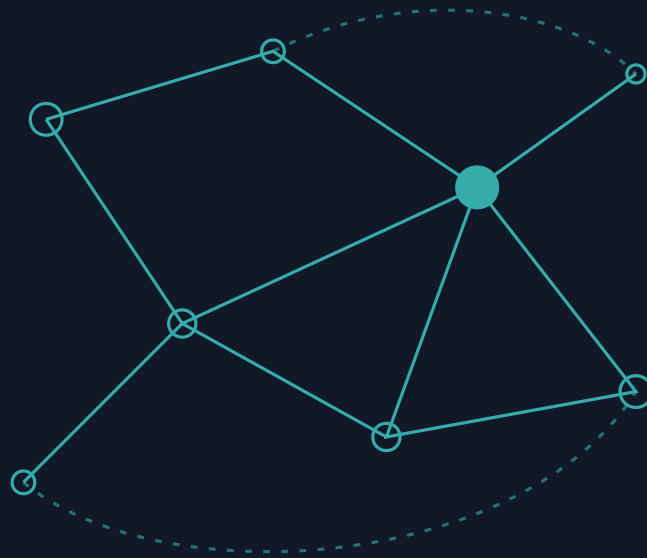




ESSAY · AO METHOD

Diagnose Before You Design

Why AO Method belongs at the start of organizational change

Three cases: introducing AI · creating shared services · growing while keeping a people-oriented culture

Most organizations begin change with an answer. "We need AI." "We need shared services." "We need to scale without losing who we are." Each sentence sounds like a strategy, but each one is really a proposed solution waiting for a problem definition. The technology, the structure, or the growth plan gets chosen first; the organization is expected to adapt afterwards.

This essay argues for the reverse sequence. Before an organization introduces AI, consolidates services, or expands, it should read itself as a system: where decisions actually flow, who can act without asking, where risk is truly held, which conversations are avoided, and which capabilities depend on a handful of people. That reading is what the AO Method provides as an organizational design diagnostic. It is not a delay before the real work. It is the part of the work that decides whether the rest will hold.

Three situations illustrate the argument. They are composite cases drawn from recurring patterns, not descriptions of a specific client, and they are meant to show a logic rather than to prove an effect size.

What the diagnostic actually looks at

A diagnostic is only worth starting with if it looks at something other diagnostics miss. Maturity assessments usually measure framework adoption: are the ceremonies in place, are the roles filled, is the tooling configured. AO starts from a different distinction. A team or a program can follow Scrum, Kanban, SAFe, or an ERP vendor's methodology perfectly while decision rights, interfaces, HR systems, and governance remain exactly as they were. Framework adoption and organization design are not the same object.

AO therefore looks at a small number of design questions:

- **Decision flow.** Value-stream thinking is usually applied to products and delivery. AO applies it to authority. Where does a decision originate, where does it wait, who can reverse it, and how often does it escalate by habit rather than by necessity?
- **The autonomy-risk boundary.** Local autonomy is safe when decisions are reversible and their consequences contained. Irreversible, high-blast-radius, or correlated decisions need guardrails or escalation. Most organizations have never drawn this line explicitly, so they either centralize everything or delegate indiscriminately.
- **Candor versus politeness.** Civil interactions can hide misalignment. A diagnostic that only hears the polite version of the organization will design for an organization that does not exist.
- **Informal power.** Formal governance shows how change is supposed to move. Informal influence networks show how it actually moves, stalls, mutates, or disappears.
- **Continuity.** Adaptive capacity is organizational only when it survives the departure of sponsors, informal leaders, and experts. If a capability lives in one person's network, it is a dependency, not a capability.
- **Change versus transformation.** Change acts on configurable structure through design and decision. Transformation acts on the living organization through learning and adaptation. The two are coupled, but confusing them leads to structural programs that expect cultural results, or cultural programs with no structural carrier.

