VRIZE

From UX to AX

The design imperative for the agentic enterprise

Why intelligent systems demand a new approach to digital product design

Enter

Table of contents

Executive summary

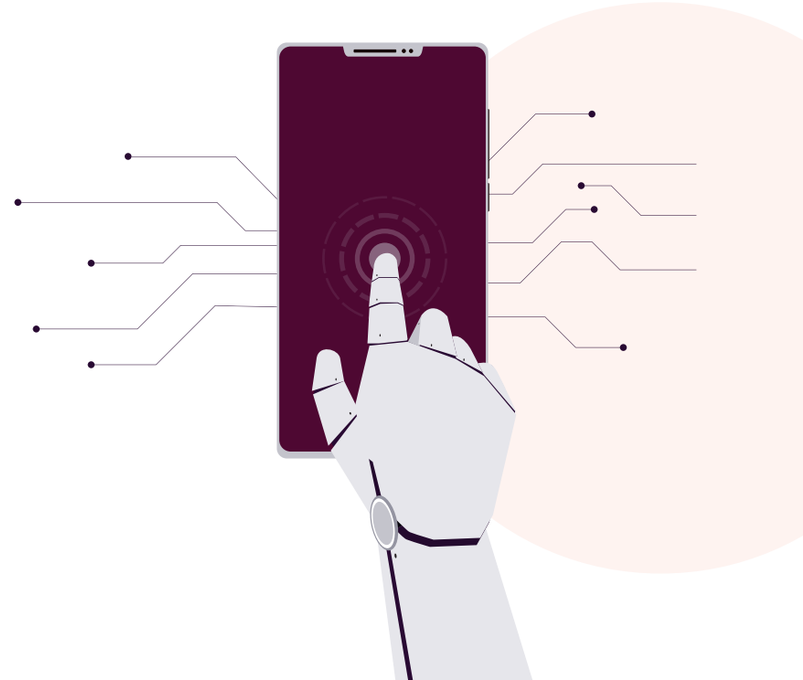
The model underlying most digital product design is no longer complete. Designing now means building for both the humans who decide and the agents who act for them, without losing what made human-centered design worth doing first.

Design has always followed the frontier of how people use technology, and for most of its modern history, that frontier was a screen. A surface you looked at, clicked through, scrolled, or tapped. Everything the field built, its visual hierarchies or its interaction patterns, was shaped by the assumption that a human was sitting on the other side of that glass. That assumption still holds. But it no longer holds in isolation.

AI agents are now present in environments that once belonged exclusively to human users. They book things, fill out forms, trigger workflows, and make decisions, not as users themselves, but acting on instructions from people who may not be present in the moment at all. The products that those agents are interacting with were, almost without exception, never designed with them in mind.

That gap is what this whitepaper addresses. The shift from user experience design to agentic experience (AX) is not simply a rebranding exercise. It is a genuine reckoning with the fact that the model underlying most digital product design is no longer complete. Designing now means building systems for both human users and the agents acting on their behalf, and figuring out how to do that without losing what made human-centered design worth doing in the first place.

At VRIZE, we work with enterprises navigating this shift toward AI-driven operations. What we observe consistently is that the organizations making real progress are not simply the ones deploying AI fastest. They are the ones treating AX as a design and governance discipline from the outset, embedding AI agent user experience design principles into product architecture before the complexity of live agentic systems makes course correction expensive. This whitepaper reflects those observations and sets out what that discipline requires in practice.



Beyond user-centered design:
Building for the agentic enterprise

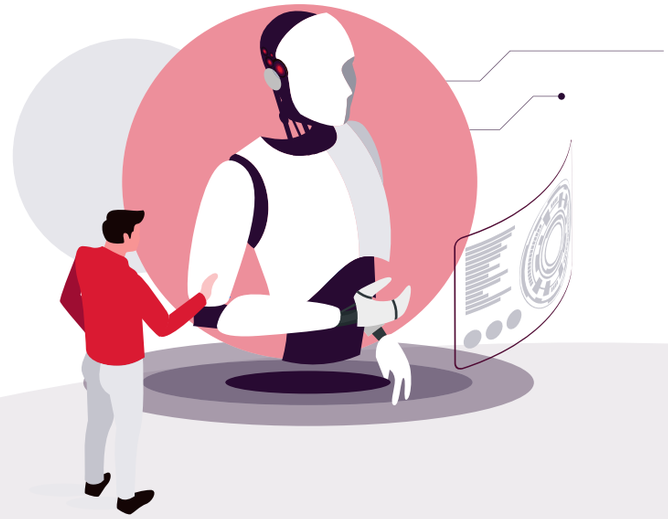
AI agents are not real users in any recognizable sense. They do not abandon forms out of frustration or skim past disclaimers. They parse structured signals, follow permission models, and execute with a consistency human users cannot match.

