

A REFERENCE & SELF-ASSESSMENT INSTRUMENT



The AI-Native Engineering Maturity Model

Ten dimensions for scaling AI — and the decisions that still have to be human.

Ed Schaefer

EXECUTIVE COACH & STRATEGIC ADVISOR

edward-schaefer.com · ed@edward-schaefer.com

v2.0 / 2026

A map for scaling AI — and a way to see where you stand

Most organizations adopt AI tool by tool. This model looks at the system around the tools: how AI is governed, measured, secured, architected, and learned. It gives a leadership team one scale to argue from: where are we today, and where do we actually need to be?

The shape. Ten dimensions, four stages each. Every dimension follows the same arc — AI use going from invisible, to visible, to codified, to embedded — and the further along a dimension gets, the more of the mechanical work AI takes over. The real question each one asks is where, as that happens, a person still has to own the decision.

STAGE	WHAT IT MEANS	WHERE AI USE LIVES
1 · Shadow	Individuals or teams use AI without org awareness, policy, or measurement.	<i>Invisible off the books</i>
2 · Sanctioned	The org acknowledges AI use and permits it with basic boundaries and an acceptable-use policy.	<i>Visible loosely controlled</i>
3 · Standardized	Documented standards and repeatable practices, codified across teams.	<i>Codified same shape org-wide</i>
4 · Integrated	Built into the operating model; continuously measured, governed, and improved.	<i>Embedded how work gets done</i>

How to use it. Read each dimension and place your org on the four-stage scale; each lists concrete signals to help you tell which stage you're in. Place yourself honestly, then ask a few other people where they'd put the org — the most useful conversation is usually in the disagreements. Then work the dimensions where the gap between where you are and where the work actually needs you to be is largest — not simply the lowest-scoring, since how far to take each one is a deliberate choice.

Going deeper. Each dimension is followed by a detailed breakdown — its components scored across the same four stages — with a scoresheet at the back to record where you land. Working with a team, score the components on your own first (budget three to five minutes each; two sittings is normal), then bring the disagreements into the room.

Why a maturity model. The shape isn't new — it borrows CMMI/DORA's ladder on purpose, because that shape surfaces where people disagree about where you are, which is the conversation worth having. What's new is using it to separate what AI now does from the calls that still have to be human.

At a glance

Each dimension measures one part of the system around the tools; the last columns name the core shift and who should answer it. How far you take each is a deliberate choice — for some, reaching Integrated is the goal; for others, a solid Standardized is enough.

#	DIMENSION	WHAT IT MEASURES	THE CORE SHIFT	WHO ANSWERS
1	Governance & Risk	Policy, boundaries, and review for AI use.	<i>From what we ban to what we enable.</i>	Exec
2	Agent Autonomy	How much an agent does on its own, and who decides.	<i>From blanket rules to earned trust.</i>	Delivery
3	Operating Model	Roles, decision rights, and review gates around AI.	<i>Give AI the same role clarity you give people.</i>	Exec
4	SDLC Integration	How deliberately AI works across the lifecycle.	<i>AI works at every stage; you place the gates.</i>	Delivery
5	Leadership	Whether leaders make non-naïve AI calls.	<i>From watching the work to judging it well.</i>	Exec
6	Tooling	How tools are selected, governed, and cost-controlled.	<i>From invisible spend to cost you can stop.</i>	Platform
7	Metrics & Measurement	How impact is measured — activity vs. outcome.	<i>Knowing what changed, and by how much.</i>	Evidence
8	Security & Data Handling	How sensitive data is protected through AI tools.	<i>Seeing the data before it leaves.</i>	Platform
9	Architecture	How AI integration survives provider and model change.	<i>Build for the swap you know is coming.</i>	Platform
10	Team Capability & Culture	How fluency, learning, and safety spread on a team.	<i>From one fluent person to a team that learns.</i>	Delivery

No dimension comes first

The ten dimensions have no order — you don't have to finish Governance before you touch Tooling. The sequence is the four stages, and every dimension climbs the same four. What the dimensions do have is owners. Use this page to plan the scoring session: who has to be in the room for which pages.

Direction & rules <i>the exec team answers</i>	Leadership · Governance & Risk · Operating Model
The work <i>delivery teams answer</i>	SDLC Integration · Agent Autonomy · Team Capability & Culture
The platform <i>platform / security answer</i>	Tooling · Architecture · Security & Data Handling
The evidence <i>feeds all three</i>	Metrics & Measurement

The only two real prerequisites

- Evidence. Almost every dimension's Stage 4 asks for proof, and proof means logs — what the AI tools actually did, where, and to what. If you build one piece of plumbing early, build the logging. Without it, nothing here reaches Integrated.
- Honesty. Every score in this document comes from people describing how the work actually happens. If engineers can't say 'we skip that gate' without it costing them, you'll score high and be wrong. Team Capability & Culture is the dimension that protects that honesty — score it early.

Where to start. Most orgs come in through Governance (a policy scare) or Tooling (a spend scare). Bottom-up adopters usually need Team Capability and SDLC Integration first. Nobody should start with Agent Autonomy — deciding how far to trust agents takes observability most orgs don't have yet.

Using AI vs. shipping AI. Some dimensions measure AI in how you build (SDLC Integration, Team Capability, most of Tooling). Others measure AI in what you ship (Architecture, parts of Security). If your teams use AI tools but ship no AI features, score those rows N/A — Agent Autonomy too, if you run no agents. N/A is a legitimate answer here.

Uneven is normal. Almost no org sits at the same stage across all ten. Stage 3 in Tooling next to Stage 1 in Metrics isn't a bad result — it's the finding, and it tells you where the next work is.

Stages aren't equal effort. Standardized in Operating Model is a written agreement; Standardized in Architecture is shipped, tested infrastructure. Compare each dimension against where it was last quarter, not against the other nine.

Two rules keep every score honest

Sanctioned means someone chose it, in writing. A capability you'd have anyway — HTTPS, a vendor's default settings — doesn't earn Stage 2. Stage 2 is a deliberate, documented decision: we allow this, and here is the boundary. If you can't point to where it's written down, score Stage 1.

Integrated means a person still confirms. Stage 4 is never 'the system handles it.' Automation that flags or contains is fine — an alert, a spend cap, an automatic rollback. Automation that decides is not: nothing should raise its own trust level, ship itself, or move budget without a human saying yes. If no one confirms, don't score Stage 4.

A note on scoring: maturity here doesn't average. An organization at stage 3 in nine dimensions and stage 1 in security isn't "stage 2.8" — the weakest dimension is usually the one that decides what happens next. Score all ten honestly, then pay attention to your lowest numbers first.

One more thing before you score: don't count the same practice twice. A handful of topics show up in several dimensions on purpose — each dimension asks a different question about the same practice. When a page feels like something you already scored, check what it's actually asking before you carry the score over:

Logging & audit	Turns up almost everywhere, asking something different each time — Security asks who touched what data, SDLC asks which stages AI touched, Governance asks what the org's total exposure is. Same logs, different questions.
Review gates	Governance asks whether a gate is required. Operating Model asks who staffs it. SDLC asks where it sits in the workflow. Agent Autonomy asks about the gate inside an agent's own loop.
Cost	Tooling owns the bill and getting it down. Architecture enforces the spend ceilings. Metrics owns cost per outcome — whether the spend buys anything.
Data rules	Governance writes the rule and the escalation path. Security enforces it: classification tiers, and which providers may see which data.
Provider strategy	Tooling picks the vendor. Architecture makes the vendor swappable. Security decides what any vendor may do with your data.

