

Enterprise Architecture & the Situational Picture at the Solution Level

What portfolio and solution leaders can learn from the discipline of enterprise architecture

Version 2.2 · As of 2026-08-18 · Audience: portfolio and solution leaders – no prior EA knowledge required · Companion document to the situational-awareness reading list

About this document: Curated, conceived and editorially owned by **Robert Seebauer**; researched and produced with the support of **Claude** (Anthropic – first edition with Claude Opus 4.8, v1.1 restyling, v2.0 deepening and v2.1 reference maintenance with Claude Fable 5, via Claude Code). This is **version 2.2** (deepened edition: Part A2 with four additional thought leaders – Hohpe, Fowler, Robert C. Martin, Hohmann). Sister document in the series: "Team of Teams & Staff Training". Earlier editions: [release archive](#) [denkschriften/<date>](#) .

Transparency notice: This memorandum was produced with the support of an AI system (Claude, Anthropic – models Claude Opus 4.8 and Claude Fable 5, via Claude Code): drafting, research preparation, typesetting and chart code. All content was substantively reviewed, edited and approved by Robert Seebauer, who holds editorial responsibility.

Executive Summary

Guiding question: Enterprise architecture (EA) is tightly coupled to the decisions made at portfolio and solution level. What do the people at the solution level need to *know* and be able to *do* – and what follows from that for producing a situational picture?

THE SINGLE MOST IMPORTANT SENTENCE IN THIS DOCUMENT

The solution level is not a "coordination layer" but the **architectural responsibility level**: it keeps the control loop between strategy (EA) and delivery (the ARTs) closed – and the **situational picture is the instrument** with which it does so.

Findings in five sentences:

- 1 EA and the solution level form a control loop, not a hierarchy.** EA provides the frame downwards (operating model, capability map, reference architectures, guardrails); the solution level reports the architectural reality back upwards (runway, debt, integration risks). Tom Graves frames EA as a "*bridge between strategy and change-management*" – it is about "*effectiveness of outcomes*", not about tending diagrams.
- 2 The language of this control loop is capabilities** – the dominant concept in all four EA works examined. Whoever thinks in features instead of business capabilities at the solution level cannot anchor progress to strategy.
- 3 The most critical single skill is translation:** Michael Carducci calls the architect the "*Organizational Rosetta Stone*" – communication failure is a leading cause of product failure. A situational picture that speaks only one language (business only, or engineering only) gets misread or not read at all.
- 4 A situational picture is only as good as the competence of those who produce and read it.** The necessary skills (capability thinking, NFR/runway stewardship, dependency coordination, trade-off discipline, systems thinking, metric literacy) decide whether the report measures *effectiveness* or mere *output*.

- 5 **Consequence for our software:** Besides the flow metrics (already in place), the solution situational picture needs four EA dimensions – **capability health**, **NFR/runway**, **risks (ROAM)**, **dependencies** – plus a **confidence flag per data source** and a **two-readings layout**. These dimensions currently have no source in the Jira data stream; they need their own lightweight input artefacts ([roadmap](#)).

Part A – The sources: four EA works in profile

All four are available to the curator as full texts. Selection: the EA/architecture-discipline titles of the 22-volume Apress corpus.

A.1 Tom Graves & Slade Beard – *Doing Enterprise-Architecture* (2025)

The work: At ~194,000 words the most extensive of the corpus. Graves – a long-standing thought leader of the "enterprise-as-a-whole" school – understands EA not as an IT discipline but as the **change capability of the whole organisation**.

Core ideas:

- EA is "*about effectiveness of outcomes, for the organization, and for the enterprise as a whole*" and "*a bridge between strategy and change-management*". Effectiveness is the guiding measure – not completeness of models.
- **Layers and patterns of change:** every change moves from an as-is to a to-be across several layers (business → application → technology) – optimising only one layer merely shifts the problem.
- Term frequency in the full text underpins the orientation: *effectiveness* (250×), *governance* (198×), *capability/-ies* (209×), *stakeholder(s)* (117×) – EA as a steering discipline, not a drawing discipline.

For us: The solution situational picture is exactly this bridge in report form – it translates strategy (the EA frame) into delivery reality and back.

A.2 Michael Carducci – *Mastering Software Architecture* (2025)

The work: A book about the **architect's role** and a "tailor-made" architecture model – against blueprint dogma, for context-specific decision-making.

