NOVOCIRCLE

THOUGHT LEADERSHIP | MARCH 2026

The Long Arc

A practitioner's guide to the six stages of EA evolution and the shift that changes everything

Ryan Schmierer
Sr. Managing Partner, NovoCircle

novocircle.com

NovoCircle is a technology advisory practice specializing in Modern Enterprise Architecture and Intelligent Automation. This paper represents the independent research and practitioner perspective of the author. This paper does not evaluate or compare specific EA tools or vendors.

Executive Summary

Enterprise architecture did not begin with TOGAF. It did not begin with Visio, or with modeling tools, or with the arrival of ArchiMate. It began the first time a human organization became too complex for any single person to hold in their head, and someone had to build an external representation of it so the organization could understand and govern itself.

That problem is as old as organized human activity. The form of the external representation has changed with every era. The clay tablet gave way to the ledger. The ledger gave way to the blueprint. The blueprint gave way to the diagram. The diagram gave way to the model. The model acquired shared languages and common frameworks. And now, for the first time in this long arc, the model itself is changing in a fundamental way: not just in how it is built, but in what it can be.

This paper argues that enterprise architecture has evolved through six recognizable stages, each one lifting a constraint the previous generation accepted as permanent. Those six stages are: The Carried Map, The Age of Diagrams, The Connected Repository, A Common Language, The Governed Model, and Architecture Without Amnesia. Each transition followed the same pattern. A capability arrived that made the previous limitation look like a choice rather than a necessity.

The sixth stage, Architecture Without Amnesia, is the most significant shift in the arc. Every previous stage improved how well architecture was captured and communicated. This stage changes what architecture can know. Before this shift, every repository in the world suffered from the same structural limitation: it preserved conclusions, not reasoning. The trade-offs considered, the alternatives rejected, the assumptions made, the organizational context that justified a decision. All of it was discarded in the synthesis. Only the summary survived.

AI ends that era. The repository can now retain the source material, the reasoning, and the organizational context alongside the model elements they produced. Architecture becomes a living organizational memory, not a periodic snapshot.

This paper describes the arc and places the current moment within it. For practice leaders, the value is navigational: the arc reveals not just where the profession has been, but what constraint is currently defining your stage and what the next step actually requires. The organizations that recognize Stage 6 early will build practices

that set the standard for the profession in the decade ahead.

Introduction: The Problem That Never Changes

Every organization, at some point in its growth, produces more complexity than any person can hold.

This is not a failure of management. It is not a technology problem. It is a structural consequence of scale. When an organization grows beyond the span of direct observation, when decisions happen in rooms you were not in, when systems interact with other systems you did not build, when the consequences of a change ripple through structures you cannot see, the organization loses the ability to understand itself. It can still function. But it cannot govern what it cannot see. And the gap between what it thinks it looks like and what it actually looks like grows wider every year.

Every era has produced methods and tools for closing that gap. Not because the people of those eras were particularly visionary, but because the gap was costing them money, causing failures, and preventing them from making good decisions. The need to see clearly is one of the oldest organizational imperatives there is.

Enterprise architecture, as a named profession, is young. The discipline of externalizing organizational complexity so that it can be seen, understood, and governed is ancient. The first is a recent chapter in the second.

This paper traces that longer arc: from the first external records of organizational complexity through the six stages of the EA profession's own development. It argues that the profession now stands at the most consequential transition in that arc. Not because AI is powerful, though it is. But because for the first time in this long history, the external representation of organizational complexity can be something genuinely new: not a summary of what was decided, but a living record of how the organization thinks.

Understanding where this transition fits in the arc is not an academic exercise. It determines how clearly architecture leaders can see the step they are standing at, and whether they recognize the transition early enough to lead it rather than react to it.

PART ONE: BEFORE THE PROFESSION

The problem of organizational complexity did not arrive with computing. It arrived with scale. Every era produced methods for closing the gap between what an organization actually looked like and what it could hold in any single person's head. The form of the solution changed. The forcing function never did.

The first external records

The earliest accounting systems we know of were not built to track wealth. They were built to manage complexity. In Mesopotamia around 3,100 BCE, Sumerian scribes inscribed clay tablets with records of livestock holdings, grain stores, and labor obligations across thousands of acres of farmland. Royal warehouse managers in Egypt maintained detailed inventories in hieroglyphic records for the same reason: the alternative, holding the state's assets in memory, was not possible.

These were not primitive precursors to modern accounting. They were organizational intelligence systems built to answer a question that had no other answer: what do we actually have, and where is it?

The form of the external representation matters less than the discipline of maintaining it. The Mesopotamian clay tablet and the modern EA repository are, at the highest level of abstraction, the same thing: a governed external record of organizational reality, maintained because the organization cannot function at scale without one.

