

A | A | A | A \ A \ A - A - A / A / A / A / A - A
A | A | A | A \ A \ A - A - A / A / A / A / A / A - A
A | A \ A \ A - A - A / A / A | A | A | A | A | A | A / A / A
A \ A \ A - A - A / A / A | A | A | A | A | A | A | A | A / A / A
A \ A - A - A / A / A | A | A | A | A | A | A | A | A | A / A / A
A - A - A / A / A | A | A | A | A | A | A | A | A | A | A / A / A
A - A / A / A | A | A | A | A | A | A | A | A | A | A | A / A / A
A / A / A | A | A | A | A | A | A | A | A | A | A | A | A / A / A
A / A | A | A | A | A | A | A | A | A | A | A | A | A | A / A / A
A | A | A | A | A | A | A | A | A | A | A | A | A | A | A / A / A
A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A / A / A
A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A / A / A

The Agentic Enterprise Architecture Model: simplify AI projects (and time spent talking)

Learn how Prompt/Workflow/Harness engineering, multi-agent architectures and Governance should be thought of in Enterprise AI.

Made with **GAMMA**

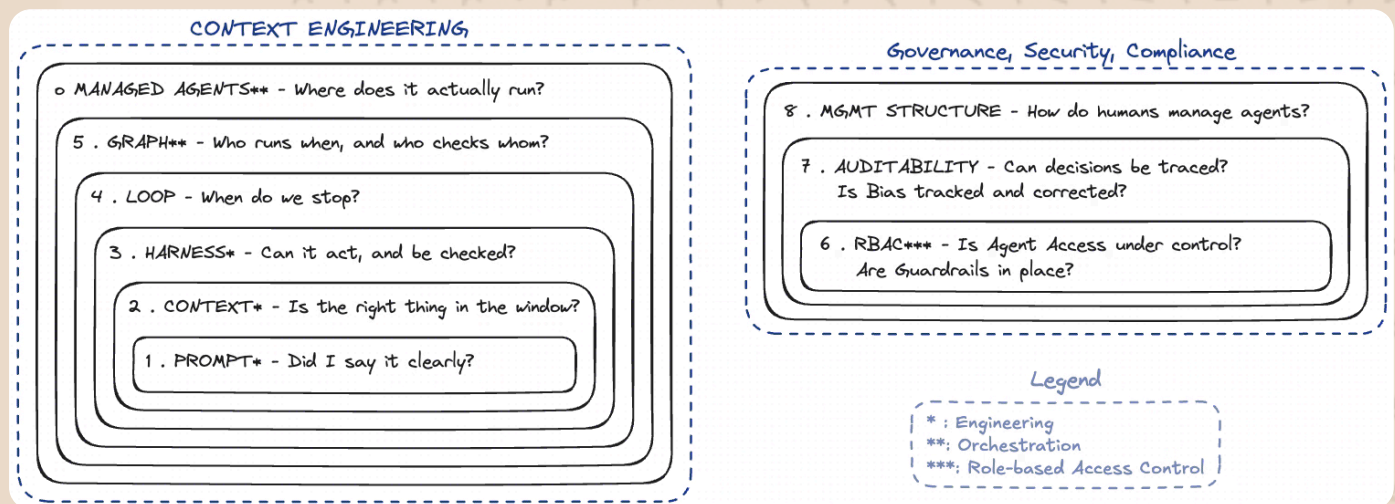
Where AI projects stumble, despite good intentions

AI adoption has moved past experimentation, but realized value remains uneven across industries due to a lack of attention to detail in the first stages. Access to the best models from central subscribers is not the real bottleneck. Enterprise AI breaks where the fidelity of the context is thin, data and process state validation comes late or is entirely "too late". And that is just during operations! When still in the earlier stages, uncertainty how to divide ownership over which outcomes, and a pervasive feeling of rapid prototyping is confused with Innovation. Innovation need not be slow, but it should not waste time either.

Microsoft's 2025 release "AI at Work Is Here. Now Comes the Hard Part" makes the central point explicit: the difficult part is that Ai, especially agentic approaches, work exceedingly well when the data inflows and the processing pipelines fit the company/team/department actual way of working. That clarity is what enables outputs to become dependable, auditable, governable and at the highest levels, completely hands-off.