Core ideas:

- **Capabilities as the "Language of the Architect"** (a chapter of its own): architecture is described through capabilities, not through products or frameworks.
- The architect as the "*Organizational Rosetta Stone*" (after Nate Schutta): translating between business language, marketing, vision and engineering language. "*Communication failures often account for a significant portion of product failures.*"
- "*Architecture Must Solve the 'Right' Problem*" – illustrated by the Laserdisc example (technically brilliant, wrong problem). **Breadth of knowledge** ("breadth is the superpower") before depth; trade-offs (111×) as a permanent condition.

For us: The two readings of the situational picture (executive ↔ engineering) are not cosmetics but the Rosetta-Stone requirement in report form.

A.3 Sean (Chunhong) Gu – *Mastering Enterprise Solution Modeling* (2024)

The work: Introduces **A-ESA** (Agile Enterprise Solution Architecture) – a method that brings EA thinking to the **solution level** in an agile, lightweight way.

Core ideas:

- **Eight architecture profiles** from strategic, lean-business, reference, integration and application architecture through microservice up to cloud/big-data architecture – a tiered modelling grid instead of one mega-framework.
- Very strong treatment of **NFRs** (*non-functional* 51×) and **governance** (264× – top score in the corpus): solution modelling without NFR/governance reference is incomplete.
- "Lean model mode": as much model as necessary, as little as possible – the agile answer to classical EA heavyweights.

For us: The most direct methodological template – A-ESA shows how EA substance (profiles, NFRs, governance) can be lifted to solution level in an agile-compatible way without collapsing into TOGAF weight.

A.4 Andre Milchman – *Unit Oriented Enterprise Architecture* (2024)

The work: EA for **large sociotechnical systems (LSS)** in the age of AI – organisations as simultaneously physical, digital and social systems.

Core ideas:

- The diagnosis: stakeholder complexity can paralyse ("*The Grand Ballet of Stakeholders*"), nobody knows the whole picture ("*Who Knows What It Looks Like?*"), information flows worse than claimed ("*The Information Masquerade*"), command-and-control secretly lives on.
- The answer: a **unit-oriented** architecture paradigm – composing the enterprise from clearly cut, accountable units; explicitly **systems thinking** (21×) and *decision-making* (46×) as core competencies.
- As the only work of the corpus it literally uses "**situational awareness**" – the bridge to our situational-picture concept and to the Team-of-Teams memorandum.

For us: Confirms the solution level as a sociotechnical unit (not just a technology slice) – and that a shared situational picture ("shared consciousness" in McChrystal's terms) is the antidote to the "information masquerade".

A.5 Supporting sources (compact)

- **Ross/Weill/Robertson – *Enterprise Architecture as Strategy*** (already on the reading list): operating model & core diagrams – the classical EA frame that Graves/Milchman modernise.
- **The technical corpus** (microservices/distributed systems): supplies the *realities* on which solution decisions founder when the knowledge is missing – CAP/eventual consistency/idempotence (*Immutable Architecture*), saga/CQRS/circuit breaker (*Practical Patterns*), SLO/error budget (*SRE with Azure*), coupling→defect proneness (*Fault Detection*).
- **SAFe Enterprise Solution Delivery / Leffingwell** (reading list): solution intent, capabilities, architecture runway as the framework vocabulary of the solution level.

Part A2 – Four additional thought leaders (new in v2.0)

Outside the Apress corpus, but formative for the discipline's self-image – four practice-oriented thought leaders whose works independently support the findings from Part A and each extend one flank: **Hohpe** the translation and integration flank, **Fowler** the evolution flank, **Martin** the structure and quality flank, **Hohmann** the business-model flank.

A2.1 Gregor Hohpe – The architect in the elevator (architectelevators.com)

The works: *Enterprise Integration Patterns* (with Bobby Woolf, 2003) · *The Software Architect Elevator* (2020) · *Cloud Strategy* (2020) · *Platform Strategy – Innovation Through Harmonization* (2024).

- The **Architect Elevator**: value is created not on one floor but by the *ride* between the penthouse (board/strategy) and the engine room (engineering). The larger the organisation, the more floors – and the more expensive it becomes when nobody rides: strategy and delivery drift apart unnoticed.
- "Architects live in the first derivative": what counts is not the state but the rate of change. Architecture work means **creating and selling options** under uncertainty (real-options thinking).
- **Enterprise Integration Patterns**: 65 messaging patterns as a shared language for asynchronous integration – still *the* catalogue (the vocabulary behind Kafka, ESB and iPaaS architectures). Integration is the core reality of every multi-ART solution.
- **Cloud Strategy / Platform Strategy**: migration and internal platforms are strategy questions, not technology questions; platforms create "innovation through harmonization", shift the build/buy line and operationalise the architecture runway.