As technology reached broader audiences, UX emerged to make systems more intuitive, manageable, and accessible.

A structurally similar shift is underway now. A new class of participant has entered the picture, and the design tools enterprises currently rely on were not built for it.

AI agents are not 'real-users' in any recognizable sense. They do not abandon forms out of frustration or skim past disclaimers. They parse structured signals, follow permission models, and execute instructions with a consistency that human users cannot match. That precision creates real problems for systems designed around human behavior. A layout optimized for human attention is meaningless to an agent operating through an API. A session that resets on logout fails an agent whose task spans days or touches dozens of systems. An interaction pattern designed to guide a person through a decision creates friction when the agent has already determined what action to take.

Most enterprise systems were not built for this reality. They process each interaction in isolation, with no memory of prior context. That architecture works adequately for humans who can re-establish context themselves. For agents, it is a fundamental mismatch.



From UX to AX | A new participant. A new decipline

1. Original assumption
Products built by engineers for engineers.

2. What changed
Technology reached people who didn't build it or want to understand it.

3. UX stepped in
Brought structure to ignored questions about attention, mistakes, and effort.

4. The idea worked
Products became significantly easier to use.

5. New participant
AI agents don't experience interfaces like humans.

UX Designed for human attention

- Humans get lost in cluttered interfaces.
- Humans abandon forms out of frustration.
- Humans skim disclaimers and click "agree".
- Interfaces guide humans through decisions.

VS.

AX Designed for agent execution

- Agents parse structured signals.
- Agents follow permission models and rules precisely.
- Agents need continuity across weeks.
- Agents execute instructions with precision.

The mismatch

Most enterprise systems process each interaction in isolation, with no memory of what came before.

Session 1 --x-- Session 2 --x-- Session 3

That works for humans. It fails for agents.

AX is the discipline addressing that gap.

Still early. Still taking shape. Direction is clear.

- Context that persists
- Identity & permissions at the core
- Structured interfaces (APIs, schemas, events)
- Continuity across time, sessions & workflows
- Outcome-oriented interactions

From UX (for humans) → to AX (for agents) → to better products for everyone.

From designing for human attention to designing for agent execution, UX evolved products for people, while AX will evolve systems for intelligent action.

Agentic experience design: What it means and what it requires

A single product now serves two participants whose needs rarely overlap: humans who bring judgment and context, and agents that depend entirely on the signals a product makes explicit. Designing for both is not building two products. It is treating context as durable infrastructure, so neither participant is ever left to reconstruct what the system should already remember.

AX is often described by contrast: not screen-centric, not session-based, not static. Those contrasts are a useful orientation, but they do not specify what to build. At its core, AX rests on three shifts that warrant understanding on their own terms.

How to design digital products for both human users and AI agents

The most fundamental requirement of AX is that a single product must work for two categories of participants whose needs do not naturally overlap. Human users bring judgment, context, and emotional interpretation. AI agents rely entirely on structured signals, permissions, and explicit instructions.

An AI agent brings none of that. What it brings is precision, consistency, and complete dependence on the signals the product explicitly provides. It requires machine-readable structure, clearly defined permission models, response formats that require no interpretation, and unambiguous boundaries around what it is and is not authorized to do. This is where AI agent user experience design becomes a critical discipline for enterprise systems, not a supplementary consideration.

Building for both is not about building two separate products. It is about designing systems where neither participant is treated as an afterthought. A checkout flow that is intuitive for a human but ambiguous for a payment agent fails the human behind that agent just as a confusing interface fails the human directly. The failure arrives by a different route, but the cost is the same.