Governance & Risk

How an organization sets policy, boundaries, and review for AI use – and treats the resulting risk.

STAGE 1 Shadow	AI is used across the org but invisible to leadership. No acceptable-use policy, prompt-hygiene standards, or data rules. Individuals decide what goes into prompts — some of it sensitive. The risk is real, and no one owns it.
STAGE 2 Sanctioned	AI is acknowledged and permitted with basic boundaries. A one-page acceptable-use policy says what data can and can't go in, which tools are approved, and where to escalate. Compliance has been consulted; people aren't guessing.
STAGE 3 Standardized	Governance is codified across teams: documented prompt-hygiene standards, reusable role-spec prompts in version control, and risk-tiered review gates. New use cases go through a defined risk-assessment step before they ship.
STAGE 4 Integrated	Governance is built into how the org operates. Risk monitoring is continuous, and each accepted risk has a named owner. Compliance, engineering, and security stay in one conversation. Governance is used to tell teams what's safe to ship faster — though keeping it current takes real, ongoing effort.

PLACE YOUR ORG

circle where you are

- ☐ 1 • Shadow no policy or rules; AI use is invisible to leadership.
- ☐ 2 • Sanctioned someone can point to a written acceptable-use policy.
- ☐ 3 • Standardized prompt standards and review gates are documented and applied.
- ☐ 4 • Integrated every high-risk use case has a named risk owner.

ASK YOURSELF

Where does your governance actually live today?

AI can draft the policy, surface the risky prompts, and model the odds. Owning the call — and answering for it later — is the part that can't be handed off.

THE MOVE

Apply the discipline to the decisions.

Deliberate calls on the specific ones, plus a standing job to keep the standards current.

Governance & Risk

Circle the stage each component is at. · Sanctioned = documented intent, not table-stakes. · Integrated = human-confirmed; automation that flags or contains is fine, that decides is not.

COMPONENT	SHADOW	SANCTIONED	STANDARDIZED	INTEGRATED	
Acceptable-Use Policy <i>Written rules defining permitted AI use, approved tools, and where to escalate questions.</i>	No written policy; each person decides what is allowed; leadership unaware AI is in use.	One-page acceptable-use policy names approved tools and escalation paths; people know the boundaries.	One policy applies org-wide, with a published tool catalog and an onboarding path when a new use case comes up.	Policy enforced through tooling defaults and revised on a set cadence as tools and exposure change.	1 2 3 4
Data Handling Rules <i>Controls over what data, especially sensitive data, may enter prompts, context, and training paths.</i>	Individuals paste whatever they want, including sensitive data; no classification or boundaries exist.	A written rule states what data can and cannot enter prompts, and where to escalate a question about it.	Data rules tied to classification tiers and enforced through approved tooling and prompt-hygiene standards.	Data boundaries monitored continuously; leakage signals surface and feed back into rules and tooling.	1 2 3 4
Governed AI Artifacts <i>Treatment of prompts, role-specs, and agent configs as managed assets rather than private scratch text.</i>	Prompts live in chat history and personal notes; nothing is shared, versioned, or reviewed.	Useful prompts shared informally within teams; a few are documented but not controlled.	Reusable role-spec prompts in version control with change review and prompt-hygiene standards.	Artifacts owned, versioned, and updated as managed assets, with provenance tracked across their lifecycle.	1 2 3 4
Human Review Gates <i>Where and how human judgment is required before AI output is acted on, scaled to consequence.</i>	Review is ad hoc and personal; whether anyone checks AI output depends on the individual.	Stated expectation that higher-stakes output gets a human check before it ships.	Risk-tiered review gates defined per use case, with criteria for what needs human sign-off.	Gates get lighter or heavier based on what past reviews actually caught, not on a standing assumption about which use case is risky.	1 2 3 4
Risk Assessment Process <i>The repeatable method for evaluating a new AI use case before it goes live.</i>	No assessment; use cases start and spread without anyone weighing the risk.	New uses are run past compliance informally; concerns raised case by case.	Defined risk-assessment process every new use case passes through before approval.	Assessment runs continuously and proportionately inside delivery, sized to the exposure of each use case.	1 2 3 4
Risk Ownership & Acceptance <i>Whether residual risk is explicitly accepted and assigned to a named accountable owner.</i>	Risk is real but unowned; no one is accountable for what AI use might cause.	Compliance is consulted case by case, and higher-profile use cases have a named risk-taker recorded in an email or a ticket.	Risk decisions documented per use case, with sign-off recorded against the assessment.	Each use case has a named risk owner; residual acceptance is recorded and reviewed over time.	1 2 3 4
Incident & Failure Response <i>How the org detects, escalates, and learns from AI-caused errors, harms, or near misses.</i>	AI errors handled privately or hidden; no path to report or escalate a bad outcome.	People know to flag serious AI failures, but handling is improvised once one occurs.	Defined intake and escalation for AI incidents, with severity criteria and a documented response.	Incidents drive routine post-reviews that update policy, data rules, gates, and artifacts.	1 2 3 4
Usage Visibility & Monitoring <i>Leadership's ability to see where, how much, and at what exposure AI is actually used.</i>	Usage is invisible to leadership; no inventory of where AI touches the work.	Approved tools give leadership a rough picture of sanctioned usage.	Usage is tracked against the tool catalog, so leadership can see it team by team.	Continuous monitoring of usage and exposure signals informs governance decisions in near real time.	1 2 3 4

Agent Autonomy

How much an AI agent is allowed to do on its own — and how that latitude is granted and changed.

STAGE 1 Shadow	No model for what agents can do. Individuals make ad-hoc calls about what agents read, write, or trigger. Some agents are unbounded; others are needlessly boxed in.
STAGE 2 Sanctioned	Agents are held to advise-and-recommend by default. Anything that writes code, deploys, or triggers an action needs a person to approve it. The line is blunt, but it's real.
STAGE 3 Standardized	A tiered trust model defines what each tier can read, write, run, deploy, and — within written, delegated limits — approve. New agent classes go through a known promotion path, and the review gate matches the tier.
STAGE 4 Integrated	Autonomy is adjusted by track record: an agent earns more latitude by performing, and loses it when it doesn't. Where a human stays in the loop is decided per use case by what's at stake, not set once for everything.

PLACE YOUR ORG

circle where you are

- ☐ 1 · Shadow no shared model for what agents may do on their own.
- ☐ 2 · Sanctioned agents can't write, deploy, or trigger without explicit approval.
- ☐ 3 · Standardized you can name the tier a given agent sits in and what it may do there.
- ☐ 4 · Integrated an agent's latitude has actually changed — up or down — based on its record.

ASK YOURSELF

How far does your AI go on its own today?

AI can run inside its tier all day. Deciding which tier it has earned — and when to pull that back — stays a human call.

THE MOVE

Build the trust ladder.

Define the tiers, set the promotion and demotion criteria, and place the human checkpoint by what's at stake.

