DRAFT — UNOFFICIAL — NOT FOR OPERATIONAL USE

ARCHITECTURE REFERENCE

# ODT-EA

USAREUR-AF

## Enterprise Architecture — 02: Schools of Thought

*Architecture Reference*

HEADQUARTERS
UNITED STATES ARMY EUROPE AND AFRICA
(USAREUR-AF)
Wiesbaden, Germany

DRAFT — NOT FOR OFFICIAL USE. FOR TRAINING PLANNING PURPOSES ONLY.

**20 MARCH 2026**

DRAFT — UNOFFICIAL — NOT FOR OPERATIONAL USE

sidebar_position: 3 title: "Schools of Thought"

# ENTERPRISE ARCHITECTURE — 02: SCHOOLS OF THOUGHT

### Why EA efforts talk past each other — and how to recognize which world you're in

## THE CORE PROBLEM

Walk into any EA conversation and you'll find smart people disagreeing on what EA is *for*. This isn't because some are wrong. It's because there are three fundamentally different answers to the question:

**"What problem does Enterprise Architecture solve?"**

Each answer is coherent. Each produces a different EA practice. Each produces different artifacts, different governance models, and different definitions of success. Most EA failures happen because leadership buys one answer and practitioners deliver another.

## THE THREE SCHOOLS

### School 1 — Enterprise IT Design

**Core belief:** EA's primary purpose is to plan and design information technology to meet business goals. Non-IT concerns are *inputs*, not *outputs*.

**What it looks like in practice:** - Application portfolio rationalization (what do we own, what's redundant, what do we retire) - Technology standards and reference architectures - Infrastructure roadmaps - Security architecture patterns - Integration and API standards

**What "done" looks like:** A coherent, documented, governed technology stack that maps to business capabilities.

**Where it works well:** Large organizations that need to consolidate, rationalize, or modernize their technology estate. Classic post-merger integration. Legacy modernization programs.

**Where it fails:** When the real problems are organizational, process-based, or data-driven — this school has no answer for those. It also tends to treat "the business" as a customer rather than a co-author of architecture.

**What leaders who want this say:**

"We need a target architecture." / "We need to rationalize our app portfolio." / "We need tech standards."

---

## School 2 — Enterprise Integrating

**Core belief:** EA's purpose is coherency across the *full enterprise* — strategy, people, process, data, and technology — including the link between strategy formulation and strategy execution.

**What it looks like in practice:** - Capability maps (what the enterprise does, regardless of how) - Value stream analysis (how work flows end-to-end) - Operating model design - Cross-domain interoperability rules - Portfolio governance connected to capability gaps - Formal views: business, data, application, technology (e.g., TOGAF, DODAF)

**What "done" looks like:** A set of governed views across all domains that inform investment decisions, shape requirements, and enable coherent transformation.

**Where it works well:** Complex, multi-domain organizations undergoing sustained transformation. Government agencies. Defense enterprises. Organizations where "IT and the business" is a false separation.

**Where it fails:** Heavy governance overhead. Requires sustained executive sponsorship. Can become too abstract or produce views nobody uses operationally. Vulnerable to "framework theater."

**What leaders who want this say:**

"We need to align IT to mission." / "We need cross-domain coherence." / "We need a common operating model."

---

## School 3 — Enterprise Ecosystem Adaptation

**Core belief:** EA's purpose is to build organizational capacity to *sense, learn, and adapt* — to create resilience and sustainability in a constantly changing environment.

**What it looks like in practice:** - Feedback loops and sensing mechanisms - Modular, composable architecture patterns (avoid lock-in) - Evolutionary design over big-bang planning - Capability-based planning that adapts to threat/opportunity changes - Architecture as a learning function, not a planning function

**What "done" looks like:** An organization that can absorb change without catastrophic disruption. Architecture decisions that explicitly preserve optionality.

**Where it works well:** Adversarial, fast-changing environments where planning horizons are short and the future is genuinely uncertain. Startups. Military/national security contexts. Digital businesses.

**Where it fails:** Hard to sell to leadership because it has no clear deliverable. Hard to govern because it resists formalization. Can justify *not* making decisions ("we're staying adaptive").

**What leaders who want this say:**

"We need to be more agile." / "We can't afford to be locked in." / "We need resilience."

## THE MATRIX VIEW

| Dimension | IT Design | Integrating | Ecosystem Adaptation |
|---|---|---|---|
| **Primary concern** | Technology coherence | Cross-domain coherence | Organizational resilience |
| **Main artifact** | Tech standards + app portfolio | Capability/value maps + views | Feedback mechanisms + modular patterns |
| **Governance model** | Standards board | Architecture review board | Portfolio + sensing functions |
| **Planning horizon** | 3–5 years | 3–5 years | Rolling / continuous |
| **Success metric** | Rationalization, cost reduction | Strategy-execution alignment | Adaptability, optionality |
| **EA sits with** | CIO/CTO | CTO/COO/CEO | CTO/Strategy |
| **Risk if wrong** | Over-engineered IT | Framework theater | Analysis paralysis |

## WHAT THIS MEANS FOR A DEFENSE/MILITARY CONTEXT

A command CTO environment almost certainly requires **School 2 (Enterprise Integrating)** as the primary stance, with **School 3 (Ecosystem Adaptation)** as an explicit design principle — because:

- Missions change faster than acquisition cycles
- Cross-domain coherence (people, process, data, systems) is a literal operational requirement
- Adversarial environments demand resilience, not just optimization
- DODAF/NATO architectures are explicitly multi-view, multi-domain (School 2 structure)
- The data layer (ontologies, semantic models, authoritative sources) only makes sense as an enterprise integration problem, not an IT design problem

School 1 is not irrelevant — you still need tech standards and platform governance — but it should be *downstream* of Schools 2 and 3, not the organizing principle.

---

## DIAGNOSING WHICH SCHOOL YOUR ORGANIZATION IS IN

Ask these questions and listen for the answers:

| Question | School 1 Answer | School 2 Answer | School 3 Answer |
|---|---|---|---|
| "What does EA deliver?" | Standards, app catalogs | Capability maps, roadmaps | Feedback loops, composable patterns |
| "Who is EA's customer?" | CIO / IT leadership | CTO + business leadership | Everyone / the enterprise itself |
| "What's the biggest EA failure mode?" | Standards not enforced | Views not used in decisions | Too rigid, can't adapt |
| "What does a good EA look like?" | Clean, consistent tech stack | Aligned strategy ↔ execution | Resilient, modular enterprise |

---

## THE HIDDEN FOURTH PROBLEM: MIXED SCHOOLS

Many EA efforts fail not because they chose the wrong school, but because **leadership assumes one school and practitioners deliver another**. Common patterns:

- Leadership wants School 2 outcomes (strategy-execution alignment) but funds School 1 activities (app rationalization)
- Practitioners deliver School 2 artifacts (beautiful capability maps) with no School 1 governance (nobody enforces standards) and no School 3 design (everything is tightly coupled)
- The EA team speaks in School 3 language ("we need to stay adaptive") as a way to avoid the hard work of School 2 ("we need to define and enforce a capability model")

**Fix:** Make the school choice explicit. State it. Document it. Get leadership agreement on what EA is *for* in your organization.

---

*Previous: 01 — Foundation | Next: 03 — Artifacts and Views*