For us: Hohpe delivers the operating manual for the Rosetta-Stone role (finding 3) – the elevator is the image behind the two-readings layout. And "first derivative" is the reason the situational picture must show **trends and rates of change**, not just snapshots.

A2.2 Martin Fowler – Evolution instead of master plan (martinfowler.com)

The works: *Patterns of Enterprise Application Architecture* (2002) · *Refactoring – Improving the Design of Existing Code* (2nd ed. 2018) · martinfowler.com/bliki as a continuously maintained reference.

- **PoEAA**: named patterns (Domain Model, Data Mapper, Unit of Work ...) create a shared vocabulary that makes newly assembled teams immediately effective – the civilian counterpart to the doctrine thesis of the Team-of-Teams memorandum ("everyone learns the same language").
- **Refactoring & the design-stamina hypothesis**: internal quality is not an end in itself – neglected design measurably throttles feature speed. The **technical-debt quadrant** (deliberate/inadvertent × prudent/reckless) makes debt discussable and steerable instead of moral.
- **Strangler Fig**: modernisation as gradually growing around the legacy system instead of a big-bang rewrite – the reference pattern for as-is→to-be transitions.

For us: the economic justification of principle 2 (*debt is a first-class citizen*) – and with Strangler Fig the standard pattern behind "modernisation roadmaps steerable as epics" (B.2). The bliki is the living reference for the engineering reading of the situational picture.

A2.3 Robert C. Martin – Boundaries, rules, craftsmanship

The works: *Clean Code – A Handbook of Agile Software Craftsmanship* (2008) · *Clean Architecture – A Craftsman's Guide to Software Structure and Design* (2017).

- **Dependency Rule:** source-code dependencies point only inwards, towards the business logic; frameworks, database and UI are replaceable details. Good architecture **maximises the number of decisions not yet made** – keeping options open instead of cementing early.
- **Screaming Architecture:** a system's structure should "scream" its purpose, not its framework.
- **Craftsmanship ethos:** code is read many times more often than it is written; "*The only way to go fast is to go well*" – discipline is a prerequisite for speed, not a brake.
- **Framing:** deliberately normative in tone – valuable as shared doctrine, to be read with Carducci's "tailor-made" corrective (A.2) rather than as a blueprint.

For us: "boundaries" are the craft tools for clean solution and ART cuts; the ethos explains why NFR and quality dimensions belong in the situational picture (finding 4, principle 2) – whoever does not surface internal quality steers blindly into speed loss.

A2.4 Luke Hohmann – Architecture meets business model

The works: *Beyond Software Architecture – Creating and Sustaining Winning Solutions* (2003) · *Software Profit Streams™* (with Jason Tanner, 2023) · *Innovation Games* (2006) · participatory budgeting for portfolio prioritisation (via his company Conteneo; adopted by SAFe into Lean Portfolio Management).

- **Tarchitecture vs. marketecture:** the technical architecture and the market/business architecture (licensing, deployment model, upgrade paths, support, branding) must be designed *together* – otherwise the technology wins and the product loses.
- **Software Profit Streams:** pricing, packaging and licence compliance as a design discipline (Profit Stream Canvas) – sustainable profitability is designed, not hoped for.
- **Innovation Games / participatory budgeting:** serious games (Buy a Feature, Prune the Product Tree, Participatory Budgeting) as structured prioritisation and alignment rituals that bundle many voices into one decision.

For us: the business reading of the two-readings layout has its standard work here (findings 3 and 5): a solution situational picture that shows runway and debt but hides profit and cost streams per capability tells only half the story. Participatory budgeting is at the same time the concrete ritual for portfolio prioritisation – the bridge to the Team-of-Teams memorandum (rituals, planning games).

Part B – How the EA discipline works (and what carries over)

Analogous to Part B of the Team-of-Teams memorandum ("how militaries train staffs"), here: how the EA profession organises its craft – as a quarry for the solution level.