Agent Autonomy

Circle the stage each component is at. · Sanctioned = documented intent, not table-stakes. · Integrated = human-confirmed; automation that flags or contains is fine, that decides is not.

COMPONENT	SHADOW	SANCTIONED	STANDARDIZED	INTEGRATED	
Action Scope Definition <i>Which classes of action an agent may take: read, write, run, deploy, and — within delegated limits — approve.</i>	No agreed action classes; one agent can deploy while another cannot even write, by accident.	Default is read and advise; write, run, and deploy each require a person to act.	Read/write/run/deploy are an explicit permission taxonomy; approve is granted only for pre-delegated, low-stakes, reversible classes a person wrote down in advance.	An agent's permitted actions widen or narrow with its demonstrated reliability for each action class — someone still signs off before the change takes effect.	1 2 3 4
Trust Tier Structure <i>The named levels of latitude an agent can hold and the criteria that separate them.</i>	No levels exist; latitude is whatever each agent's author wired up at build time.	Two effective states only: advisory by default, or elevated after explicit human sign-off.	A multi-tier ladder with named levels and stated criteria for the latitude each grants.	Tier definitions are revised as evidence accumulates; the ladder itself evolves with observed behavior.	1 2 3 4
Tier Assignment & Change <i>The named process and owner for placing an agent on the ladder and moving its tier up or down.</i>	Latitude shifts when someone edits config; no record of who widened it or why.	Raising latitude needs case-by-case approval; no defined path, criteria, or owner.	A known process places new agent classes on the ladder against published criteria.	Agents rise on track record and drop on regression through an evidence-driven, human-confirmed process, without a per-change request cycle.	1 2 3 4
Human Checkpoint Placement <i>Where a human checkpoint sits in an agent's workflow and the rationale for its position.</i>	Human involvement is incidental; some risky steps run unattended, some trivial ones are gated.	A person reviews before any consequential action by blanket default, regardless of risk.	Review gates are positioned to match tier, so checkpoints track the latitude granted.	Each remaining checkpoint is a deliberate per-use-case choice, justified by risk and reversibility.	1 2 3 4
Behavioral Evidence Base <i>The recorded history of agent actions and outcomes that informs latitude decisions.</i>	No record of what agents did or how often they erred; latitude rests on impression.	Outcomes are recalled anecdotally; approvers rely on recent memory, not recorded history.	Agent actions and outcomes are logged against tier and reviewed on a set cadence.	A continuous performance signal feeds calibration and informs tier changes a person still approves.	1 2 3 4
Blast-Radius Containment <i>The limits on how much damage an agent's actions can do if it behaves wrongly.</i>	No bounds on impact; an errant agent can affect production or spend with nothing to stop it.	Consequential actions are caught by the human approval step rather than by any technical limit.	Each tier carries defined guardrails: scoped credentials, environment limits, rate and spend caps.	Containment shrinks or grows with the same evidence that moves an agent's tier; rollback and kill paths are exercised regularly.	1 2 3 4

Operating Model

How roles, decision rights, and human review gates are arranged around AI work.

STAGE 1 Shadow	No defined operating model. Who decides, reviews, and owns is worked out case by case. Some decisions get over-engineered; others fall through the cracks.
STAGE 2 Sanctioned	Basic roles are named — who approves a tool, who reviews high-risk uses. Some decision rights are clear; human-in-the-loop placement is informal but acknowledged.
STAGE 3 Standardized	Decision rights are documented across use cases. Operator, reviewer, and approver roles are clear. Where a human review gate belongs is treated as a design decision, since different work needs different gates.
STAGE 4 Integrated	The operating model treats AI as a participant you design for: it gets a role spec, explicit decision rights, and a working agreement for what it's allowed to do. Those evolve as the work changes.

PLACE YOUR ORG

circle where you are

- ☐ **1 · Shadow** who decides, reviews, and owns AI work is worked out case by case.
- ☐ **2 · Sanctioned** someone can say who approves a new AI tool and who reviews high-risk output.
- ☐ **3 · Standardized** operator, reviewer, and approver roles are written down and staffed on purpose.
- ☐ **4 · Integrated** AI has a role spec and decision rights the way a team member would.

ASK YOURSELF

Who's accountable when the AI gets it wrong?

AI can do the work. Who runs it, who checks it, and who owns the outcome is a design decision a person has to make.

THE MOVE

Give AI a real role.

A defined role, clear decision rights, and a gate sized to the stakes — the same clarity you'd give a person.

Operating Model

Circle the stage each component is at. · Sanctioned = documented intent, not table-stakes. · Integrated = human-confirmed; automation that flags or contains is fine, that decides is not.

COMPONENT	SHADOW	SANCTIONED	STANDARDIZED	INTEGRATED	
Decision Rights <i>Who is authorized to decide what about AI use, tooling, and outputs.</i>	Authority is unassigned; each AI decision is settled in the moment, some over-escalated, some made by no one.	A few high-stakes decisions have a named owner; most authority is still resolved case by case.	Decision rights are documented across common use cases; people know who decides, who is consulted, who is informed.	Decision rights are sized to risk and revised when the work itself changes.	1 2 3 4
Role Definition <i>The distinct roles people hold around AI work (operator, reviewer, approver, owner).</i>	No named roles; one person improvises operator and reviewer at once, with no separation.	Basic roles named for select cases such as approving a tool or reviewing risky output; coverage is partial.	Operator, reviewer, approver, and owner roles are defined and separated across the work that uses AI.	Roles are specified and staffed on purpose, including AI as a designed participant with a defined remit.	1 2 3 4
Review Gate Placement <i>Where human review sits in the flow of AI work, and how that placement is chosen.</i>	Human review happens at random or not at all; whether a check exists depends on who is involved.	Review for high-risk uses is acknowledged and informally expected, but placement is inconsistent.	Gate placement is a deliberate design choice; checks vary by the risk and reversibility of the work.	Gates are added where stakes rise and removed where they only add delay.	1 2 3 4
Working Agreements & Specs <i>The written norms, role specs, and team agreements that govern how AI shows up in the work.</i>	No shared norms; each person sets their own unstated rules for how and when to use AI.	A few conventions circulate by word of mouth; nothing is written down or applied consistently.	Role specs and working agreements are documented and referenced; teams share a common picture of expected behavior.	Specs and working agreements are living documents, updated as the team reviews practice and learns from it.	1 2 3 4
Ownership & Accountability <i>Who is accountable for the outcome when AI contributes to a decision or deliverable.</i>	When AI output causes a problem, accountability is unclear; blame lands on whoever happened to be nearby.	For a few high-risk uses, a person is named as accountable for the result, even if AI produced it.	Accountability for AI-assisted outcomes is assigned to defined owners across the main categories of work.	Owners are accountable for AI-assisted results end to end, and accountability is reassigned when work moves between teams.	1 2 3 4

SDLC Integration

How deliberately AI participates across the software lifecycle, stage by stage.

STAGE 1 Shadow	AI is used ad hoc at individual stages — one team for code generation, another for tests, nobody coordinating. SDLC documentation doesn't mention AI; leadership can't see it.
STAGE 2 Sanctioned	AI is explicitly part of some stages, usually coding and sometimes review. Planning, testing, and operations are catching up. The picture is uneven.
STAGE 3 Standardized	AI participates across the relevant stages with a defined human review gate per stage. Documented practices say what AI does and where humans stay in. New integrations follow a known pattern.
STAGE 4 Integrated	AI is designed into the lifecycle, with review gates that move as capability and risk change. Throughput goes up — and review becomes the new constraint, so reviewer capacity and the gates get managed on purpose rather than assumed.