None of these questions is exotic. What matters is asking them before the solution is fixed, because every one of them changes what the right solution looks like.

CASE 1

The company wants to introduce AI

The usual starting point

A mid-sized company decides to "introduce AI." A steering committee is formed, a platform is selected, and a list of use cases is compiled from departmental wish lists: automated reporting, a customer-service assistant, contract review, forecasting. Pilots start in parallel. Six months later, several pilots work technically, few have changed how the business operates, and one has produced an embarrassing error that nobody was clearly accountable for.

The technology did what it was asked to do. The problem is that nobody had decided what the organization was asking it to do, or who owned the consequences.

What the AO diagnostic reveals first

An AO diagnostic does not start with use cases. It starts with decisions. For each candidate area, it asks three questions that tend to reframe the whole initiative:

- 1. Who may accept, override, or escalate an AI recommendation?** If the answer is "whoever happens to be using the tool," the organization has delegated authority without designing it.
- 2. What is the first use case, and what downstream harm occurs if it is wrong?** This locates the initiative on the autonomy-risk boundary. A drafting assistant for internal notes and an agent that adjusts customer credit limits are not the same governance problem.
- 3. Is the sponsor seeking throughput inside the current model, or willing to redesign accountability and decision rights?** Both are legitimate. They are different programs with different costs.

The diagnostic then looks for three structural conditions that AI tends to amplify rather than fix.

Ambiguity that has not been partitioned. Humans remain better placed for meaning-making, legitimacy, ethical judgment, and accountability. Agents are suited to structured execution where goals, boundaries, and validation are explicit. When an organization has not separated these, it either asks AI to carry judgment it cannot carry, or it wraps simple execution in so many approvals that the benefit disappears.

Ontology debt. Terms such as "customer," "risk," "deal," or "done" often mean different things in different departments. People absorb this ambiguity through conversation. Agents inherit it and scale it. A diagnostic that surfaces contested definitions, and assigns an owner to each, prevents the AI initiative from industrializing an existing misunderstanding.

Missing social brakes. Technical execution runs fast; human governance loops run slowly. Without explicit checkpoints, sampling reviews, and exception-based escalation, the faster loop silently outruns the slower one, and agent authority expands through drift rather than decision.

Why starting here matters

Without the diagnostic, AI tends to become what can be called a "faster horse": existing workflows and artifacts run more quickly, but role boundaries, cadences, and decision rights stay unchanged. That is sometimes enough. But when leadership expects transformation from AI, it has to be designed as a change in who decides what, not as a software rollout. The AO diagnostic makes that choice visible before money and credibility are spent, and it produces a concrete artifact: a decision-by-decision map of where AI may act autonomously, where it must escalate, and who owns the result.

CASE 2

The company wants to optimize costs by creating shared services

The usual starting point

A group with several business units sees duplicated Finance, HR, IT, and Procurement functions. The business case is straightforward: consolidate into a shared service center, standardize processes, reduce headcount, and gain economies of scale. A target operating model is drawn, a ticketing system is configured, and service-level agreements are written.

Within a year, a familiar pattern appears. Routine requests move through queues that are slower than the old local arrangement. Business units start building shadow capabilities to get things done. Exceptions are resolved through personal relationships between managers. The savings on the spreadsheet are partly real and partly shifted into hidden coordination cost, rework, and delay.

What the AO diagnostic reveals first

AO treats shared services as an organizational integration problem, closer to a merger than to a process-standardization exercise. Before choosing a structure, the diagnostic reads how each unit currently makes decisions, which cultural and governance differences exist, and where the real dependencies sit. Different entities often bundle HR, Finance, and IT in materially different ways for good reasons. Standardizing before understanding those reasons removes adaptations that were doing useful work.

The AO Shared Services Model then offers a design vocabulary that separates three kinds of work that a classic shared service center tends to merge:

- **Platform** holds shared infrastructure, standards, compliance guardrails, and capabilities that delivery units should not rebuild. Its role is to enable, not to command. When routine exceptions require central approval, Platform turns into a control tower and absorbs decision rights that belong elsewhere.