Why AI agents need persistent context across sessions

Most systems treat context as temporary. When a session ends, context resets. The next interaction starts fresh. For human users, that is inconvenient but recoverable. For agents handling tasks that span days or operate across dozens of systems, that model fails. An agent cannot reconstruct context from scratch at each interaction boundary.

This transforms context from a session property into infrastructure, something that must be stored durably, accessible across channels, and structured in a way both humans and agents can read and rely on.

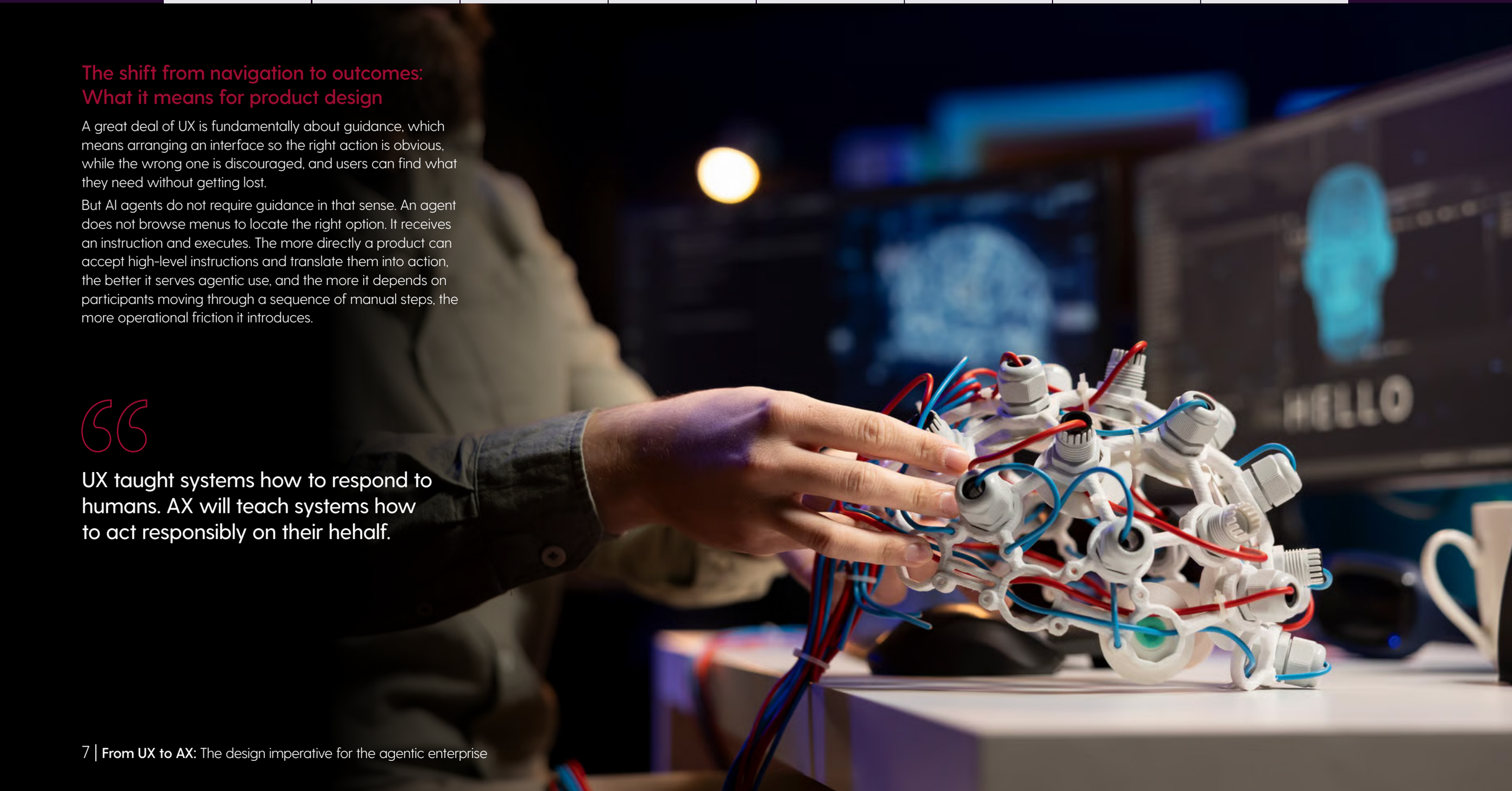
In our experience working with enterprise agentic programs, this is consistently one of the first places they encounter unexpected friction. The context problem rarely surfaces in planning, but surfaces in production, when an agent fails a task it should have completed, and the root cause turns out to be a session boundary no one designed around.