PLACE YOUR ORG

circle where you are

- ☐ 1 · Shadow AI shows up at scattered stages, with no coordination or visibility.
- ☐ 2 · Sanctioned AI's role in coding or review is written into how you work, on purpose.
- ☐ 3 · Standardized each stage AI touches has a defined human review gate.
- ☐ 4 · Integrated you manage reviewer capacity as a real constraint, not an afterthought.

ASK YOURSELF

Where does AI actually touch your delivery?

AI can take a pass at every stage. Where you put the review gates — and how far you trust each one — is yours to decide.

THE MOVE

Gate by what a miss costs.

Map where AI touches delivery, put a gate where a mistake would hurt, and watch review load as throughput climbs.

SDLC Integration

Circle the stage each component is at. · Sanctioned = documented intent, not table-stakes. · Integrated = human-confirmed; automation that flags or contains is fine, that decides is not.

COMPONENT	SHADOW	SANCTIONED	STANDARDIZED	INTEGRATED	
Lifecycle Stage Coverage <i>Which SDLC stages (plan, design, code, review, test, deploy, operate) AI actually participates in.</i>	AI touches one or two stages by individual choice, mostly coding; other stages untouched.	AI named in a few stages, usually coding and review; planning, test, and ops lag.	AI participates across each stage the team decided it belongs in, with that decision written down — which stages are in scope, which are deliberately excluded.	AI is a designed participant end-to-end; adding it to a new stage is a deliberate decision.	1 2 3 4
Cross-Stage Context Handoff <i>How AI-relevant context and artifacts carry across handoffs between lifecycle stages.</i>	Each stage's AI use is isolated; context is dropped at every handoff between stages.	A few adjacent stages share context informally; most handoffs still reset to zero.	Defined handoffs preserve AI context and artifacts across stages following a known pattern.	Context and intent carry across the lifecycle by design; handoffs retain prior decisions and rationale.	1 2 3 4
Per-Stage Human Review Gates <i>Where in each stage a human must inspect, approve, or override AI output before work proceeds.</i>	Whether AI output is reviewed depends on the individual; no defined gate, coverage unknown.	Gates exist on one or two stages, usually code review; other stages run ungated.	A defined review gate per stage, with written criteria for what the human must check.	Gate placement is risk-calibrated; over- and under-review both get corrected.	1 2 3 4
Practice Documentation <i>Whether AI's role and the human's responsibility at each stage are written down and findable.</i>	Nothing is documented; SDLC docs omit AI entirely; the practice lives in individual heads.	A few stages have informal notes; coverage is uneven and the notes go stale.	Documented practice states what AI does and where the human stays in, per stage.	Documentation is kept current with practice and feeds onboarding and audit directly.	1 2 3 4
New-Stage Integration Intake <i>The path a team follows to add AI to a stage it does not yet use.</i>	Each new use is improvised from scratch; outcomes depend on who attempts it.	Teams copy one group's approach informally, with no defined template or entry criteria.	New integrations follow a known sequence: scope the role, set the gate, document, then roll out.	An intake step weighs fit, risk, and gating before any new-stage adoption, and feeds results back.	1 2 3 4
Footprint Instrumentation <i>Whether AI's participation across lifecycle stages — which stages it touches, not how each call performs — is observable from telemetry, not just claimed.</i>	AI's footprint is invisible above the team; no signal shows where it actually runs.	Sanctioned stages emit some usage signal; the rest of the lifecycle stays dark.	Instrumentation reports where AI participates and where gates fire, consistently across stages.	Lifecycle AI participation is observable in delivery telemetry and reviewed in normal delivery oversight.	1 2 3 4

Leadership

Whether leaders make non-naive calls about AI — informed by enough firsthand context to judge it well.

STAGE 1 Shadow	Leaders know AI exists, but most haven't taken a formal position. A few use it; many don't. Strategy doesn't mention AI; where it shows up, it's off the books.
STAGE 2 Sanctioned	Leadership has named AI as permitted. Strategic intent is stated at a high level — 'we'll use AI where it makes sense' — true and still vague. Leaders are on record but speak about the work from outside it.
STAGE 3 Standardized	Leaders have enough firsthand exposure to make non-naive calls — where to trust AI, where to gate it, what to fund. AI shows up in strategy with measurable goals, and leadership adjusts as adoption data comes in. As the tools speed up, the leadership job grows: intent, review gates, and the safety for people to be honest about what's working.
STAGE 4 Integrated	AI decisions are consistently well-judged, because the people making them have hands-on context and use it to coach others. That experience shapes how the org sets strategy, runs operations, and develops people.

PLACE YOUR ORG

circle where you are

- ☐ 1 · Shadow most leaders haven't taken a real position on AI.
- ☐ 2 · Sanctioned leaders have taken an actual position on AI, in writing, in the strategy.
- ☐ 3 · Standardized a leader can describe a real AI trade-off they made, not just one they approved.
- ☐ 4 · Integrated a leader can name an AI call they reversed on the evidence — and their team can name it too.

ASK YOURSELF

Am I leading AI from inside the work, or from the sidelines?

AI can draft the memo and surface the numbers. What's worth doing, who to trust with what, and where judgment has to stay human — that's the job.

THE MOVE

Judge the calls, not the keystrokes.

Get enough hands-on context to make non-naive calls, then hold leaders to the quality of those calls.

Leadership

Circle the stage each component is at. · Sanctioned = documented intent, not table-stakes. · Integrated = human-confirmed; automation that flags or contains is fine, that decides is not.

COMPONENT	SHADOW	SANCTIONED	STANDARDIZED	INTEGRATED	
Personal Practice <i>Whether leaders use AI hands-on in their own work versus directing it from outside.</i>	A few leaders experiment privately; most have never opened the tools themselves.	Leaders back AI use but still speak about the work from outside it.	Leaders use AI in their own work, visibly enough that teams can see it.	Hands-on use is routine; leaders work in the same tools alongside their teams.	1 2 3 4
Public Stance <i>Whether leaders take and stand behind a formal, on-record position on AI.</i>	Most leaders hold no stated position; AI sits outside official messaging.	Leadership names AI as permitted and goes on record endorsing its use.	Leaders state where AI applies and where it does not, and act consistently with it.	Leaders revise their public position on AI — where it applies, where it doesn't — and say so when they got it wrong.	1 2 3 4
Strategic Direction <i>How explicitly AI shapes stated strategy and direction-setting.</i>	Strategy makes no mention of AI; any use falls outside the official plan.	Strategy names AI at a high level: we will use it where it makes sense.	AI appears in strategy with measurable goals tied to business outcomes.	AI-native assumptions shape strategy and are tested against delivery results, then revised when the evidence says the strategy was wrong.	1 2 3 4
Investment Posture <i>How deliberately leaders fund tools, capability, and capacity for AI.</i>	No dedicated funding; spend runs through individual expenses or stays hidden.	Reactive funding approved case by case after teams ask for it.	Budgeted lines for tools and enablement, sized against adoption targets.	Funding is rebalanced as evidence on what the last cycle's spend returned comes in, not only at budget season.	1 2 3 4
Adoption Visibility <i>Whether leaders can measure real AI use and impact across the org.</i>	Leaders have no view of who uses AI or to what effect.	Leaders know use is permitted but hold no measure of actual uptake.	Leaders track adoption and impact signals and review them on a cadence.	Adoption and impact measures are instrumented and watched continuously.	1 2 3 4
Expanded Leadership Role <i>How leaders absorb the work that grows as AI speeds delivery: intent, review, safe risk-taking.</i>	No recognition that faster tools shift what leadership must own.	Leaders sense more oversight is needed but the new work stays undefined.	Leaders set explicit intent, review gates, and conditions for safe risk-taking.	Leaders rework their own role as delivery speed changes, and review whether the intent, gates, and risk conditions they set are producing better calls.	1 2 3 4
Developing Others <i>Whether leaders build AI capability in their people through structured development, not just permission.</i>	Capability-building left to individuals; leaders neither model nor coach it.	Leaders encourage people to learn AI but provide no structured development.	Leaders run repeatable enablement so teams build AI skill on purpose.	Leaders can point to specific people whose AI capability they built, and it shows in those people's work.	1 2 3 4