- **Plexus** manages demand, prioritization, service interfaces, and escalation between Platform and the delivery units. It replaces both rigid ticket queues and informal favor networks with an explicit coordination layer where non-standard requests and capacity trade-offs are negotiated in the open.
- **Swarms** are the business-facing teams that consume shared capabilities and keep authority over reversible, contained decisions.

The organizing rule is that escalation follows risk, not function. A local team decides locally when a choice is reversible and contained. Irreversible, high-blast-radius, correlated, or precedent-setting decisions move through Plexus to Platform guardrails. This is the autonomy-risk boundary applied to enabling functions.

The diagnostic also tests continuity. If exception paths, service knowledge, and decision rights live in the heads of a few architects or informal fixers, the new structure is fragile regardless of how clean the organization chart looks.

Why starting here matters

Cost optimization that ignores decision flow tends to relocate cost rather than remove it. The AO diagnostic lets leadership distinguish real savings from savings that will reappear as decision latency, rework, tied-up capital, slower time-to-market, or key-person dependency. These categories can be measured and discussed with a CFO without inventing improvement percentages in advance. The outcome is not an argument against shared services. It is a shared service design that protects enterprise coherence while keeping the business able to act.

CASE 3

The company wants to grow while keeping its people-oriented culture

The usual starting point

A founder-led company of a few hundred people has grown on trust, proximity, and a strong sense of care. Leaders know most employees by name. Decisions happen in hallway conversations. Onboarding is informal but effective because new people sit next to experienced ones. The company now plans to double in size, open new locations, and possibly acquire a smaller competitor.

Leadership says, sincerely, that it wants to "keep the culture." The usual response is a values workshop, a culture deck, an employer-branding campaign, and perhaps a new layer of managers to handle the span of control. Two years later, employees say the company "doesn't feel the same," long-tenured staff feel bypassed, new hires feel the values are slogans, and decisions that used to take a day take a month.

What the AO diagnostic reveals first

The central insight is that a people-oriented culture in a small company is rarely only a set of values. It is a set of structural conditions that happen to be produced by size: short decision paths, direct access to leaders, visible consequences, and dense informal networks that carry both information and trust. Growth dissolves these conditions. Values statements cannot replace them.

An AO diagnostic therefore reads the culture as operating logic:

- **Where do decisions really happen?** Calendars and meeting portfolios show whether decisions sit in open forums or in a few trusted relationships. What works at 150 people becomes a bottleneck at 400.
- **Which parts of the culture depend on specific people?** The continuity test asks what happens when the founder, the long-standing operations lead, or the informal "culture carrier" is unavailable. If care and candor flow through one person, they will not scale.
- **Is the organization permeable?** A people-oriented culture can be warm and still closed. Diversity of membership only creates adaptive capacity when dissenting knowledge receives equivalent consideration in decisions. Rapid growth often brings people who think differently; a culture that cannot absorb their challenge will either expel them or split.
- **Is candor real or polite?** Close-knit cultures sometimes avoid conflict to protect relationships. At small scale this is manageable. At larger scale, undiscussables accumulate and become structural.
- **How does onboarding transmit operating logic?** Adaptive onboarding makes decision rights, escalation boundaries, safe-to-fail behavior, and the purpose of rituals explicit through pairing, modeling, and early bounded decisions in real work. Documentation alone does not carry culture.

Growth theory helps frame timing. Greiner's model describes how each growth stage eventually produces its own crisis, often a crisis of autonomy or control. Beer's Viable System Model helps identify which regulatory function is failing. AO positions itself as the design layer beside these lenses: it proposes interaction rules, recursive work zones, sensing mechanisms, and feedback loops that let the organization grow without converting care into bureaucracy.