The accounting insight that still echoes

The shift from single-entry to double-entry bookkeeping in fifteenth-century Italy is one of the most important moments in the history of organizational intelligence, and it is almost never discussed outside accounting history. Single-entry bookkeeping recorded outcomes: what a merchant had, what they had spent. What it could not recover was *why*. The ledger captured results but discarded reasoning.

Double-entry bookkeeping changed that. By recording every transaction in two accounts simultaneously, once as a debit and once as a credit, the system created a structure from which the reasoning behind any outcome could be recovered. The result and the path to the result were preserved together.

This distinction, between a record that captures *what* and a record that also captures *why*, is one of the central ideas of this paper. We will return to it when we reach Stage 6. The relevance to enterprise architecture is not metaphorical. It is structural.

The blueprint as a coordination technology

The engineering blueprint represents a different moment in this history: not the externalization of knowledge that already exists, but the creation of a shared representation that coordinates work that has not happened yet. A blueprint does not describe a building that exists. It describes one that will exist, in enough detail that dozens of specialists, structural engineers, electricians, plumbers, can each do their part without coordinating directly with all the others. The blueprint is the coordination mechanism.

This is precisely what enterprise architecture does for technology-intensive organizations. The architecture model is a shared representation of what the organization is trying to become, structured in enough detail that technology teams, business units, security functions, and procurement can each make decisions consistent with each other without constant cross-team negotiation. The coordination logic the blueprint embodies, the idea that a shared external representation enables distributed coherent action, is the deepest justification for enterprise architecture as a discipline.

The pattern

Across all of these examples, the same dynamic recurs. Organizational complexity grows past the capacity of direct knowledge. An external representation is built in response: initially informal, gradually more structured, eventually governed. The representation enables the organization to see itself, make better decisions, and manage what would otherwise be invisible. Then complexity grows again and the limitation of the existing representation becomes clear. A new form emerges.

Enterprise architecture is the current chapter of this story. What this paper argues is that the chapter is turning.

PART TWO: THE SIX STAGES

The Arc from Informal Practice to Architecture Without Amnesia

Every EA practice in the world sits somewhere on this arc. Most have passed through some stages and are navigating a transition. A few have reached the frontier. Understanding the stages, what they look like from the inside, what value they deliver, what eventually makes them insufficient, is the precondition for understanding where your own practice is and what the next step requires.

The stages are developmental. You cannot skip one without paying the cost later. An organization that jumps from Stage 1 to Stage 4 because a consultant configured a tool for them will find that the Stage 2 and Stage 3 work still needs to be done. It has just been deferred and is now harder because it needs to happen inside an already-configured environment.

The stages also do not progress uniformly across an organization. A Stage 5 corporate EA function can coexist with Stage 2 PowerPoint diagrams in business unit SharePoints. The framework is most usefully applied to a specific EA practice and its immediate organizational context, not to an enterprise as a whole.

Figure 1. The six stages of enterprise architecture evolution.

The table below maps each stage at a glance before the detailed descriptions that follow.

Stage	Name	Organizing idea	What triggers the transition
1	The Carried Map	Architecture exists only inside one person	A retirement, resignation, or failure exposes the invisible map
2	The Age of Diagrams	Architecture survives the meeting, but not the year	Diagrams are always out of date and cannot be trusted for decisions
3	The Connected Repository	Elements have identity; the model has memory	The repository is growing but becoming harder to trust as it grows
4	A Common Language	Architecture becomes portable across people and organizations	Language is consistent; structural choices are not
5	The Governed Model	The repository earns the right to be trusted as data	Opportunity signal: repository is clean, AI tools are ready
6	**Architecture Without Amnesia**	AI ends the era of conclusions without reasoning	Capacity constraint is visible; the transition is technically possible now

Stage 1: The Carried Map

The organizing idea: Architecture exists only inside one person.

There is an architect in almost every organization at this stage, even if nobody calls them that. They may be the senior infrastructure engineer who has been there for twelve years. The IT director who personally reviewed every system implementation since the company outgrew its first server. The solutions architect who built the integration layer and is the only person who understands how it behaves under load.

These people carry an extraordinarily detailed map of the organization's technology landscape. They know which systems interact, which integrations are fragile, which decommissioned component is still silently receiving traffic from something that was supposed to have been updated two years ago. They are the organization's institutional memory. When a new initiative arrives, the project team calls them first: not because their title says architect, but because they are the only person who can answer the question "what breaks if we do this?"

The map is real. It is accurate. It is also invisible. It exists nowhere except inside that person.

At Stage 1, the organization functions because individuals compensate for the absence of shared structure. Systems were built to solve individual problems, with no reference to anything else. Integration is reactive. Connections are added as needs arise, not designed as part of a coherent whole. The result is what practitioners call spaghetti architecture: point-to-point connections, duplicated data, and systems that nobody fully understands in aggregate. The organization does not experience this as a crisis. It experiences it as normal.