Tooling

How AI tools are selected, governed, and cost-controlled.

STAGE 1 Shadow	Everyone picks their own tools. The same model is paid for three ways across three teams; spend has no owner and surfaces after the fact. A workload can recompute the same paid result every run and burn budget unnoticed.
STAGE 2 Sanctioned	An approved tool list exists. Procurement tracks basic cost; SSO covers the common tools. Exceptions still happen but are now visible. The org knows total spend but not per-workload cost.
STAGE 3 Standardized	A tool catalog with approval status, per-team cost allocation, and vendor governance. Cost optimization is practiced — cheap-model routing, prompt caching. Guardrails live in code: separate keys per workload, build checks that fail on paid calls made outside the cached, shared client.
STAGE 4 Integrated	Tool and model choices are reviewed against the work as it changes. The org can tie a cost to the workload that caused it and say what value came back. The multi-vendor setup is deliberate, and the cost instrumentation runs where the work runs.

PLACE YOUR ORG

circle where you are

- ☐ 1 · Shadow everyone picks their own tools and the spend has no owner.
- ☐ 2 · Sanctioned there's an approved tool list and someone owns the spend.
- ☐ 3 · Standardized you can allocate AI cost to a team, and guardrails live in code, not a wiki.
- ☐ 4 · Integrated you can attribute a cost to the specific workload that caused it.

ASK YOURSELF

Would anything actually stop a runaway AI workload before the bill arrives?

AI can route the work, cache what's stable, and log every call's cost. Which spend is worth it, and where the hard ceiling goes, is the budget owner's to set.

THE MOVE

Put the guardrails in the code.

Persist paid results, isolate the blast radius, and fail the build on a silent-cost bug.

Tooling

Circle the stage each component is at. · Sanctioned = documented intent, not table-stakes. · Integrated = human-confirmed; automation that flags or contains is fine, that decides is not.

COMPONENT	SHADOW	SANCTIONED	STANDARDIZED	INTEGRATED	
Tool & Model Selection <i>How the org decides which AI tools and models are permitted and chosen for a given task.</i>	Each engineer picks their own tools; no shared list; the same model is procured separately by three teams.	An approved tool list exists; off-list exceptions still happen but are now visible to leadership.	A catalog records approval status, owner, and intended use; new tools pass a defined intake before use.	Tool fit is re-evaluated when a new model ships or a task shifts; selection is matched to each job.	1 2 3 4
Cost Visibility & Attribution <i>How clearly AI spend is measured, owned, and traced to the work that caused it.</i>	Spend has no owner and surfaces only after the fact; per-team and per-workload cost are unknown.	Procurement tracks total spend, but the org cannot say what any single workload consumes.	Cost is allocated per team; budgets have named owners; large workloads are tracked individually.	Each call's cost is attributed to its workload and reviewed against the attribution model when usage patterns change.	1 2 3 4
Cost Optimization Practice <i>Deliberate techniques teams use to reduce spend while holding the quality the task needs.</i>	No optimization technique in use; a workload may recompute the same paid result on every run.	Teams know spend is high and make ad hoc cuts, but apply no standard optimization technique.	Routing easy tasks to cheaper models and caching repeated prompts are documented, expected practice.	Routing and caching are tuned from live cost-and-quality signals; optimization defaults are revisited when a provider's prices change.	1 2 3 4
Automated Tool Controls <i>Technical controls in the codebase and pipeline that enforce safe, efficient tool use without relying on policy memos.</i>	Shared keys, no checks; nothing in code prevents costly or unsafe call patterns.	SSO gates who can access approved tools, but runtime spend and call patterns are not enforced in code.	Separate keys per workload; build checks fail on paid calls that bypass the cached shared client; spend limits enforced automatically.	Controls and call-level instrumentation run inside the work, flagging cost and misuse as code ships.	1 2 3 4
Vendor & Multi-Provider Strategy <i>How the org manages provider relationships, lock-in risk, and the mix of vendors it depends on.</i>	No vendor governance; whichever provider an engineer signed up for becomes a hidden dependency.	Procurement knows the main vendors and holds contracts, but provider choice is largely incidental.	Vendor governance covers terms, data handling, and renewals; the provider list is reviewed on a schedule.	A deliberate multi-provider posture; workloads can shift across vendors as price, capability, and risk change.	1 2 3 4

Metrics & Measurement

How AI's impact is measured — activity versus outcome.

STAGE 1 Shadow	No AI-specific metrics, so investment runs on perception. Some teams claim big gains, others none, and neither can be checked. With no baseline, no one can tell later whether anything moved.
STAGE 2 Sanctioned	The first numbers count activity — active users, calls, tokens — and spend shows up at the invoice. Real progress, and a trap: volume starts to look like value. The productivity debate is still settled by anecdote.
STAGE 3 Standardized	Metrics shift to quality, flow, and cost per outcome. Delivery measures are recomputed for non-deterministic work — where teams have wired eval and review gates into merge, a clean merge means those AI-specific checks passed, not just CI. Cost becomes per-user and per-team with ceilings.
STAGE 4 Integrated	Measurement runs as a loop. Outcome metrics tie to business value, and AI performance is tracked against a baseline, so the question is always 'better than what, and by how much.' A jump in usage gets questioned before it gets celebrated.

PLACE YOUR ORG

circle where you are

- ☐ 1 · Shadow there are no AI-specific metrics; it runs on perception.
- ☐ 2 · Sanctioned you track any AI number at all, even just usage.
- ☐ 3 · Standardized you measure cost per outcome, not just activity, against a ceiling.
- ☐ 4 · Integrated you compare AI performance to a baseline and act on the gap.

ASK YOURSELF

How do I know whether our AI investment is actually working?

AI can build the dashboards. What a number means, and what to change because of it, is an argument someone still has to win in the room.

THE MOVE

Measure outcome against a baseline.

Set the baseline, track outcome against it, and let the result drive a real operating decision.

Evidence anchor: DORA's annual research is the measurement baseline this dimension assumes you know.