The AEA model

The 8-shell Agentic Enterprise Architecture model shows how AI (xLM) applications need to be split up so that every part of a well-run organisation can maximise the amount of signal they contribute to the design and development process while minimising the time and repeats it takes to do that. Once AI is embedded into real delivery workflows, teams are no longer dealing only with ordinary software lifecycle concerns. They must manage new issues like prompt drift, context drift, outgrowing harness boundaries, agent coordination, autonomous action, and guardrails at a speed that humans cannot execute at.



The diagram shows nested layers of AI project components. The diagram is divided into two main sections: 'CONTEXT ENGINEERING' on the left and 'Governance, Security, Compliance' on the right. The 'CONTEXT ENGINEERING' section contains five nested layers: 1. PROMPT* - Did I say it clearly?; 2. CONTEXT* - Is the right thing in the window?; 3. HARNESS* - Can it act, and be checked?; 4. LOOP - When do we stop?; 5. GRAPH** - Who runs when, and who checks whom?; and an outer layer o MANAGED AGENTS** - Where does it actually run?. The 'Governance, Security, Compliance' section contains three nested layers: 6. RBAC*** - Is Agent Access under control? Are Guardrails in place?; 7. AUDITABILITY - Can decisions be traced? Is Bias tracked and corrected?; and 8. MGMT STRUCTURE - How do humans manage agents?. A legend at the bottom right defines the markers: * : Engineering, ** : Orchestration, *** : Role-based Access Control.

Figure 1: The mandatory 8-shell / 8-layer model image. This is the central visual anchor for the briefing.

The key failure domains for AI projects after the strategy has been set:

1

Context and Knowledge Mgmt

The elephant in the room. What kills AI enthusiasm?

1. Dirty data
2. Fragmented data
3. Poor data availability

These are most often the root cause of negative outcomes. Do you really expect *everyone* to become an AI-native Full-Stack SWE? Of course you don't. But data, context and Information habits are exactly what kills ambitious projects early: overburdening the users by not thinking through the lifecycle of the jobs that AI gets to handle.

2

Workflow Validation

Rapid prototyping in software is now a real advantage for everyone, but only if teams design validation and human review into the new workflows. AI as a term has been highly diluted, but we can agree on one idea: AI is expected to take on work, sometimes even entire jobs. So the natural first step is to answer: what is the job?

Once that is mapped, that is when acceleration begins. The second step is to know, where possible, when a job has been well done. To validate it. Humans do it via trust grown over months of collaboration. AI can't read human preferences yet. But it will be able to. Being ready for that time means understanding how to validate AI work today without wasting time. Thankfully, there are numerous solutions to that, but expect some change management overhead at the beginning!

3

Governance, Ownership, and Scale

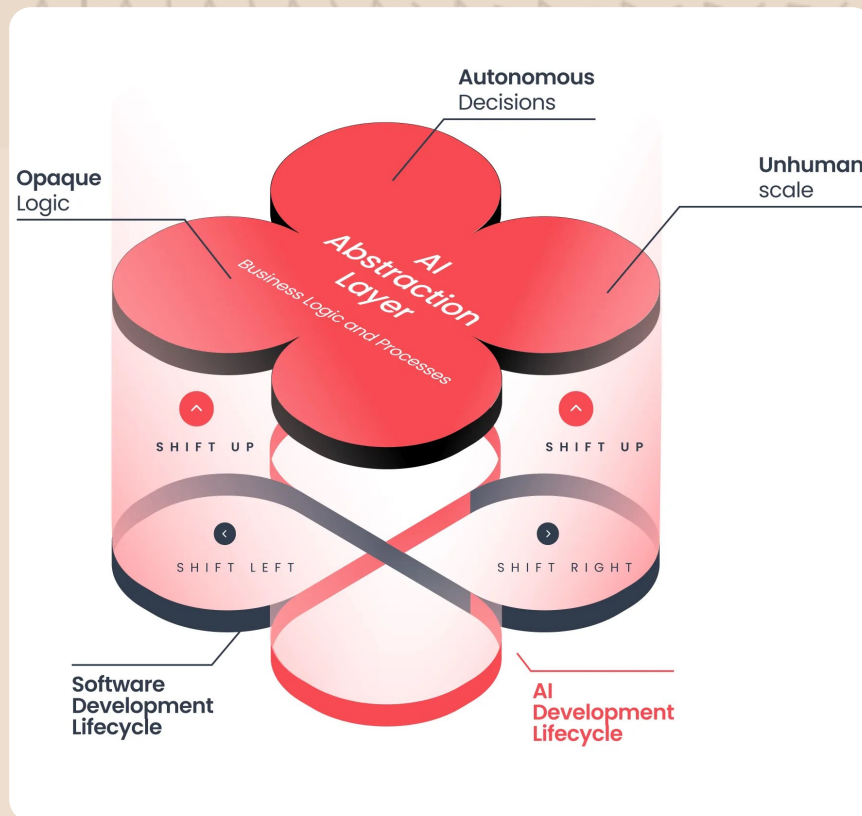
AI solutions as a foundation for decision making become risky when decisions travel faster and farther than the organization can explain, audit, or supervise.