What triggers the transition

The carried map has a fatal vulnerability: it leaves when the person leaves. A retirement, a resignation, or an acquisition can remove the map from the organization overnight. A 12-year infrastructure lead takes a new role at another company. Within three months the organization has no reliable record of which legacy integrations are still active, which decommissioned components still have live downstream dependencies, or why critical systems were built the way they were. The knowledge didn't leave gradually. It left the day the email account was deactivated.

Other triggers are less dramatic but equally common: a major initiative that requires integrating systems nobody has mapped, a new leadership mandate, or a security audit that reveals nobody can answer basic questions about data flows. In each case, the organization encounters the same discovery: the map is in one person's head, and it can't function strategically on that basis.

What has to change

The first change is cultural, not technical. The organization has to accept that architecture is a shared function, not an individual attribute. The tool is secondary at this stage. A SharePoint wiki, a shared notebook, a simple diagramming file: any of these can hold Stage 1's first external record. The commitment to maintaining something is more important than what that something is built in.

Stage 2: The Age of Diagrams

The organizing idea: Architecture survives the meeting, but not the year.

The most common Stage 2 tool is not Visio. It is PowerPoint.

This observation surprises some practitioners, but it should not. When an organization commits to capturing its architecture for the first time, it reaches for the tools it already knows how to use. Architects build system landscape slides in PowerPoint. They create data flow diagrams in Word documents. They maintain Visio files on SharePoint. Some teams use draw.io or Lucidchart for collaborative sessions. What all of these have in common is that they are presentation and illustration tools being

asked to do the work of an architecture repository.

These tools share a fundamental limitation: a box is a shape. It has no identity, no type, no properties, and no persistent relationships. The "Payments Application" box on the system landscape slide and the "Payments Application" box on the integration diagram are two independent rectangles with the same label. Change one and the other stays wrong. The result is a pathology that every Stage 2 practitioner recognizes: diagrams age into fiction.

Consider a PowerPoint diagram posted to SharePoint after an architecture review, found by a new developer eight months later. The diagram wasn't wrong when it was created. It was a point-in-time snapshot, accurate on the day it was drawn. But three other projects were already in flight that would change the landscape, and nothing in the format provided any mechanism for capturing those changes. By the time the developer found it, it had never been updated and had no way to be. The developer built a service based on what they found and discovered that the application they integrated with had been decommissioned. The diagram didn't fail because nobody maintained it. It failed because it couldn't be maintained.

Stage 2 delivers genuine and meaningful value. For the first time, stakeholders can see the architecture without asking the person who carries it. That value is real. It is also time-limited.

What triggers the transition

The diagnosis is almost always the same sentence: "We have documentation but we can't trust it." The underlying problem is not that the diagrams are out of date. It is that the format provides no mechanism for keeping them current. When a system changes, nothing in a PowerPoint file automatically changes with it. Maintenance requires human memory and discipline. At scale, both run out.

What has to change

The shift from a presentation tool to a modeling tool is the most significant transition in the EA evolution. It requires accepting a fundamentally different mental model of what an architecture artifact is. The tool doesn't change first. The understanding of what a repository is, and why an element is different from a shape, has to come first.

Stage 3: The Connected Repository

The organizing idea: Elements have identity; the model has memory.

The move from Stage 2 to Stage 3 is a conceptual shift as much as a tool shift. In Stage 2, a box is a shape. In Stage 3, a box is an element: an object with a unique identity, a

defined type, properties that describe it, and relationships that connect it to other elements. That element exists once in the repository and can appear on any number of diagrams. Change its name once and it changes everywhere. The diagram is no longer the artifact. It is a view of the underlying model.

The value becomes clear in a crisis. When Y2K forced organizations to assess every system that processed date-dependent transactions, most found the exercise took months, and in some cases years, because nobody had a reliable map of what their applications actually did or how they connected. Two decades later, when COVID-19 forced organizations to shift to remote work overnight, IT leadership asked: what applications does the helpdesk team depend on to support customers, and can all of it run remotely? For most organizations, it still took more than a week to answer. For a team with a connected repository, it was a 30-second query. The repository didn't make the crisis smaller. It made the organization capable of seeing itself clearly enough to respond.

The market for Stage 3 tools spans a wide range, from platforms oriented toward practitioner modeling and deep configurability, to platforms oriented toward enterprise management and executive reporting. Both categories have genuine Stage 3 capability in the sense that matters: elements with persistent identity and queryable relationships. The difference between them becomes significant at Stage 5.

What value the organization gets

A shared source of truth the organization can actually reason from. Impact analysis becomes possible. Architecture review boards can evaluate proposals against a model rather than against individual diagrams. The architecture team can answer questions about the landscape without calling the person who built it five years ago.

What triggers the transition

Stage 3's limitation is not technical. It is human. The repository has been built by multiple people with different modeling habits. An application modeled by the infrastructure architect is named differently from the same application modeled by the application architect. Relationship types are used inconsistently. The repository is growing, but it is becoming harder to trust as it grows. The diagnosis: "Our repository is valuable but only we can navigate it. And even we're not always consistent with each other."

