4.5	What happens when someone on this workflow says "I don't know"?	A confident sign-off is expected.	Doubts can be raised. Nothing is done with them.	There is a written route for it: check the source, ask the specialist, escalate.	

Area 4 total (out of 10) _____

On 4.2: the engineer who picks the model owns the failure if the cheap one gives a worse answer, and somebody in another department owns the invoice if the expensive one costs ten times more. Choosing the most capable model every time is the rational move in that setup, and the spend grows until finance reaches for a cap. Microsoft had projected at least $1 billion of internal spending on Anthropic for 2026, and according to The Information's reporting that estimate has since been cut by more than a third, while internal Claude Code users at Meta reportedly fell from about 60,000 to about 30,000 (neither company confirmed the numbers). Asking people to be efficient will not prevent that. What helps is showing each team its own spend every week, where the model gets chosen.

On 4.3: if the reviewer is a senior person fixing the model's mistakes, you have bought a power tool for experts. That is a fine thing to buy, as long as you price and staff it as one and take the labor saving out of the business case.

Area 5: Governance and risk

How much the AI may do on its own depends on the task. Two properties decide it: how cheaply you can check the work, and how easily you can undo it. Drafting, summarizing, classifying, or staging a change a person approves can run with a lot of freedom. Moving money, messaging customers and touching production cannot. Spending more on compute does not change those two properties. Meta still plans to spend around $145 billion on AI infrastructure in 2026, and Mark Zuckerberg has told staff that agentic AI development "hasn't really accelerated in the way that we expected".

#	Question	Score 0	Score 1	Score 2	Score
5.1	What may it touch?	It runs with the full access of whoever uses it.	Limited by policy. No system enforces the limit.	Its own identity, read-only or scoped to what this task needs.	
5.2	How many steps does it take before a person looks?	Nobody has counted.	Counted. No checkpoint placed.	Counted, with a checkpoint before the first costly or permanent step.	
5.3	Can its actions be undone?	It sends, pays or changes live systems directly.	Some actions can be reversed, some cannot.	It drafts, stages or recommends. A person executes.	
5.4	Could you say afterwards what it did, in what order, with which credentials?	No record.	Only the vendor's logs.	A record you hold and can read.	
5.5	What happens when it is wrong, or switched off?	No plan.	"We would fix it."	Written: who is told, how a person reverses it, and how the work gets done without the AI.	

Area 5 total (out of 10) _____

Count the steps. Small error rates multiply. At 90% reliability per step, a five-step task finishes correctly about 59% of the time, and a ten-step task about 35%. The demo was one run, while production is the same chain thousands of times, so put the human checkpoint where the arithmetic says it belongs.

Your scoring sheet

> Copy the five area totals here and add them up. Mark the band for each area using the key under the table, then read the next section.

Area	Maximum	Your score	Band (fix / pilot / scale)
1. Use case and business case	10		
2. How cheaply the output can be checked	10		
3. Data and systems	10		
4. People and ownership	10		
5. Governance and risk	10		
Total	**50**		

Band	Each area	Total	What it means
Fix first	0 to 4	0 to 24	Do not fund a build. Fund the missing basics.
Small pilot	5 to 7	25 to 37	Fund a narrow, time-boxed test designed to answer the open questions.
Scale	8 to 10	38 to 50	Move to production in stages, with review rates that fall as evidence comes in.

The lowest area sets the pace. A total of 38 with a 3 in checking or in governance is a fix-first project. Strength in one area does not pay for a gap in another, so read the five area scores before you read the total. The cut-offs are a working guide for a conversation, so treat a score one point either side of a line as on the line.

Lowest-scoring area _____

Overall band after applying the lowest-area rule _____

How to read your score

Fix first

The use case may be a good one. What is missing is the ground it has to stand on, and a pilot run now would tell you little. Spend the next few weeks on the cheapest missing pieces, in this order:

1. Name one owner for the outcome and write the use case as a single task.
2. Write down what result would justify production, and measure today's baseline.
3. Find or build the check. If the task has no cheap way to confirm an output, your first AI investment is that check. Skip it and you get a demo that a person babysits forever.
4. Collect about fifty real, closed cases where you already know the right answer.

Then score again. If the use case still sits in this band, look at your task list for one with a tighter feedback loop. An invoice either matches or it does not, and work like that can be tested in weeks.