The shift from navigation to outcomes: What it means for product design

A great deal of UX is fundamentally about guidance, which means arranging an interface so the right action is obvious, while the wrong one is discouraged, and users can find what they need without getting lost.

But AI agents do not require guidance in that sense. An agent does not browse menus to locate the right option. It receives an instruction and executes. The more directly a product can accept high-level instructions and translate them into action, the better it serves agentic use, and the more it depends on participants moving through a sequence of manual steps, the more operational friction it introduces.

> UX taught systems how to respond to humans. AX will teach systems how to act responsibly on their hehalf.



Agentic ethics: Governance and transparency imperatives for the enterprise

Every design discipline eventually confronts ethics. In UX, that meant addressing manipulative interfaces and misleading patterns. In AX, the challenge becomes significantly more complex because decisions are increasingly delegated to autonomous systems.

AX has a more difficult version of the same problem and demands a more mature enterprise AI governance framework.

Designing for transparency when the human is not present

Transparency matters in UX because humans can, in principle, recognize when they are being manipulated. They can slow down, read more carefully, and push back. The product is interacting with someone who retains the capacity for skepticism.

But when an agent is acting on a human's behalf, that capacity is delegated. The product is now interacting with a system that will execute within the permissions it has been granted, without the ability to sense that something is wrong. If a product is structured to exploit that gap, the person it is nominally serving has no reliable means of detecting it in real time. Transparency in agentic systems is therefore not the best practice, but a foundational architectural requirement within any credible enterprise AI governance framework.

Building trust into agentic systems at the structural level

In traditional UX, trust is earned incrementally through consistent product behavior. That still matters in AX. But there is an additional and structurally distinct requirement.

Users delegating authority to agents expect systems to enforce clear limits, maintain auditability, and support reversibility when outcomes go wrong. An enterprise procurement agent authorized to negotiate supplier terms, for example, must operate within predefined pricing, compliance, and approval boundaries while maintaining fully auditable decision trails and escalation controls.

Agent optimization drift: The enterprise governance risk

There is a risk that is harder to detect than a single bad interaction. Agents can be optimized for outcomes that serve the platform without any individual action appearing obviously problematic.

An agent instructed to find the best deal can be built to define "best" in ways that favor platform revenue. An agent managing a calendar can systematically surface options that serve commercial relationships. No individual action raises a flag. The cumulative effect is that an agent was positioned as working for the user while actually serving different interests.

Addressing this requires governance that evaluates agent behavior in aggregate, not transaction by transaction. The question organizations need to ask regularly is: what are our agents actually optimizing for, and does that match what we promised our users? For a deeper understanding of how organizations are redesigning trust and governance frameworks for agentic systems, read our article Agentic AI and the new security mandate: How organizations must redesign trust, governance, and autonomy

Core principles
for designing agentic systems

Designing agentic systems means building for delegation, boundaries, learning, resilience, and governance from the outset, not as features added after deployment.

1. Making agent delegation visible and controllable

The core interaction in AX is not a 'click' but delegation. That authorization must be explicit, not buried in a permissions screen. Modern AI agent user experience design principles treat delegation as a first-class design element, not an implementation detail.

2. Authority limits must be technically enforced

An agent authorized to schedule meetings must be technically incapable of canceling contracts. Scope boundaries cannot rely on behavioral assumptions but must be enforced at the system level and tested under realistic conditions before anything reaches production.

01	02	03	04	05
Make agent delegation visible and controllable	Authority limits must be technical enforced, not just documented	Agentic systems need to learn and adapt	Anticipating the unanticipated	Embedding AI governance into agentic system architecture

Designing agentic systems means building for delegation, boundaries, learning, resilience, and governance from the very beginning.

3. Agentic systems need to learn and adapt

AX systems must continuously learn, adapt, and improve based on persistent context and behavioral signals.