AI explainability alone is unsatisfactory as a solution when a bridge somewhere has already collapsed.

Knowing your business's core invariants, or "must-be-truths", at the implementation stage is the biggest lever you have, so that each layer can work and nudge agentic outputs in that direction. That way, specifying and setting up Governance and Audit solutions in a lean way can be done in what feels like no time at all.

Shift-left versus shift-right

In software engineering before ubiquitous AI, "shift left" meant to make assumptions and review criteria explicit much earlier in the delivery process. "Shift right" meant to observe and course-correct based on live behavior, trace outcomes, and keep adjusting under changing conditions (aka. agile).



For AI to become an operationally dependable business pillar, only relying on **shifting right** makes everyone's job harder, as the pace of change clouds judgment. **Shifting left** risks getting left behind. The only surviving strategy therefore needs to ping-pong between both extremes not just during development and implementation, but for the entire lifecycle of the solution, which clearly changes roles and team structures over time. Three things need to continuously evolve at once:

- Dependable generation (layers 1-4): the Harness must self-improve, whether it is code, documents, ideation, scenario simulation: any AI-accelerated job will be bottlenecked if the end-user has to boss the AI tools around.
- Dependable parallelization (layer 5 - 6): a multi-agent architecture that keeps scopes explicit and can be centrally managed by the right person can relieve entire departments of AI anxiety.
- Dependable validation (layer 7+): treat review, tracing, and correction as a must-have.

Each of these three is bottlenecked by the other two. The difference lies in what your organization notices. Good PM and engineering practices (obviously accelerated by AI themselves) derisk that immensely.

AI readiness means data readiness

Ultimately, if your organization is not at least at level 3 data readiness, then simply subscribing to centralized AlaaS will not materially change outcomes. But what is level 3 data readiness?!

I don't put too much credence into formulaic readiness assessments and I don't believe you should either. Approaching an abstract gold standard in data readiness is a great goal, but as long as neither regulation mandates it (DORA, CRA, NI2, AIA etc.) nor it becomes a deep disadvantage in landing new business, fixating on assessments will only put you further behind. An experienced AI Consultant should be able to see where you lack in less than a full workweek (with a reasonable scope, not full enterprise transformation) and adjust timelines, expectations and project plans accordingly.

However, a self-assessment based on your own gut-feeling is valuable as a communication aid, just like the AEA 8-shell model is.

Here is my current (Aug 2026) preferred roster of questions for self assessment (with a bias toward sovereign data, which is NOT as simple as you'd think): [Stack Overflow's AI Maturity Matrix](#)

References

[1] IBM Global AI Adoption Index 2022 — <https://www.ibm.com/downloads/cas/GVAGA3JP>

[2] McKinsey, The State of AI in Early 2024 — <https://www.mckinsey.com/~media/mckinsey/business%20functions/quantumblack/our%20insights/the%20state%20of%20ai/the-state-of-ai-in-early-2024-final.pdf>

[3] Microsoft, AI at Work Is Here. Now Comes the Hard Part — <https://www.microsoft.com/en-us/worklab/ai-at-work-is-here-now-comes-the-hard-part>

[4] Microsoft Work Trend Index 2025 — <https://www.microsoft.com/en-us/worklab/work-trend-index/2025-the-year-the-frontier-firm-is-born>

[5] GitLab 2024 Global DevSecOps Report — <https://about.gitlab.com/resources/developer-survey/2024/>

[6] Harness, The State of Software Delivery Report — <https://cdn.prod.website-files.com/6222ca42ea87e1bd1aa1d10c/677e767b5ad65ea20c02bfe6/The%20State%20of%20Software%20Delivery%20Report.pdf>