What has to change

A shared modeling language. The transition to Stage 4 is about giving the team a common vocabulary that is defined externally rather than invented internally, so that the work of one architect is readable by another without a guided tour.

Stage 4: A Common Language

The organizing idea: Architecture becomes portable across people and organizations.

The EA profession settled on a set of shared modeling languages that now define the standard vocabulary of the practice. ArchiMate, maintained by The Open Group, provides a shared notation for the business, application, and technology layers of enterprise architecture and the relationships between them. UML gives software architects a standard notation for systems and behavior. BPMN gives process architects a language that business analysts and developers can both read without a translation layer. SysML extends the model into systems engineering.

The shift from Stage 3 to Stage 4 is a shift from private vocabulary to shared vocabulary. What makes this work in practice is not training certifications. It is the combination of a common language, a consistent set of methods applied across the team, and clearly defined modeling standards that give every architect the same starting point. A new architect joining from a different organization can review existing models and contribute new ones within days: not because they have credentials, but because the language, the methods, and the standards exist independently of any single team.

Most Stage 3 and 4 platforms support the major modeling languages to some degree. The depth of that support matters. Some platforms implement ArchiMate as a native metamodel, enforcing its typing rules and validating relationships according to the specification. Others support ArchiMate as a notation: the visual symbols are available, but the underlying repository does not enforce the semantic rules. A repository built with ArchiMate symbols but without ArchiMate semantics is a Stage 2 diagram with better icons.

What triggers the transition

The team has the language but not the scaffold. Every new architecture program starts from scratch on the same structural decisions: what does a "capability" mean in this organization? What is the correct top-level taxonomy for the application portfolio? The notation is consistent but the structural choices are not.

Two business units using ArchiMate produce application layers that cannot be merged because the element types were applied with different assumptions. The diagnosis: "We're all using the same language now but we're still producing inconsistent architecture. We need a framework, not just a notation."

What has to change

Adoption of reference frameworks, TOGAF for process and governance and industry reference models for structural starting points, combined with the governance practices that make those frameworks operational rather than decorative. This is the entry to Stage 5.

Stage 5: The Governed Model

The organizing idea: The repository earns the right to be trusted as data.

Stage 5 is the most complex to describe because it encompasses two meaningfully different internal positions. An organization at the entry of this stage has adopted reference frameworks, established governance processes, and built architecture practices that look mature from the outside. An organization at the far end of this stage has something more specific: a repository whose contents are semantically consistent, machine-readable, and trustworthy as a data source. These are not the same thing.

The gap between them is invisible until you try to do something automated with the repository, and then it becomes the only thing that matters.

The entry position: framework adoption

TOGAF, BIAN, TMForum, DoDAF, UPDM: these reference frameworks change the economics of the architecture practice. The first six months of a new architecture program shift from defining what architecture means to doing architecture work. The Architecture Review Board runs. Governance processes are defined. Standards exist on paper.

But the repository tells a different story. Here is what that looks like in practice. The EA team is preparing the quarterly portfolio report for the CIO. They pull the data directly from the repository. The numbers don't add up. Application counts differ between domains. The same system appears twice under different names. Capability mappings are incomplete in three business units. The team spends two days manually reconciling the data before the report can go out. This happens every quarter.

The modeling standards existed in a governance document. Nothing in the repository itself prevented an architect from doing it differently. The process was governed. The model was not.

A significant split in how tools approach this

At Stage 5, a structural difference between EA platform categories becomes consequential.

Dimension	Pre-configured platforms	Flexible / customizable platforms
Starting point	Governed structure from day one. Dashboards and reports ready immediately.	Blank repository. Metamodel, element types, and governance all configured by the team.
Configuration required	Minimal. Organization conforms to the vendor's model of architecture.	Significant. Team defines their own metamodel, languages, and validation rules.
Language coverage	Pre-defined. Domain-specific notations or specialized languages may not fit.	Any language, any metamodel. ArchiMate, UML, BPMN, SysML, and custom notations all supported.
Ceiling	Bounded by the vendor's mental model. Deep customization is constrained.	Unlimited. Any architecture domain is achievable given the configuration investment.
AI impact	Limited change. Governance structure was already in place.	Significant change. AI now handles the configuration work that was the primary barrier. The ceiling advantage is restored without the historic cost.

Figure 2. The trade-off between pre-configured and flexible EA platforms, and how AI changes it.

The end goal is the same for both approaches: a governed repository that can be trusted as data. The path is different. The pre-configured platform gets there by ensuring data enters a governed structure from the start. The flexible platform gets there through metamodel design and governance configuration, an investment that AI is increasingly able to support.