Metrics & Measurement

Circle the stage each component is at. · Sanctioned = documented intent, not table-stakes. · Integrated = human-confirmed; automation that flags or contains is fine, that decides is not.

COMPONENT	SHADOW	SANCTIONED	STANDARDIZED	INTEGRATED	
Baseline & Counterfactual <i>Whether AI performance is compared against a defined reference point ('better than what, by how much').</i>	No reference point; gains asserted with nothing showing what they improved on.	Pre-AI numbers cited occasionally, but no consistent reference for comparing AI output.	A documented baseline per workflow; AI results compared against it on a fixed cadence.	Baselines re-derived when the work itself changes shape; each claimed gain carries a magnitude and comparison point.	1 2 3 4
Activity vs. Outcome Separation <i>Whether measurement distinguishes usage volume from delivered value.</i>	Neither tracked; impact judged by impression and individual conviction.	Activity counted (active users, calls, tokens); volume read as value by default.	Metrics split into activity and outcome; quality and flow measured alongside usage.	Activity and outcome reported as separate ledgers; a usage spike prompts a why-question.	1 2 3 4
Delivery Metric Adaptation <i>Whether delivery and quality measures are reworked for non-deterministic, AI-assisted work.</i>	Standard delivery metrics applied unchanged; AI-specific quality effects stay invisible.	Teams see old metrics misfit AI work but keep using them as-is.	Delivery measures are recomputed for AI work; where eval or review-gate signals exist, they are tracked alongside CI rather than assumed.	Adapted metrics built into the pipeline; eval and review signals feed delivery measures automatically.	1 2 3 4
Cost & Unit Economics <i>Whether AI spend is tracked, attributed, and expressed per unit of outcome.</i>	Spend untracked or buried; no view of what AI costs the organization.	Total spend visible at the invoice; not attributed to teams, users, or work.	Cost computed per user and per team with ceilings; cost-per-outcome reported.	Cost anomalies surface and route to owners; unit cost weighed against the value produced.	1 2 3 4
Leading-Indicator Use <i>Whether early-signal metrics are defined and acted on before trailing results land.</i>	Nothing forward-looking; problems surface only after they land in production or budget.	Trailing counts exist; the gap to early signals is acknowledged but none are defined.	Leading indicators defined and monitored to anticipate quality, adoption, and cost shifts.	Leading indicators trigger intervention before lagging outcomes move; the two reconciled each cycle.	1 2 3 4
Metric Governance & Improvement Loop <i>Whether someone owns the metric set and uses it to change decisions over time.</i>	No owner; whatever numbers exist are gathered ad hoc and rarely revisited.	A few dashboards exist, owned informally; reviewed when someone asks, not on a schedule.	Defined owner and review cadence; metric definitions documented and changes tracked.	Metrics feed a standing loop that reshapes investment, and the metric set itself is revised when it misleads.	1 2 3 4

Security & Data Handling

How sensitive data is protected as it flows through AI tools.

STAGE 1 Shadow	No AI-specific data policies. Sensitive material — customer information, financials, your own IP — may be flowing into prompts, and no one would know. Individuals decide what's safe to paste. The exposure is real and invisible.
STAGE 2 Sanctioned	Basic data classification reaches AI tools. Sensitive categories are named with a do-not-put-into-AI line; SSO and paid accounts cover common cases. Keys and scopes are separated per workload so one exposure stays contained.
STAGE 3 Standardized	Classification is documented and mapped to use cases — which providers may touch what, what's encrypted. Compliance requirements are written down. An audit trail records what AI saw and produced; secure patterns are codified into templates.
STAGE 4 Integrated	AI data security is woven into the overall posture and watched continuously. Controls live in code — a build check can catch a path that sends sensitive data to a provider without recording it, so most unsafe versions get stopped before they ship. A green build then means nothing sensitive moved without a record.

PLACE YOUR ORG

circle where you are

- ☐ 1 · Shadow no data rules; sensitive data may be going into prompts unseen.
- ☐ 2 · Sanctioned there's a named do-not-put-into-AI data line people actually know.
- ☐ 3 · Standardized data classes are mapped to providers, and there's an audit trail of what AI saw.
- ☐ 4 · Integrated a control in code catches sensitive data heading to a provider unrecorded.

ASK YOURSELF

Could you say what your AI tools have already seen?

AI can classify the data, enforce the rules in code, and log every exposure. What an exposure is worth, and what the org can live with, is a judgment a person owns.

THE MOVE

Design for visibility first.

Map data to providers, record what AI touches, and put the check in code so the risky path is caught early.

Evidence anchor: OWASP's LLM Top 10 is the floor for the risks this dimension scores.

Security & Data Handling

Circle the stage each component is at. · Sanctioned = documented intent, not table-stakes. · Integrated = human-confirmed; automation that flags or contains is fine, that decides is not.

Scope. This covers data handling. The adversarial surface — prompt injection, tool-call abuse, supply chain — is a distinct concern a future version adds as its own component.

COMPONENT	SHADOW	SANCTIONED	STANDARDIZED	INTEGRATED	
Data Classification & Sensitivity Mapping <i>Whether data sensitivity is defined and matched to which AI providers or models may process it.</i>	No AI-specific classes; each person judges case by case what is safe to paste.	Data is sorted into sensitivity tiers, each with a stated rule for which AI tools may touch it.	Classes documented and mapped to use cases; which provider may touch what is written down.	Classification travels as metadata; tool access and model routing derive from it automatically.	1 2 3 4
Identity, Access & Tenancy Controls <i>How human identity, authorization, and provider tenancy scope who can reach AI systems and data.</i>	Personal logins and free accounts; access undefined; no separation between users or workloads.	SSO and paid accounts cover common cases; access controls applied to the obvious risks.	Role-based access and tenancy boundaries aligned to data classes, enforced low in the stack.	Access and tenancy verified continuously, least-privilege by default, drift detected and corrected.	1 2 3 4
Secrets & Credential Hygiene <i>How API keys, tokens, and machine credentials for AI services are issued, scoped, and rotated.</i>	Keys shared, embedded in prompts or scripts, reused across people and workloads.	Keys and scopes separated per workload; obvious shared-key cases retired.	Keys vaulted, narrowly scoped, rotated on a schedule, kept out of code and prompts.	Short-lived scoped credentials issued automatically; a detected exposure triggers rotation and alerts.	1 2 3 4
Provider Boundaries, Retention & Training Use <i>How data is contained at the provider edge, including encryption, retention, and use of data for training.</i>	Flows to providers unknown; retention windows and training-on-data terms unexamined.	Provider retention windows and training-use terms are read and recorded per vendor; encryption and opt-out settings are verified rather than assumed.	Provider terms and retention contractually aligned to data classes; flows documented per use case, with the gap between contractual guarantee and technical verification acknowledged.	Provider routing enforced in the pipeline; an unrecorded sensitive-data path to a provider blocks release.	1 2 3 4
Compliance & Regulatory Alignment <i>How regulatory and contractual obligations are mapped onto AI data handling.</i>	AI use sits outside compliance scope; obligations for AI-touched data unconsidered.	Relevant regimes named; AI use flagged where regulated data is plausibly involved.	Compliance requirements written down and mapped to AI use cases and the controls that satisfy them.	Compliance evidence generated from live controls; obligations re-checked when the posture or a regulation changes.	1 2 3 4
Audit Trail, Monitoring & Exposure Response <i>Whether AI inputs, outputs, and data access are recorded, watched, and acted on when something leaks.</i>	No record of what was sent to or returned by AI tools; exposure happens with no signal.	Some logging on approved tools; account access roughly visible; leaks found by chance.	Audit trail of inputs, outputs, and data accessed, retained and reviewable; a defined exposure procedure.	AI data flows monitored continuously; anomalies surface and route into response without manual triage.	1 2 3 4
Secure-by-Default Building Blocks <i>How reusable, pre-vetted patterns and components for handling data in AI workflows reach engineers.</i>	No reusable patterns; everyone reinvents handling; safe approaches depend on individual knowledge.	A few approved tools and guardrails exist; reviews gate the obvious cases before launch.	Secure patterns codified into templates and libraries engineers reuse by default.	The safe path is the default path; building blocks carry the controls, and reviews shift to exceptions.	1 2 3 4

Architecture

How AI is integrated technically – and how well it survives provider and model change.