B.1 What enterprise architects actually do

Beyond framework folklore, EA work consists of four recurring activities:

- 1 Mapping**
Capability maps, as-is/to-be landscapes, dependency and information flows. (Graves: as-is/to-be across change layers; Milchman: unit composition.)
- 2 Setting the frame**
Reference architectures, principles, guardrails, NFR standards – *enabling* autonomy through clear guardrails (A-ESA governance; *Pro Azure Governance*: policy-as-code as its lived form).
- 3 Supporting decisions**
Making options and trade-offs explicit, documenting assumptions (Carducci: trade-offs as a permanent condition; ADR practice).
- 4 Feeding back**
Replaying architectural reality (debt, risks, runway) into strategy and investment decisions.

B.2 The tools in plain language

EA tool	What it delivers	Solution-level counterpart
Capability map	Shared business language; investments become addressable	Map solution capabilities to ARTs; report capability health
Reference architecture / profiles (A-ESA)	Reusable patterns per layer	Solution intent + architecture runway
NFR catalogue	Makes qualities measurable	NFR register with target/actual per solution
Principles & guardrails	Autonomy within guardrails	Policy-as-code, compliance evidence in the report
ADR / decision log	Decisions + assumptions traceable	Decision/assumption log in the situational picture
As-is/to-be across layers	Thinking change through completely	Modernisation roadmaps steerable as epics

B.3 What the EA profession can get wrong (lessons from the sources)

- Ivory tower:** models without delivery contact – Graves' antidote is the fixation on effectiveness; Milchman's diagnosis ("information masquerade") describes the failure mode.
- Framework dogma:** Carducci argues explicitly against blueprint copying ("tailor-made"); A-ESA against model heavyweights ("lean model mode").
- Governance as brake instead of enabler:** the modern line (A-ESA, Azure governance) turns governance into automated, serving guardrails.

Transfer: The solution level adopts the four activities from B.1 in a lighter form – map, frame, support decisions, feed back – and avoids the three failure modes. The situational picture is its feedback instrument.

Part C – Transfer: knowledge, skills, and the situational-picture consequence

Short version; full derivation with the role matrix in the curator's working archive.

C.1 The control loop as the organising frame

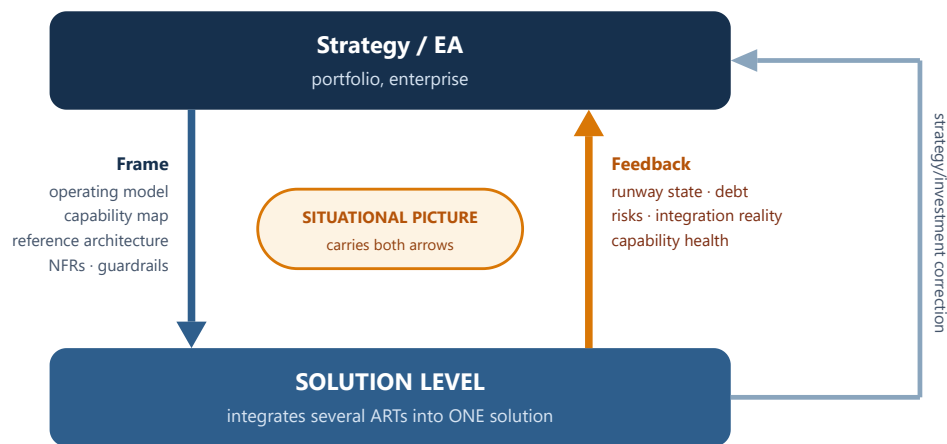


Fig. 1 – The control loop: EA provides the frame downwards, the solution level reports the architectural reality back – the situational picture carries both directions.

The situational picture carries both arrows: it makes the frame and the feedback visible – or the control loop exists only on slides.

C.2 Foundational knowledge (all participants)

Capability thinking & operating model · solution intent/NFRs/architecture runway · realities of distributed systems (CAP, eventual consistency, idempotence, saga/CQRS) · as-is/to-be across change layers · governance-as-enablement · measurement/metric foundations (flow + SLO + quality; GQM; Goodhart) · regulation where applicable.

C.3 Skills – necessary vs. helpful (the dividing criterion)

Necessary (without them the function breaks or the situational picture becomes misleading)

cross-ART dependency/integration coordination · NFR/runway stewardship · capability/value-stream modelling · trade-off decisions under uncertainty · translation business→architecture→engineering (Rosetta Stone) · risk facilitation (ROAM) · correct metric interpretation · systems thinking.

Helpful (amplifiers)

deep stack expertise · Wardley mapping · set-based design · red-teaming/premortems · storytelling with data · MLOps/AI awareness · calibrated estimation/forecasting (Hubbard, Monte Carlo).