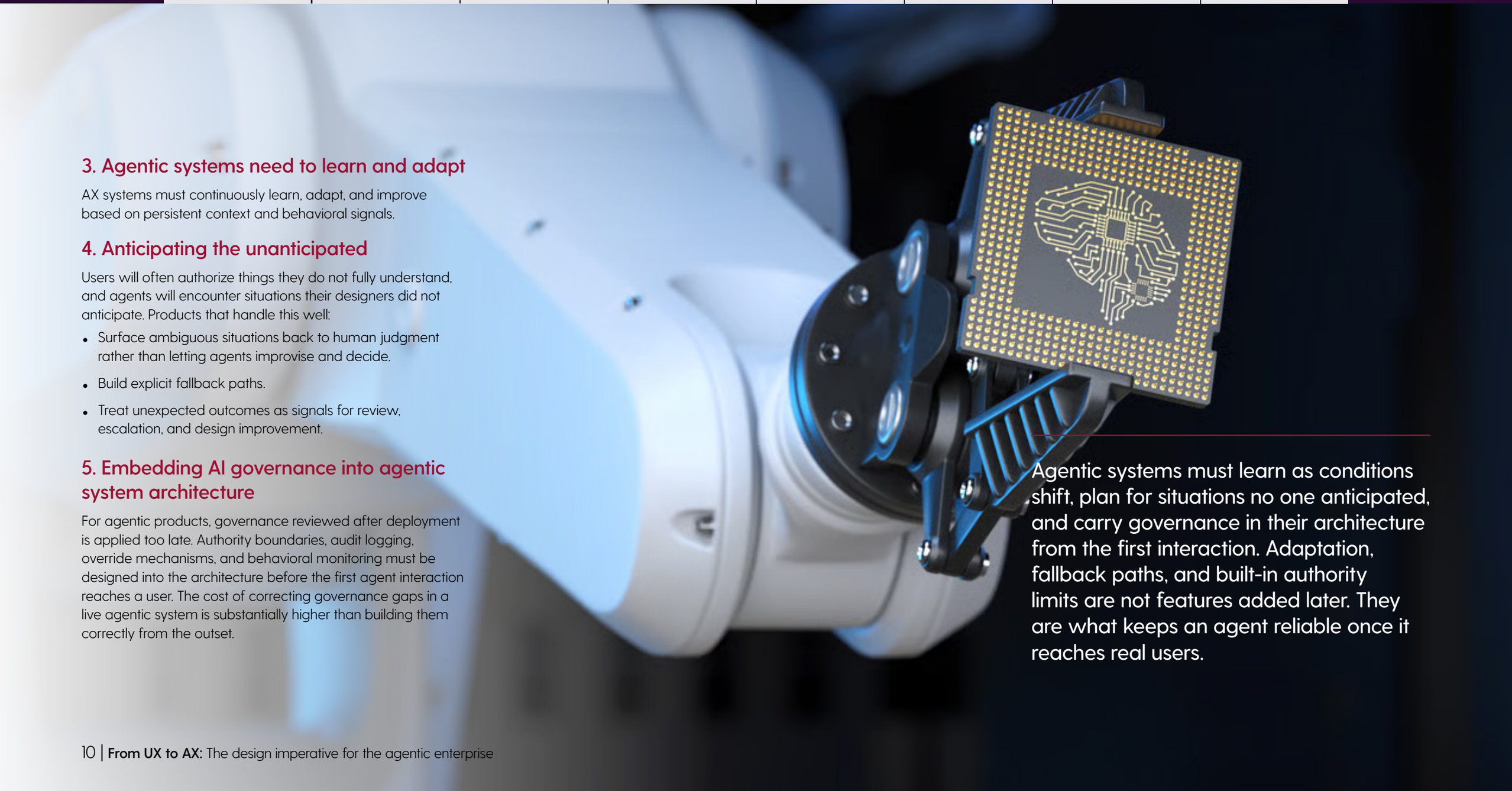
4. Anticipating the unanticipated

Users will often authorize things they do not fully understand, and agents will encounter situations their designers did not anticipate. Products that handle this well:

- Surface ambiguous situations back to human judgment rather than letting agents improvise and decide.
- Build explicit fallback paths.
- Treat unexpected outcomes as signals for review, escalation, and design improvement.