STAGE 1 Shadow	AI is bolted onto whatever exists. A model call is wired in where it fits that afternoon; the provider is hardcoded, the cost-and-latency choice made once and forgotten. Nothing is reusable, and a provider's bad hour breaks the integration.
STAGE 2 Sanctioned	Patterns start to repeat. A couple of teams settle on the same chat or retrieval shape and one writes it down. Architecture reviews ask about AI but stay shallow. The model is still 'whatever SDK we use,' so swapping providers means rewriting consumers.
STAGE 3 Standardized	Integration patterns are documented and reused. Provider selection moves into config with a written rationale per route — small-model-first with a fallback. Consumers call a published contract; a tiered router walks a fallback chain that gets logged.
STAGE 4 Integrated	The architecture assumes model-agnostic operation. A vendor change or a cheaper model is mostly a config change, and the seams hold when a provider's SDK or contract shifts underneath — the prompt and eval tuning a swap still needs is a known, fast loop, not a rewrite. Observability and budget limits are built in from the start.

PLACE YOUR ORG

circle where you are

- ☐ 1 · Shadow AI is bolted in ad hoc, with providers hardcoded.
- ☐ 2 · Sanctioned an AI integration pattern is written down and reused by more than one team.
- ☐ 3 · Standardized provider selection lives in config with a fallback, behind a published contract.
- ☐ 4 · Integrated swapping a model is a config change, and you can trace which provider served a call.

ASK YOURSELF

Could we change our model vendor next quarter without rebuilding?

AI can route across providers and log every hop. What cost, latency, and capability a problem actually needs is a trade a person still makes.

THE MOVE

Put the seam where the swap will land.

Assume every provider and model will change, and build the seams that absorb it without a rewrite.

Architecture

Circle the stage each component is at. · Sanctioned = documented intent, not table-stakes. · Integrated = human-confirmed; automation that flags or contains is fine, that decides is not.

COMPONENT	SHADOW	SANCTIONED	STANDARDIZED	INTEGRATED	
Provider Abstraction <i>How thoroughly the architecture isolates model and provider specifics from the code that consumes AI.</i>	Provider SDK called directly at each use site; vendor names and model IDs hardcoded throughout.	A few teams share a client, but consumers still bind to one vendor's SDK shape.	Consumers call a published internal contract; provider choice lives in config with per-route rationale.	Model-agnostic operation is the default; a vendor swap is a config change at the wiring layer, with prompt and eval re-tuning a known, fast loop rather than a rewrite.	1 2 3 4
Integration Patterns <i>Whether common AI shapes (chat, retrieval, tool-calling) exist as reusable, documented building blocks.</i>	Each integration hand-wired where it fit that day; nothing named, nothing reusable.	Recurring shapes emerge; one team writes down a chat or retrieval pattern others copy.	Integration patterns documented and reused across teams as the expected starting point.	Patterns ship as platform components; new AI work composes them instead of rebuilding from scratch.	1 2 3 4
Model Routing & Fallback <i>How requests are matched to a model tier and what happens when a route fails.</i>	One model picked once for everything; a provider's bad hour breaks the integration outright.	Manual model choice per feature; no tiering, and recovery from an outage requires a code change.	A tiered router treats cheap, medium, and smart as model-agnostic tiers and walks a logged fallback chain.	Routing adapts to availability; cheaper models promote via config and fallbacks are regularly exercised.	1 2 3 4
Cost & Latency Control <i>How spend and response-time targets are set, defaulted, and enforced at the architecture layer.</i>	Cost and latency chosen once and forgotten; no per-route targets or visibility.	Teams notice spend after the bill arrives; rough latency awareness but no enforced ceilings.	Per-route cost and latency budgets documented; small-model-first defaults lower spend by design.	Budget enforcement and latency targets sit inside the call path and reject or downgrade requests that exceed them.	1 2 3 4
Observability of AI Calls <i>What is captured about model calls (route, fallback, tokens, latency, failures) to debug and reason about behavior.</i>	Model calls opaque; a failure surfaces as a generic error with no trace of what was called.	Ad hoc logging on a few calls; no consistent record of model, route, or fallback taken.	Route selection and fallback chains logged in a documented, queryable form across services.	Every call emits tier, tokens, latency, and fallback; this telemetry feeds routing and budget decisions.	1 2 3 4
Contract Versioning & Evolution <i>How the published AI interface changes over time without forcing consumers to rewrite.</i>	No stable interface; a prompt or response-shape change silently breaks downstream callers.	An informal shared shape exists, but changing it means hunting down and editing every consumer.	The internal contract is versioned with documented deprecation, so consumers migrate on their own schedule.	Interface and transport changes flow through compatible versions; old and new consumers run side by side during migration.	1 2 3 4

Team Capability & Culture

How AI fluency, learning, and psychological safety spread across a team.

STAGE 1 Shadow	AI fluency lives in individuals, not the team. A few people are good; most aren't, and there's no shared sense of what good looks like. One or two become the team's 'AI person.' The team can end up with output it can't explain — and when those people leave, the capability leaves too.
STAGE 2 Sanctioned	The org invests in basic AI literacy and some groups build informal habits — a real but uneven step. Underneath, status dynamics shift: junior engineers worry about looking dependent, senior engineers rework a craft they spent years building. It already shapes who feels safe asking for help.
STAGE 3 Standardized	AI literacy becomes part of onboarding, and teams write down their practices — when to use these tools, when to skip them, how to review output. The leader's real work here is building the safety to flag a confident-looking answer that's wrong without it costing status. Without that safety, the honest self-assessment everything else depends on never happens.
STAGE 4 Integrated	The team keeps adjusting its practices based on what works, and a new capability gets discussed before it spreads. Learning happens on the work itself — in review, in pairing, in arguing over a tricky output. The team also watches its own cost: reviewer load, the strain of supervising agents, the skills people worry about losing.

PLACE YOUR ORG

circle where you are

- ☐ 1 · Shadow one or two people carry the AI fluency; the team doesn't.
- ☐ 2 · Sanctioned the org has put real money into AI literacy, however unevenly.
- ☐ 3 · Standardized AI practices are written into onboarding, and it's safe to question AI output.
- ☐ 4 · Integrated a new capability gets a team conversation before it spreads, and the team tracks its own load.