The tiering is **role-relative** (STE ↔ solution architect ↔ solution management ↔ system team): see the matrix in the preparatory paper.

C.4 The five situational-picture principles (distilled)

1 Capability before feature

The report shows the health of business capabilities, not ticket counts – otherwise it measures output instead of effectiveness (Graves).

2 Debt is a first-class citizen

NFR fulfilment and runway state stand on equal footing with flow – invisible architectural debt is the most expensive risk.

3 Two readings, one picture

Executive view and engineering view from the same data (Rosetta Stone, Carducci) – never two separate truths.

4 Confidence belongs to the number

Every pooled figure carries provenance and degree of trust – against the "information masquerade" (Milchman).

5 The situational picture is a governance instrument

ROAM risks, decisions and assumptions are part of the report – it *supports* decisions instead of merely illustrating them (EA activities 3 and 4 from B.1).

Part D – Concrete programmes & implementation

D.1 Enabling the participants (buyable/learnable)

- **EA/architecture education:** context-driven rather than framework certificate collecting – reading path: Carducci (role) → Graves (enterprise view) → A-ESA (solution method) → Milchman (sociotechnical); flanked by Ross/Weill. *Deepening (v2.0, Part A2):* Hohpe (*Software Architect Elevator*) as the entry into the mediator role, Fowler (*Refactoring*, PoEAA) for the evolution and pattern craft, Martin (*Clean Architecture*) for boundaries and dependency rules, Hohmann (*Beyond Software Architecture, Software Profit Streams*) for the business-model flank. TOGAF courses at most as vocabulary training.
- **Decision & staff competence:** already anchored in the memoranda series – 12-month curriculum, red-teaming, calibration (Hubbard), superforecasting: see the Team-of-Teams memorandum, Part D. The EA skills dock on there as a module (recommendation: capability mapping + NFR workshop as a dedicated curriculum building block).
- **Technical foundations:** reading paths in the curator's working archive (e.g. "distributed data/communication": Immutable Architecture → Go & NATS; "operations/metrics bridge": Cloud-Native DevOps → SRE with Azure → Fault Detection).

D.2 Getting started for the impatient (90 days)

Weeks 1–2 Start mapping

Sketch the solution's capability map (workshop, 1 day) + create the NFR register (the ten most important).

Weeks 3–6 Structure risks & decisions

Establish the ROAM board; extend PI planning with a dependency register; start the decision log (lightweight ADR).

Weeks 7–12 Extend the situational picture

Extend the situational picture with the four EA dimensions (initially maintained manually, tooling follows); introduce the two-readings structure; first retrospective on the situational picture itself ("does it measure effectiveness?").

D.3 Software implementation

The technical roadmap for the `portfolio` module lives in `docs/portfolio_roadmap_solution-lagebild.md` – organised by data provenance: **Phase A** (computable from existing Jira data: confidence flag, extended summary), **Phase B** (new lightweight input artefacts: capability map, NFR/runway register, ROAM board, decision log, dependency register), **Phase C** (external integrations: SLO/error budget, DORA/quality, Jira REST).

THE ROADMAP'S CORE STATEMENT

The EA dimensions currently have **no source in the data stream** – they need their own inputs; **that is exactly where the value is created.**

Connectivity to the existing series (memoranda)

1

Team-of-Teams memorandum

There leadership/process (shared consciousness, empowered execution, staff training) – here structure/architecture (capabilities, runway, governance). Together they form the two halves of solution enablement: *how we lead* and *what we talk about*.

2

Situational-awareness reading list

This memorandum deepens its "Level 3 – Large Solution" (A–D) architecturally; the skills list there (C) is sharpened by the necessary/helpful tiering.

3

Agile-metrics series

The BGS principle (observed/estimated/system-generated) structures the software roadmap; the flow metrics remain the economic backbone, the EA dimensions add the structural one.

Bibliography

Primary (EA corpus; full texts held by the curator): Graves, Tom / Beard, Slade: *Doing Enterprise-Architecture*. Apress, 2025. · Carducci, Michael: *Mastering Software Architecture – A Comprehensive New Model and Approach*. Apress, 2025. · Gu, Sean (Chunhong): *Mastering Enterprise Solution Modeling – A Guide to Agile Enterprise Solution Architecture (A-ESA)*. Apress, 2024. · Milchman, Andre: *Unit Oriented Enterprise Architecture – Constructing Large Sociotechnical Systems in the Age of AI*. Apress, 2024.