What drives the move from entry position to mature position within Stage 5 is often an AI augmentation initiative. Not because AI is required to reach the mature position, but because automated tools are the first to expose the gap between process governance and repository quality. A well-governed process with a poorly governed model produces unreliable automated outputs. That test, more than any maturity assessment, reveals what still needs to be done.

The far end of Stage 5: semantic consistency and machine-readable governance

The organization at the far end of Stage 5 has done something specific. It has taken a repository built up over years and made it semantically consistent. Every element has the correct type. Governance rules are encoded in the model itself, not documented in a policy on SharePoint. Validation runs continuously and generates work items for remediation rather than relying on architect discipline.

The same quarterly portfolio report that once required two days of manual reconciliation now runs directly from the repository. No cleanup. No reconciliation. No judgment calls about which version of a system name is correct.

But the more important change is what the manual process was hiding. When someone has to reconcile data before it goes to the CIO, they make choices: what to include, what to flag, what to smooth over. Sometimes those choices are innocent. Sometimes they are not. Either way, the report reflects someone's interpretation of the data as much as it reflects the data itself.

The governed repository eliminates that gap. When the model is the source, the dashboard is a window, not an argument.

This is the distinction that the existing EA maturity literature consistently misses. Most maturity models measure organizational governance: do you have an Architecture Review Board, do you have senior management involvement, do you have a defined architecture development process. None of them measure the semantic quality of the repository as a data asset.

An organization can score well on every dimension of the TOGAF Architecture Capability Maturity Model and still have a repository that a machine cannot reason over, because the governance they have built is governance of their process rather than governance of their model.

When AI augmentation becomes the objective, this gap stops being theoretical. A well-governed process with a poorly governed repository will produce an AI-assisted practice that generates unreliable outputs. The model is the data. The quality of the data determines the quality of everything built on it.

What triggers the move to Stage 6

This stage ends with an opportunity signal rather than a failure signal. The repository is clean and governed. AI tools are available and ready. The capacity constraint is visible. The question shifts from "how do we make the repository trustworthy?" to "what do we do now that it is?"

Stage 6: Architecture Without Amnesia

The organizing idea: AI ends the era of conclusions without reasoning.

Every EA repository ever built has suffered from the same fundamental limitation. It is a lossy compression of organizational reality.

To understand what this means, consider what actually happens when an architect builds a model. They take in an enormous volume of input: conversations with

stakeholders, observations of running systems, project documentation, design reviews, post-implementation analyses, informal organizational knowledge accumulated over years, disagreements, trade-offs, competing priorities, and historical decisions that made sense in a context that no longer exists. The architect synthesizes all of this input into the model.

The synthesis is the architect's central professional act. It is also an act of destruction.

The trade-off debate that required three Architecture Review Board sessions is encoded as a single relationship between two elements. The three alternatives considered and rejected are not in the repository. The assumption that justified an architectural decision in 2019 is not preserved alongside the element that decision produced. The organizational context that made a particular choice sensible at the time is gone. The repository is the conclusion. It is not the reasoning.

This limitation has been accepted as permanent because it was, in fact, permanent. Human architects can synthesize. They cannot also retain and structure the entire source context of everything they synthesized. The trade-off was not a choice. It was a constraint.

AI removes the constraint.

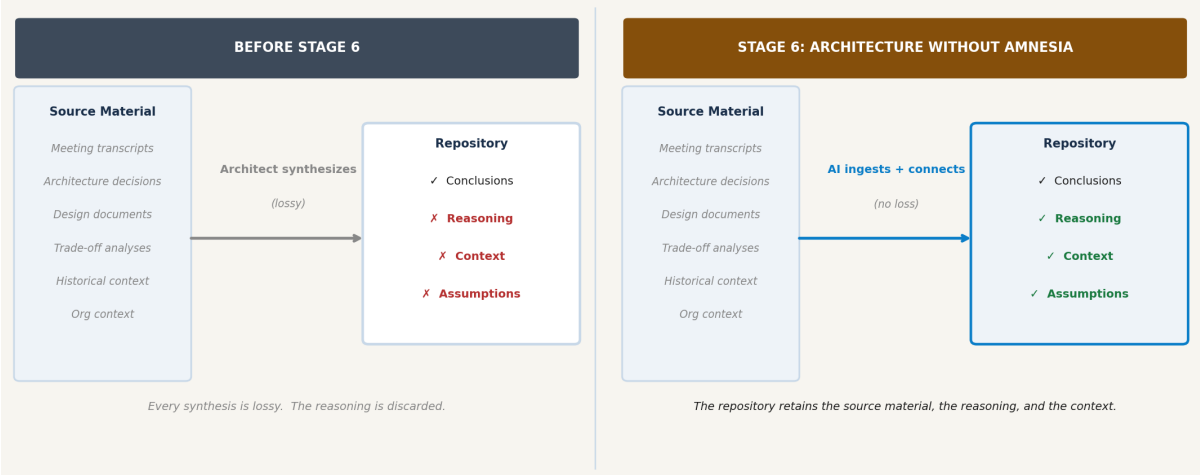

Figure 3. Before Stage 6, every repository discards reasoning and retains only conclusions. Stage 6 changes what a repository can be.