ASK YOURSELF

Is my team actually learning AI, or just leaning on one person who is?

AI can spread the mechanics, draft the practices, and do a lot of the first pass. Building the safety that lets someone ask the honest question out loud is the leader's to create.

THE MOVE

Make the honest question safe.

Build enough safety that someone can ask 'is this right?' out loud — then fluency becomes something the whole team can question and improve.

Team Capability & Culture

Circle the stage each component is at. · Sanctioned = documented intent, not table-stakes. · Integrated = human-confirmed; automation that flags or contains is fine, that decides is not.

COMPONENT	SHADOW	SANCTIONED	STANDARDIZED	INTEGRATED	
Distribution of Fluency <i>How widely practical AI skill is spread across the team versus concentrated in a few people.</i>	A few individuals are skilled, most are not, and the 'AI person' becomes a bottleneck.	Some groups build informal habit, so capability grows but stays uneven across roles.	A baseline of fluency is expected of every role; coverage no longer rests on one person.	Fluency is broadly held and routinely refreshed; one person's departure leaves capability intact.	1 2 3 4
Safety to Challenge Output <i>Whether people can flag a confident-but-wrong AI answer without it being read as a status move.</i>	Questioning AI output feels risky; people defer to it or stay silent rather than look behind.	A few will push back, but doing so still reads as inexperience to some peers.	Flagging a wrong answer is an expected review behavior; leaders design explicitly for it.	Challenge is routine at every level, including disagreeing with the tool, and the team periodically checks whether people still feel safe doing it.	1 2 3 4
Cross-Level Help-Seeking <i>How AI reshapes who feels safe admitting a gap, and how juniors and seniors relate around it.</i>	Status around AI is invisible and unmanaged; asking for help reads as admitting a deficiency.	Tension surfaces openly: juniors fear looking dependent, seniors feel their craft devalued.	Norms frame asking for help as expected practice, and pairing makes it a normal default.	Help moves freely in both directions across levels; seniority and AI skill are decoupled, and the team notices when status starts to block asking again.	1 2 3 4
Skill-Building Mechanism <i>The mechanism by which the team actually grows AI capability over time.</i>	Learning is private trial and error; what one person discovers rarely reaches anyone else.	The org funds basic literacy training, but it sits apart from real day-to-day work.	Structured teaching is reinforced on real tasks through scheduled review and pairing.	Learning happens on the work itself — structured review, pairing, debating tricky output — and the team tracks whether capability is actually spreading, not just activity.	1 2 3 4
Codified Practice <i>Whether shared written norms exist for when to use AI, when to skip it, and how to review its output.</i>	No shared sense of what good looks like; each person sets their own approach.	A few groups hold informal rules of thumb, but nothing is written or applied consistently.	Norms for use, skip, and review are documented and built into onboarding for new hires.	The playbook is versioned against real outcomes, and onboarding updates whenever it changes.	1 2 3 4
Adopting New Capability <i>How a new AI tool or technique enters the team's regular way of working.</i>	Individuals adopt new tools on their own; the team has no view of what is in use.	Adoption is permitted and visible, but driven by individual initiative more than the team.	New capability is checked against documented criteria before it enters regular use.	New capability is trialed against the documented criteria, and the team reviews how the last few adoptions actually landed before the next one spreads.	1 2 3 4
Well-Being & Sustainability <i>Whether the team tracks the human cost of working with AI: reviewer load, supervision strain, skill erosion.</i>	Reviewer load and supervision strain go unnoticed; fears about losing skills stay private.	Strain is acknowledged in conversation but not tracked or acted on in any consistent way.	The team names the risks and sets norms on review load and which skills to protect.	The team monitors load, supervision strain, and at-risk skills, and adjusts how it works in response.	1 2 3 4

SCORE IT

Where you landed — and what to move first

Rolling it up. A dimension sits at the stage most of its components clear; the components below that line are your first moves. Score the components for your two or three focus dimensions first — you don't need all ten in one sitting.

Exclude any N/A components before you count. On a tie between two stages, the dimension sits at the lower one.

Mark the stage each component is at (1 Shadow – 4 Integrated), note the dimension's overall stage, then commit to one or two moves.

#	DIMENSION	COMPONENTS AT SHADOW / SANCTIONED	DIM STAGE
1	Governance & Risk		1 2 3 4 N/A
2	Agent Autonomy		1 2 3 4 N/A
3	Operating Model		1 2 3 4 N/A
4	SDLC Integration		1 2 3 4 N/A
5	Leadership		1 2 3 4 N/A
6	Tooling		1 2 3 4 N/A
7	Metrics & Measurement		1 2 3 4 N/A
8	Security & Data Handling		1 2 3 4 N/A
9	Architecture		1 2 3 4 N/A
10	Team Capability & Culture		1 2 3 4 N/A

OUR FIRST MOVE

Dimension _____ Component _____ From stage ____ By when _____

Suggested first session. 90 minutes — take your three lowest-confidence dimensions, have people pre-score the components solo, then reconcile only the disagreements. Half-day — all ten, split the pre-work across the four owner groups on the map page (Exec / Delivery / Platform / Evidence).

Where most orgs are — and the first move from here

Across these ten dimensions, most organizations sit at Stage 1 or 2: AI is in wide use, loosely permitted, and nobody can say how well it's working. That's normal — the tools arrived faster than the habits around them.

Getting to Standardized and Integrated is rarely about buying more tools. It's the unglamorous work: writing the standards down, putting review gates where the risk actually is, and keeping the people who own those standards close to the work.

The first move is small: run the component grids for the one or two dimensions where the gap hurts most, and let the lowest components — not the dimension average — set the priority. Each dimension's MOVE tells you how.

Integrated doesn't hold itself.

The pressure to skip the gate just this once never goes away. Integrated doesn't mean the pressure is gone — it means you'd notice when you gave in. Re-score every quarter or two: the dimensions that moved, and the ones that slid back, are the real report.

Run this with your team.

Put the ten dimensions in front of the people who'd have to do the work. Place the org together, argue out the gaps, and leave with one or two dimensions you're actually going to move — and a name on each. That argument is worth more than the score.

This is a readiness and maturity tool, not an audit methodology: it tells you where you stand and what to move next, not what evidence an auditor would accept.

Where I can help

I'm Ed Schaefer, an AI-Native Transformation Advisor. I've spent my career as a coach and a delivery leader, and I now help engineering organizations adopt AI in a way that holds up — without giving up the human judgment that has to stay with a person.

I work with leaders, executives, and teams on the parts of AI adoption that aren't really about the tools: the operating model, governance and guardrails, where human judgment belongs, and the culture and capability that make any of it stick. Most of that playbook already exists — in agile, lean, DevOps, and leadership development. AI just raises the stakes.

What that looks like in practice

Run this maturity assessment with your leadership team, and turn where you land into a 90-day plan.

Design the operating model and guardrails for AI-native engineering — the governance, the roles, the review gates.

Work through a single dimension that's stuck, with the leaders and teams who own it.

Get in touch

If any of this is live for your org, I'd be glad to talk — whether that's running the assessment together or working through the one dimension that's nagging you.

edward-schaefer.com · linkedin.com/in/edward-schaefer

The AI-Native Engineering Maturity Model · v2.0 · edward-schaefer.com/ai-native-engineering-maturity-model