Why starting here matters

Keeping a people-oriented culture during growth is a design problem, not a communication problem. If leadership does not deliberately rebuild the structural carriers of its culture, such as explicit decision rights at the right level, distributed sensing, protected channels for dissent, and continuity beyond key people, the culture will be replaced by whatever structure growth imposes by default. That default is usually more hierarchy and more control, introduced by well-meaning people under pressure. The AO diagnostic gives leadership the chance to choose the structure instead of inheriting it.

The common pattern across the three cases

The three situations look different: a technology decision, a cost decision, and a growth decision. Underneath, they share the same failure mode and the same remedy.

	Introducing AI	Shared services	Growth with culture
Solution chosen first	Platform and use cases	Consolidated center and SLAs	Values program and extra management layer
Hidden design question	Who owns decisions made with or by AI?	Which decisions stay local, which escalate?	Which structural conditions produce the culture?
Typical failure	Faster horse, or unowned errors	Savings shift into delay and shadow work	Culture becomes a slogan; decisions slow
Core AO lens	Ambiguity partitioning, autonomy gating, social brakes	Platform, Plexus, Swarms; risk-based escalation	Decision flow, permeability, continuity, onboarding
First diagnostic output	Decision map with autonomy envelope	Service design with explicit escalation rules	Map of culture carriers and their dependencies

In each case, the organization treats a design question as an implementation question. In each case, the decisive variables are decision flow, the autonomy-risk boundary, candor, informal power, and continuity. And in each case, the cost of discovering these variables late, after the platform is bought, the center is staffed, or the new managers are hired, is far higher than the cost of reading them first.

This is also why the diagnostic must come before method choice. Sector and function shape regulation, risk, cadence, and interface constraints, but they do not determine whether Scrum, Kanban, SAFe, a shared service center, or an AI operating model is the right answer. Method selection should follow diagnosis, not precede it.

Objections worth taking seriously

"We don't have time for a diagnostic." The diagnostic does not need to be long. Its purpose is to make the key design choices explicit, not to produce an exhaustive audit. The real question is whether the organization prefers to discover its decision bottlenecks in a bounded diagnostic or in a failed rollout.

"We already know our problems." Leaders usually know their symptoms. The diagnostic's value lies in connecting symptoms to design causes, and in hearing what the organization says when it is not being polite. Informal power, avoided conversations, and key-person dependencies are rarely visible from the top.

"A diagnostic is just consultants delaying the work." A diagnostic that ends in a report is a delay. An AO diagnostic ends in design decisions: where authority sits, where escalation is required, which experiments to run first, and what evidence will show whether they work. Interventions are framed as hypotheses to test, not as target states to impose.

"Can you guarantee results?" No honest diagnostic can. What it can do is bound claims to evidence, make assumptions visible, and define in advance what success and failure will look like. That discipline is itself a protection against expensive self-deception.

What starting with AO actually changes

Starting with AO as an organizational design diagnostic changes three things.

First, it changes the question. "How do we implement AI, shared services, or growth?" becomes "What must be true about our decision flow, risk boundaries, and culture for this to work?"

Second, it changes the sequence. Structure is designed to carry the intended behavior, instead of behavior being expected to adapt to whatever structure the solution imposes.

Third, it changes ownership. The diagnostic distributes responsibility across the system: executives own the enabling conditions, middle managers translate intent and protect flow, teams own delivery and adaptation, and individuals own candor and learning. Coaches and advisors inform trade-offs; they do not impose practices. An engagement designed this way is also designed for its own ending: the organization should leave more able to diagnose itself.

AI, shared services, and growth are all legitimate ambitions. Each one will reshape the organization whether or not anyone designs that reshaping. The point of starting with AO is to make sure the reshaping is chosen rather than suffered.

Pierre Neis is an organizational transformation advisor and the creator of the AO Method. He works with leadership teams on adaptive operating models, AI governance, and enterprise transformation through Menschgeist.