The two levels of Stage 6

There is a surface-level story and a deeper story about what AI augmentation means for enterprise architecture. Both are true. Only the deeper one is original.

The surface-level story is the productivity story. AI agents handle the mechanical, repetitive work that currently consumes the majority of architect time: transcribing meeting notes into models, formatting reports, checking elements against standards, populating properties from source documents, running impact analyses. This work is real and it takes real hours. Automating it frees architects for the work that actually requires them: analysis, judgment, stakeholder relationships, strategic decisions. This story is important. It is also the less important of the two stories.

The deeper story is about what a repository can become. AI systems can ingest, retain, and continuously process source material that has always been available to architects but has never been incorporated into the repository: meeting transcripts, architecture decision records, design documents, email threads, change request rationale, stakeholder feedback, trade-off analyses, historical context.

AI can connect this material to the model elements it informed, surface the assumptions embedded in the current architecture, and identify when an assumption that justified a decision has since been invalidated by a change in the organization.

The result is not a better diagram. It is a fundamentally different kind of artifact. The repository at Stage 6 is not a snapshot of what the architecture team believes the organization looks like. It is a living representation of what the organization actually is: how its technology landscape is structured, why it is structured that way, what decisions have been made and on what basis, and what the architectural implications of current changes are.

This is architecture with memory. Architecture without amnesia.

Consider what this changes in practice. An architect can ask: "What assumptions did we make when we decided to consolidate our regional data centers, and do any of those assumptions still hold?" The system pulls the architecture decision record, the stakeholder meeting transcript, the trade-off analysis, and the three alternatives that were rejected. It surfaces two assumptions that the organization's current infrastructure strategy has quietly invalidated. That conversation happens in twenty minutes.

Before Stage 6, it required a week of document archaeology, if the source materials still existed at all. Most of the time they did not.

The double-entry bookkeeping insight from Part One is worth revisiting here. Single-entry bookkeeping recorded *what* happened but not *why*. Double-entry bookkeeping added the ability to recover the reasoning from the record. That transition took five centuries to play out and reshaped commerce, capital, and organizational accountability. Enterprise architecture is making the same transition, at

a faster pace, with the help of a new kind of tool. The EA repository is becoming a double-entry system for organizational intelligence.

The architect's role shifts

This transition changes what architects do at the most fundamental level. Before Stage 6, the architect's primary role is synthesis. They take in the world and distill it into the model. After Stage 6, the architect's primary role is governance. The AI continuously synthesizes the organization's context into the model. The architect reviews, validates, directs, and governs that synthesis. They make the judgment calls that the AI cannot make.

This is not a smaller role. It is a different kind of authority. An architect who governs a trustworthy, continuously updated, context-rich organizational intelligence system holds a fundamentally different position in the organization than an architect who produces diagrams on request. The first is a strategic function. The second, however skilled, will always be vulnerable to the question "couldn't we just hire fewer of them?"

The organizations that understand this distinction, that Stage 6 is not about replacing architects but about changing what architecture means, are the ones who will lead the profession into its next chapter.

PART THREE: THE FRAMEWORK IN CONTEXT

Why Different Organizations Travel This Arc Differently

The six-stage model describes a general developmental arc, not a single fixed path. Four dimensions shape how organizations travel through it.