Small pilot

You know enough to learn something from a test. Keep it small and design it so it cannot flatter you:

- [ ] One task, not the whole workflow.
- [ ] Graded on your own closed cases, against the human baseline.
- [ ] Operated by a competent non-expert, since that is who you will deploy to.
- [ ] After each task, a note on who solved it: the model, or the person's prompting and correction.
- [ ] Every output reviewed at the start, and every correction stored with its reason.
- [ ] Cost recorded per finished unit, including the reviewer's time.
- [ ] The AI drafts, stages or recommends. A person sends, commits or executes.
- [ ] A date on which the owner decides: stop, extend, or move to production.

Use the pilot to raise your weakest area. If Area 2 scored 5, the pilot's main output is a timed, trusted check. If Area 4 scored 5, it is a reviewer who has caught the model being wrong.

Scale

You have a task, a check, an owner and controls. Scaling here means widening the scope a step at a time.

- [ ] Lower the review rate task by task as the evidence earns it, and put a date on the next review of those thresholds.
- [ ] Give the sampling rate an owner. It is a risk appetite expressed as a number, and it belongs to the person accountable for the outcome.
- [ ] Track cost per finished unit alongside usage, and show each team its own spend weekly.
- [ ] Re-run your test set when a new model arrives, on a fixed rhythm (quarterly suits most teams) instead of reacting to each launch.
- [ ] Split the work by difficulty once you have data: a cheaper model for the routine majority, the expensive one for the hard minority.
- [ ] Watch whether confidence in reviews is rising while outcomes stay flat. That is a warning sign.

What a low score in each area is telling you

Low area	What it usually means	First move
1. Use case	You are planning by job title or by tool.	List the 20 workflows that consume the most hours in the function. Pick one task from that list.
2. Checking	The use case demos well and will be slow to trust.	Time one check. Build the test set. If no cheap check can exist, choose a different first use case.
3. Data and systems	The context is in people's heads, or the stack has too many new parts.	Write down the exceptions. Start storing corrections. Keep everything except the AI boring.
4. People	Nobody owns the result, the bill or the review.	Fill the three seats with names before spending anything else.
5. Governance	The AI has more reach than the task needs.	Narrow it: read-only access, a checkpoint before the costly step, a record you hold.

Red flags that override the score

Any one of these puts the project in "fix first", whatever the total says. I have listed them separately because each can hide behind a good score in the other four areas.

- [ ] **No single named owner.** The outcome belongs to a committee, a vendor or "the AI team".
- [ ] **No agreed test of success.** Nobody wrote down, before the pilot, what result would justify production.
- [ ] **The only evidence is a demo or a leaderboard.** The vendor picked the task, the operator and the framing, and public benchmarks cover the work that is cheap to grade, which may not be yours.
- [ ] **It only works in your best person's hands.** You have measured your employee, and their time is the cost you wanted to remove.
- [ ] **Nobody in the building can sign off the output.** Expert-grade work produced cheaply is worth little if no one can confirm it is right.
- [ ] **An irreversible action with no checkpoint.** The AI can move money, message a customer or change production before a person or a hard guardrail looks.
- [ ] **The business case counts the savings and leaves out the checking.** Budget the people and tooling to verify before you budget the savings.
- [ ] **You could not say afterwards what it did.** No record of its actions, their order, or the credentials it used.
- [ ] **The model was chosen before the task.** "Which LLM?" is the third or fourth question on a roadmap. If it was the first, go back to Area 1.

Red flags ticked _____

Decision (fix first / small pilot / scale / drop) _____

The one thing to fix before the next review _____

Who will fix it, and by when _____

Date to re-score this use case _____

One last use for this sheet. Score a second and third use case the same way and put the three side by side. The one your team is most excited about is often the one that demos best, and the one with the highest score in Area 2 is usually the better place to start.

Further reading

- Point your AI where the checking is cheap, the argument behind Area 2 and the reason a better model raises the checking bill.
- Your best people are the worst judges of your AI pilot, on staffing the pilot with a non-expert and tracking who solved each task.
- Nobody has benchmarked AI on 90% of the work your company does, for the task inventory of 20 workflows and the test of fifty real cases with a human baseline.
- Your backlog is the only benchmark that matters, on building a test from your own closed work and measuring cost per task.
- The engineer who picks the model never sees the bill, on cost per finished unit, weekly showback and keeping a way out.
- The Autonomy Budget, for the compounding arithmetic (59% at five steps, 35% at ten) and where to place the checkpoint.
- Boring technology is an AI strategy, on keeping everything around the AI dull and re-evaluating models on a schedule.
- Your best AI use case may not need a frontier model, on what a model does when unsure and why you keep the corrections.

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.