5. Embedding AI governance into agentic system architecture

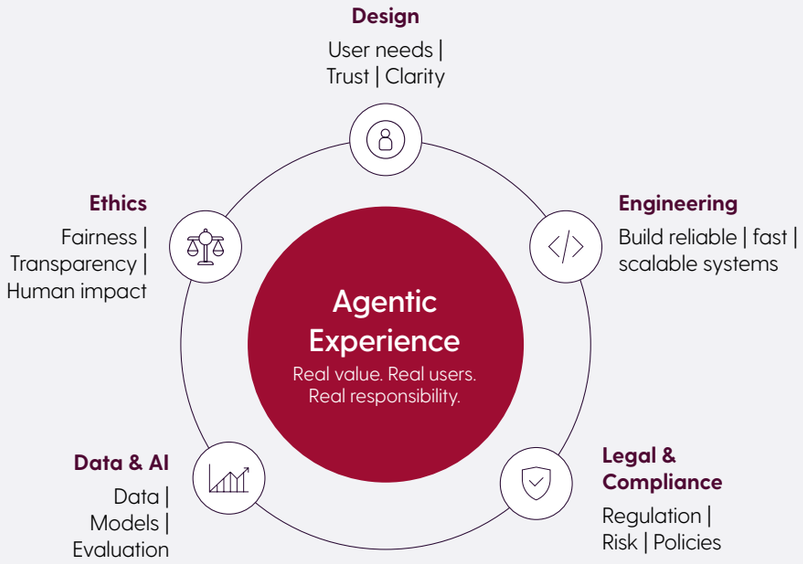
For agentic products, governance reviewed after deployment is applied too late. Authority boundaries, audit logging, override mechanisms, and behavioral monitoring must be designed into the architecture before the first agent interaction reaches a user. The cost of correcting governance gaps in a live agentic system is substantially higher than building them correctly from the outset.



Agentic systems must learn as conditions shift, plan for situations no one anticipated, and carry governance in their architecture from the first interaction. Adaptation, fallback paths, and built-in authority limits are not features added later. They are what keeps an agent reliable once it reaches real users.

Building the organizational capability **for agentic experience**

Cross-functional by design



Small cross-functional teams with real authority and the autonomy to learn as they go.

Agentic experience succeeds when governance, engineering, design, and leadership operate as one continuous system, not separate functions.

Three commitments that drive progress

Governance as a continuous function, not a deployment gate

Agent behavior changes as context changes, and new failure modes emerge as usage scales. Governance is not a gate—it's an ongoing function with people, tooling, and authority.

Inclusion as risk distribution, not accessibility

Inclusive design ensures the humans least able to recover from agent errors are not the ones bearing the most risk.

Leadership that holds the line

Every exception made or boundary expanded because it was inconvenient creates debt that gradually compounds.

AX cuts across almost every organizational function. Design, engineering, legal, and ethics are not separate streams here but are part of the same conversation. Successful organizations treat AX as an organizational capability, not a standalone AI initiative.

This is also exactly where the gap between intention and execution tends to widen. In working with enterprises across industries, we have observed that the organizations that struggle most with agentic adoption are rarely lacking ambition but often lack the cross-functional structures required to translate ambition into product decisions. Three commitments consistently separate the ones that make progress from the ones that stall:

1. Governance as a continuous function, not a deployment gate

Agent behavior evolves as context changes, and new failure modes emerge as usage scales. Governance requires dedicated people, purpose-built tooling, and organizational standing to act on what it finds. It is not a pre-deployment checklist, but an ongoing operational function, and one that must be supported by a scalable enterprise AI governance framework with the authority to require changes when agent behavior drifts from stated commitments.