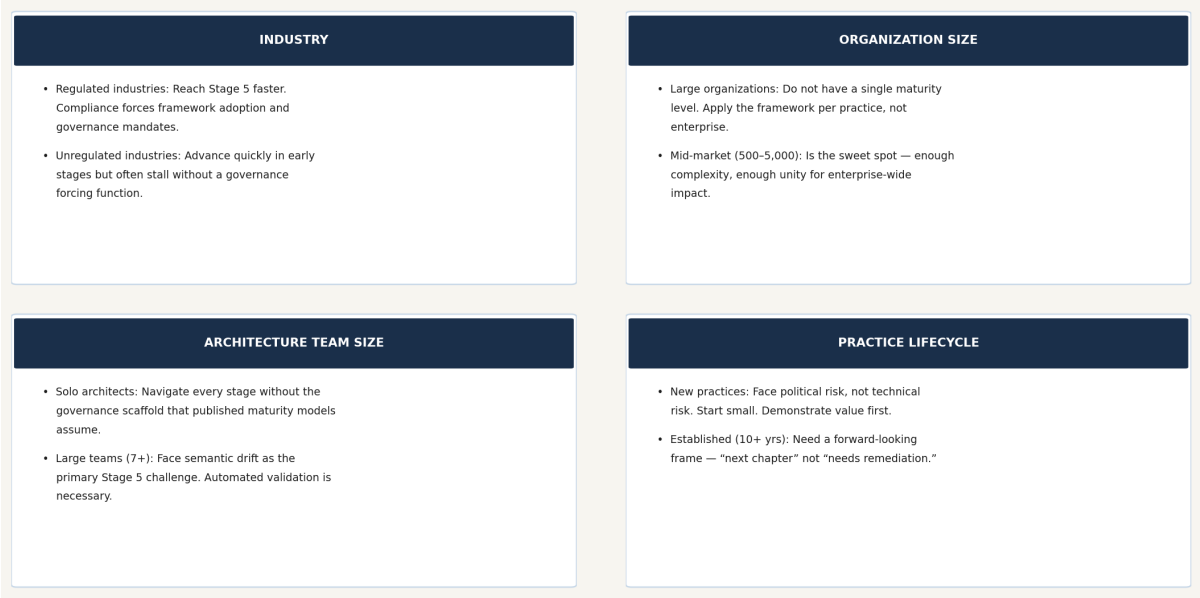

Figure 4. Four dimensions that shape how organizations travel the arc.

Industry is the strongest determinant of pace in the early stages. Regulated industries, financial services, healthcare, insurance, defense, and utilities, tend to reach Stage 5 faster because compliance forces framework adoption and governance mandates. The irony is that regulatory pressure can produce Stage 5 governance form with Stage 3 repository substance underneath: policies and review boards in place, but a model that cannot pass a machine-readability test. Recognizing this pattern requires looking at the model itself, not the governance documentation surrounding it. Unregulated industries move faster in early stages and stall in later ones: no forcing function for governance means Stage 3 and 4 gains often regress when the digital transformation program that funded them ends.

Organization size determines how the framework applies. Large organizations do not have a single maturity level. They have a distribution. A Stage 5 corporate EA repository can coexist with Stage 2 PowerPoint diagrams in business unit SharePoints. The model applies to a specific EA practice and its immediate context, not to the enterprise as a whole. The mid-market organization, 500 to 5,000 employees, is the sweet spot: enough complexity to need genuine EA capability, enough unity for a small team to have enterprise-wide impact. For these organizations, the Stage 5 to Stage 6

transition is the only path to comprehensive coverage without growing an already-stretched team.

Architecture team size shapes which challenges dominate. The solo architect, far more common than the formal literature acknowledges, faces every stage transition without the team, governance board, and senior management engagement that all published maturity models assume. Small teams of two to six are the model's sweet spot: large enough to need coordination, small enough for informal governance to work until the repository outgrows it. Large teams face consistency at scale as their primary Stage 5 challenge. Semantic drift is constant with seven or more architects modeling in the same repository; automated validation becomes necessary regardless of tool philosophy.

Practice lifecycle determines the primary obstacle. New practices face political challenges, not technical ones: the critical risk is starting too big, arriving with governance mandates before demonstrating value. Established practices three to ten years old are the most common candidates for the Stage 5 to Stage 6 transition: the repository is real, the practice has standing, and the capacity constraint is felt. Entrenched practices, ten or more years established, have the deepest resistance to stage transitions. The Stage 5 to Stage 6 conversation for these practices requires an explicitly forward-looking frame. The message cannot be "your repository needs remediation before AI augmentation is viable." It must be "here is the next stage in the arc your practice has been building for a decade."

PART FOUR: WHAT THE ARC TELLS US

Four Implications for Architecture Practice Leaders

The long arc has implications that extend beyond historical curiosity. For the EA practice leader sitting in 2026, it suggests four specific things.

The resistance is the signal

Every stage transition in this arc was resisted by people who were successful in the current stage. The diagramming practitioner who resists the modeling tool is not being irrational. Their diagrams work. The Stage 4 architect who resists TOGAF is not being obstructionist. Their practice functions. The resistance is evidence that the current stage is delivering real value, which is exactly when the constraint of that stage is hardest to see.

The most useful question an architecture leader can ask about their own practice is not "what is working?" but "what has started to feel like an unavoidable limitation that we've just learned to live with?" The answer is almost always the constraint that defines the next stage transition. The diagrams that are always out of date. The repository that requires a guided tour. The governance that lives in documents rather than in the model. The architecture that the organization consults rather than uses continuously. Each of these is the current stage announcing its ceiling.

Most practice leaders can name theirs without hesitation. The harder step is deciding to treat it as a signal rather than a fact of life.

The stage cannot be skipped but the pace can be accelerated

Skipping a stage has predictable consequences. The organization that jumps from Stage 2 to Stage 5 because a consultant configured a sophisticated tool will find that Stage 3 and Stage 4 work still needs to be done, now inside an environment designed for a more mature practice than the team is ready to operate.

But the pace of transition can be accelerated. The organizations that move through the early stages quickly share a common pattern: they produce visible value at each stage before asking for investment in the next one. They build the political capital for Stage 4 by demonstrating what Stage 3 makes possible. They build the case for Stage 6 by making the Stage 5 quality work visible as an outcome rather than an overhead.