Additional core sources (Part A2): Hohpe, Gregor / Woolf, Bobby: *Enterprise Integration Patterns – Designing, Building, and Deploying Messaging Solutions*. Addison-Wesley, 2003. – Hohpe: *The Software Architect Elevator – Redefining the Architect's Role in the Digital Enterprise*. O'Reilly, 2020. – Hohpe: *Cloud Strategy – A Decision-Based Approach to Successful Cloud Migration*. Architect Elevator Series, 2020. – Hohpe / Danieli / Landreau: *Platform Strategy – Innovation Through Harmonization*. Architect Elevator Series, 2024. – Web: architectelevator.com. · Fowler, Martin: *Patterns of Enterprise Application Architecture*. Addison-Wesley, 2002. – Fowler: *Refactoring – Improving the Design of Existing Code*, 2nd ed. Addison-Wesley, 2018. – Web: martinfowler.com (bliki). · Martin, Robert C.: *Clean Code – A Handbook of Agile Software Craftsmanship*. Prentice Hall, 2008. – Martin: *Clean Architecture – A Craftsman's Guide to Software Structure and Design*. Prentice Hall, 2017. · Hohmann, Luke: *Beyond Software Architecture – Creating and Sustaining Winning Solutions*. Addison-Wesley, 2003. – Tanner, Jason / Hohmann, Luke: *Software Profit Streams™ – A Guide to Designing a Sustainably Profitable Business*. Applied Frameworks, 2023. – Hohmann: *Innovation Games – Creating Breakthrough Products Through Collaborative Play*. Addison-Wesley, 2006.

Supporting (selection from corpus & reading list): Perry, Michael L.: *The Art of Immutable Architecture*, 2nd ed. Apress, 2024. – Christudas/Telang: *Practical Microservices Architectural Patterns*. Apress, 2025. – Huete Beloki, Unai: *The Art of SRE with Azure*. Apress, 2025. – Sharma/Syed: *Fault Detection in Microservice Architectures*. Apress, 2026. – Rahman, Rezwanur: *Pro Azure Governance and Security*.

Apress, 2025. – Ross/Weill/Robertson: *Enterprise Architecture as Strategy*. HBR Press, 2006. – Leffingwell/Knaster: *SAFe Distilled* / *SAFe "Enterprise Solution Delivery"*. – McChrystal et al.: *Team of Teams*, 2015 (→ dedicated memorandum in this series).

Internal references: Team-of-Teams memorandum "Team of Teams & Staff Training" · further working documents (reading list, preparatory paper, EPUB inventory/synthesis) in the curator's working archive · software roadmap: [docs/portfolio_roadmap_solution-lagebild.md](#).

Version history

Version	Date	Changes
1.0	2026-08-06	First edition: four EA sources in profile (Graves, Carducci, A-ESA, Milchman), EA craft & transfer, five situational-picture principles, 90-day entry path, interlocking with the Team-of-Teams series and the software roadmap.
1.1	2026-08-07	Visual redesign in the series style of the Team-of-Teams memorandum (cover, cards, numbered circles, control-loop figure instead of ASCII diagram); "single most important sentence" on page 1 as the new series standard; page-break/reading-flow rules (headings stay with their following text). Content unchanged from 1.0.
2.0	2026-08-07	Deepened edition: new Part A2 with four additional thought leaders – Hohpe (<i>Architect Elevator</i> , <i>Enterprise Integration Patterns</i> , <i>Cloud Strategy</i> , <i>Platform Strategy</i>), Fowler (<i>PoEAA</i> , <i>Refactoring</i> , Strangler Fig, martinfowler.com), Robert C. Martin (<i>Clean Code</i> , <i>Clean Architecture</i>), Hohmann (<i>Beyond Software Architecture</i> , <i>Software Profit Streams</i> , Innovation Games/participatory budgeting); D.1 reading path extended; reading-list supplement v3.
2.1	2026-08-14	Reference maintenance for the online publication: cross-references to the series now link to the online overview, workshop paths removed; cross-references no longer carry version numbers. Content unchanged.
2.2	2026-08-18	Transparency notice per the series standard (AI support, review, editorial responsibility) added to the header; series versioning principles 5/6. Content unchanged.

Source briefing "Enterprise Architecture & Solution Situational Picture" · Version 2.2 · Solution/Portfolio Management · As of 2026-08-18 · curated by Robert Seebauer, produced with Claude (Anthropic) · Current editions online: <https://jaegerfeld.github.io/situation-report/denkschriften/>