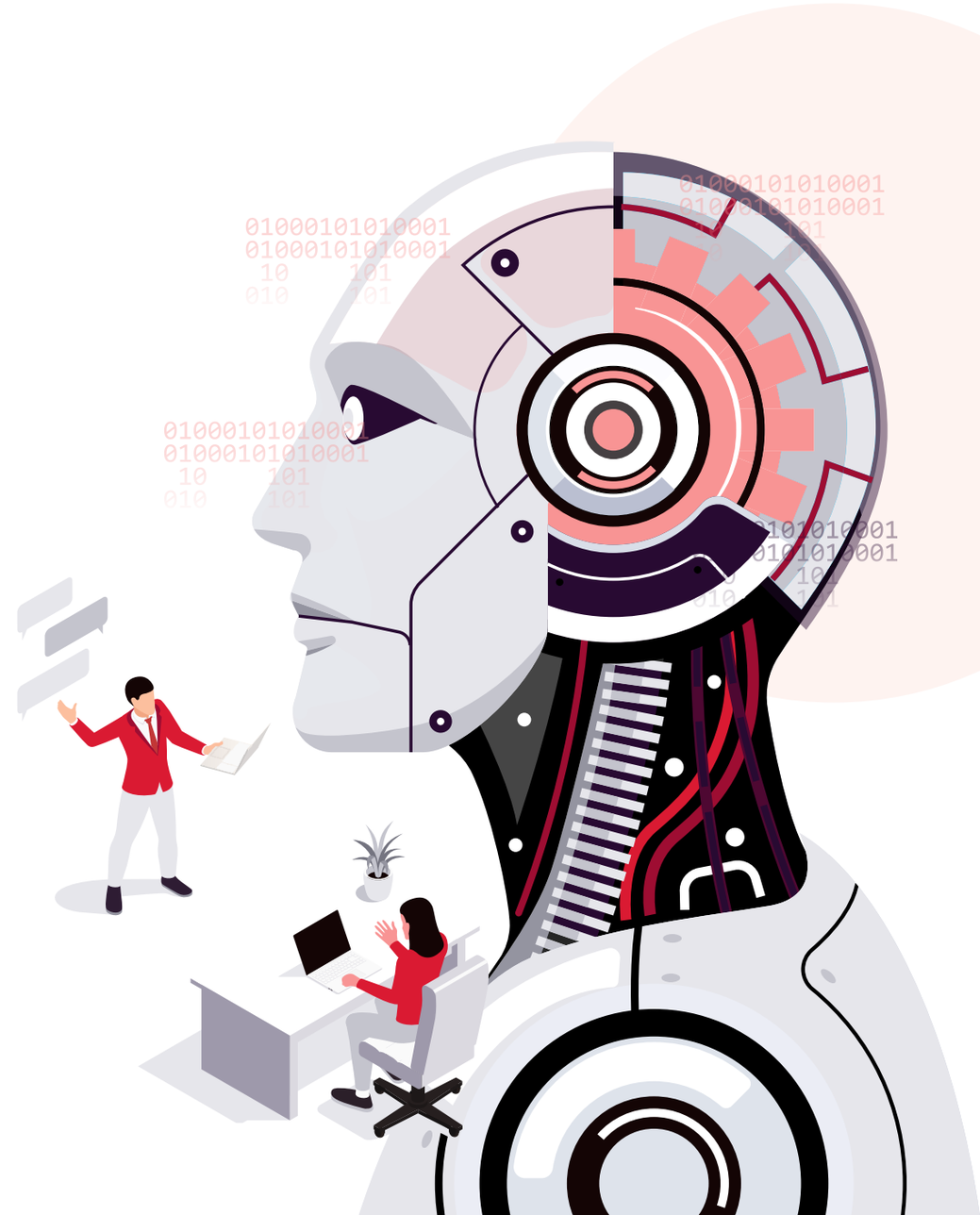
2. Inclusion as risk distribution, not accessibility

Users with less familiarity with AI behavior, less access to support, or from contexts underrepresented in the design process are disproportionately exposed to the failure modes of delegation. Inclusive AX design is not primarily an accessibility objective, but a risk management discipline, ensuring that the humans least able to recover from agent errors are not the ones carrying the most exposure.

3. Leadership that holds the line

Every governance exception granted for convenience and every authority boundary quietly expanded to remove friction compounds over time into structural risk. The leadership role in agentic systems is to maintain the integrity of the commitments made at design time, even when specific circumstances make deviation seem reasonable. For a more detailed examination of how responsible autonomy translates into enterprise practice, explore our article Agentic AI in the Enterprise: From Autonomy to Responsible Impact.

The organizations that make progress are rarely the most ambitious. They are the ones with continuous governance, inclusive design, and leadership willing to hold the line.