AI is now accelerating the later transitions in a way that was not previously available. Configuration cost, which was a genuine barrier to Stage 5 governance on flexible platforms, is dropping. Repository quality remediation, which previously required months of manual rework, can now be substantially accelerated.

The transition from Stage 5 to Stage 6, which requires connecting a governed repository to an AI environment, is a weeks-long technical project for a practice that has done the Stage 5 work properly. The organizations that invested in repository quality before AI augmentation became a priority will find that investment paying off in ways that were not predictable when it was made.

Governance is only as strong as the model

The most consequential gap in current EA maturity thinking is the conflation of process governance with repository quality. Every major maturity model measures the organization's governance processes. None of them measure whether the governance is encoded in the model itself.

A governance policy that lives in a SharePoint document requires a human to read it, remember it, and apply it. An architect under deadline won't always do that. A governance constraint encoded in the model's validation rules cannot be bypassed without someone actively choosing to bypass it. The first kind of governance produces compliance in audits. The second kind produces consistency in the repository.

For AI augmentation, only the second kind works. An AI agent querying a repository inherits the governance that is structurally encoded in the model. It doesn't inherit the governance that is described in a policy document.

This is the single most important practical implication of the arc for organizations currently planning AI augmentation initiatives. Audit your governance, not your process documentation. The question is not "do we have an Architecture Review Board?" but "does the repository enforce its own consistency?"

The next transition is already possible

Every transition in the arc has followed the same logic: a capability arrived that made the previous limitation look like a choice rather than a necessity. The organizations that recognized it early built practices that looked, within five years, like the obvious way the profession worked.

The capability that enables Stage 6 exists now. AI systems can ingest organizational context at scale. Repository connections to AI environments are technically straightforward for a practice that has completed Stage 5. The epistemological shift, from a repository that remembers conclusions to a repository that preserves reasoning, is not a prediction about what AI will eventually be able to do. It is a description of what it can do today, applied to a repository that meets the preconditions.

The preconditions are Stage 5 in substance, not just in form. A semantically consistent, machine-readable, governed repository. Governance encoded in the model, not described in a policy. A team that understands what they are building toward and has made the quality investment the transition requires.

For architecture practice leaders, the question is not whether this transition will happen. It will happen across the profession. The question is whether it happens by design, with a clear sequence, measurable preconditions, and outcomes the practice can report, or by drift, improvising one AI tool at a time without a coherent model for how the pieces connect.

The profession has been here before. It has always found its way forward. The architects who helped it do that were the ones who looked at the next step and moved.

Closing Note

NovoCircle works at the intersection of enterprise architecture and intelligent automation. Our advisory practice helps organizations understand where they are in the arc described in this paper and what the next stage requires.

This paper is intended as an independent contribution to the practice of enterprise architecture. If it provokes useful thinking about where your practice is and where it is going, it has done its job.

Ryan Schmierer can be reached at ryan@novocircle.com. NovoCircle can be found at novocircle.com.

About the Author

Ryan Schmierer is the Sr. Managing Partner of NovoCircle, a technology advisory practice specializing in Modern Enterprise Architecture and Intelligent Automation. He brings more than 25 years of professional experience, including enterprise architect roles at Cisco and Microsoft, and has spent the past decade running an EA tools and consulting practice that has worked with architecture teams ranging from two people to five hundred, across a wide range of industries.

His focus has always been business performance, not technology for its own sake. The tools are means. What matters is the work done with them and the impact that work has on how an organization operates and competes. That conviction shapes everything NovoCircle does.

Methodology Note

This paper draws on four sources of evidence, weighted equally. First, the established formal EA maturity literature: TOGAF's Architecture Capability Maturity Model, the GAO Enterprise Architecture Management Maturity Framework (v2.0), LeanIX's maturity model derived from work with more than 1,100 organizations, CMMI-based frameworks, and the SAP Enterprise Architecture Framework.

Second, practitioner and analyst commentary from Architecture & Governance Magazine, EDUCAUSE, Info-Tech Research Group, and independent practitioner writing. Third, recent AI-readiness literature from Bizzdesign, Digital Science, and others working at the intersection of EA and AI augmentation. Fourth, the author's direct practitioner experience across more than 25 years of enterprise architecture work.

Where this paper departs from existing frameworks, it does so explicitly and for stated reasons. The distinction between organizational governance and repository semantic quality within Stage 5, the configuration-cost argument about EA tool categories and AI's effect on it, and the lossy compression framing of the Stage 6 transition are the author's original contributions based on practitioner observation. They are offered as positions to be tested against the reader's own experience, not as conclusions drawn from a formal research methodology.

Published March 2026. NovoCircle. novocircle.com

Ryan Schmierer is the Sr. Managing Partner of NovoCircle.