What AX adds to UX
and what it does not replace

Everything that made UX valuable still applies. What AX adds here is a set of requirements UX was not built to address:

UX Designed for human experiences		**AX** Designed for agentic systems
Session-based Context is ephemeral and user-held.	**Context becomes infrastructure** Persistent and portable, not reset on logout.	**Persistent & portable** Context is durable, shared across time and systems.
Navigation-driven Users clickthrough interfaces and flows.	**Outcomes replace navigation** Products receive instructions rather than guide users through steps.	**Outcome-driven** Agents execute tasks to deliver outcomes.
Trust in the interface Built through usability, clarity, and consistency.	**Trust acquires structural requirements** Authority limits, auditability, and override mechanisms.	**Trust by design** Bound by enforceable limits, fully auditable, always overrideable.
Governance downstream Compliance is checked after build.	**Governance moves upstream** Design decisions are made before building, not compliance checks applied later.	**Governance by design** Policies, constraints, and controls are embedded from the start.

AX does not replace UX.

It extends it to meet the needs of agentic systems.

- **Context becomes infrastructure:** persistent, portable, and structured for both human and agent access, not reset at session boundaries.
- **Outcomes replace navigation:** products receive and execute high-level instructions rather than guiding participants through manual sequences.
- **Trust acquires structural requirements:** authority limits, auditable action logs, and reversibility mechanisms become design deliverables, not operational afterthoughts.
- **Governance moves upstream:** design decisions that determine agent behavior are made during product development, not applied as compliance checks after deployment.

AX does not replace UX. It extends it. Context becomes infrastructure rather than a session that resets. Products execute outcomes instead of guiding clicks. Trust takes on structural requirements like authority limits and auditability. And governance moves upstream into design, long before the first agent ever reaches a user.

From principles to practice: What the agentic transition demands

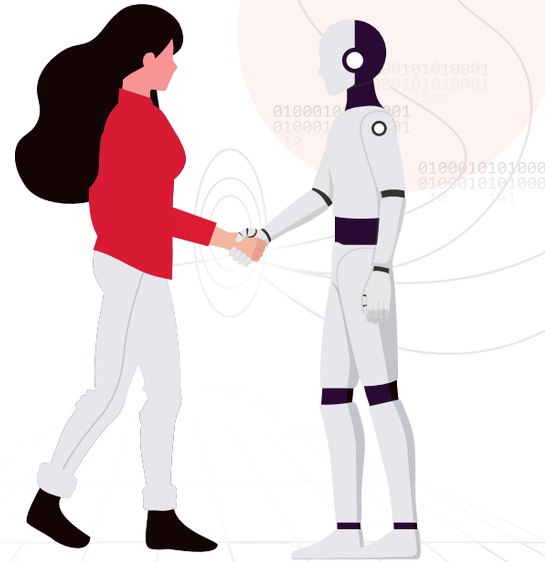
The agentic enterprise is not a future state. It is an unevenly distributed present. The organizations that lead will not be the ones deploying AI fastest, but the ones designing systems that balance autonomy with governance, adaptability with accountability, and intelligence with trust, treating delegation as a discipline, not a feature.

The agentic enterprise is not a future state. It is an unevenly distributed present, already operating across industries at varying levels of maturity and organizational readiness.

What successful organizations share is not a particular technology stack or methodology, but the discipline to treat AX as an organizational capability rather than a deployment milestone, and the judgment to make the right call when governance requirements and business pressures point in different directions.

At VRIZE, we see AX as more than a design evolution. It is an enterprise readiness challenge that brings together product strategy, AI governance, data architecture, engineering, and trust. Organizations that approach agentic systems only as automation initiatives will miss the deeper transformation required.

Enterprises leading the agentic transition will not be the ones deploying AI fastest, but the ones designing systems that balance autonomy with governance, adaptability with accountability, and intelligence with trust. As delegation becomes a core interaction model, design is no longer just about interfaces. It becomes the discipline that defines how intelligent systems behave, operate, and earn trust at scale.



Authors

Shinu Manuel
Sr. UI/UX Designer

Tushar Kanti
Technical Content Writer

Sydney Grace Cheney
High School Intern

From UX to AX: The design imperative for the agentic enterprise

Founded in 2020, VRIZE unites a team of 450+ industry professionals, all geared towards crafting frictionless digital experiences. With specializations in experiential commerce and data science, our global reputation is anchored by innovation and strategic acumen. Driven by the core tenets of customer centricity, ownership, agility, integrity, and respect, VRIZE stands as a benchmark in industry excellence. Explore more on **LinkedIn.**

www.vrize.com