

Enterprise-Architektur & das Lagebild auf der Solution-Ebene

Was Portfolio- und Solution-Verantwortliche von der Disziplin der Unternehmensarchitektur lernen können

Version 2.2 · Stand 18.08.2026 · Zielgruppe: Portfolio- und Solution-Verantwortliche – EA-Vorwissen wird nicht vorausgesetzt
· Begleitdokument zur Lagebild-Literaturliste

Zu diesem Dokument: Kuratiert, inhaltlich durchdacht und verantwortet von **Robert Seebauer**; recherchiert und erstellt mit Unterstützung von **Claude** (Anthropic – Erstfassung Claude Opus 4.8, Ausstattung v1.1, Vertiefung v2.0 und Verweispflege v2.1 Claude Fable 5, via Claude Code). Dies ist **Version 2.2** (vertiefte Fassung: Teil A2 mit vier ergänzenden Vordenkern – Hohpe, Fowler, Robert C. Martin, Hohmann). Schwesterdokument der Reihe: „*Team of Teams & Stabsausbildung*“. Ältere Stände: [Release-Archiv](#) [denkschriften/<Datum>](#) .

Transparenzhinweis: Diese Schrift wurde mit Unterstützung eines KI-Systems (Claude, Anthropic – Modelle Claude Opus 4.8 und Claude Fable 5, via Claude Code) erstellt: Entwurf, Recherche-Aufbereitung, Satz und Grafik-Code. Alle Inhalte wurden von Robert Seebauer inhaltlich geprüft, redaktionell bearbeitet und freigegeben; er trägt die redaktionelle Verantwortung.

Executive Summary

Ausgangsfrage: Enterprise-Architektur (EA) hängt eng mit den Entscheidungen auf Portfolio- und Solution-Ebene zusammen. Was müssen die Beteiligten auf der Solution-Ebene *wissen* und *können* – und was folgt daraus für die Erzeugung eines Lagebilds?

WICHTIGSTER EINZELSATZ DES DOKUMENTS

Die Solution-Ebene ist keine „Koordinationsschicht“, sondern die **architektonische Verantwortungsebene**: Sie hält den Regelkreis zwischen Strategie (EA) und Umsetzung (ARTs) geschlossen – und das **Lagebild ist das Instrument**, mit dem sie das tut.

Befund in fünf Sätzen:

- 1 **EA und Solution-Ebene bilden einen Regelkreis, keine Hierarchie.** EA gibt nach unten Rahmen (Operating Model, Capability-Landkarte, Referenzarchitekturen, Guardrails); die Solution-Ebene meldet nach oben die architektonische Realität (Runway, Schulden, Integrationsrisiken) zurück. Tom Graves fasst EA als „*bridge between strategy and change-management*“ – es geht um „*effectiveness of outcomes*“, nicht um Diagrammpflege.
- 2 **Die Sprache dieses Regelkreises sind Capabilities** – in allen vier untersuchten EA-Werken das dominierende Konzept. Wer auf Solution-Ebene in Features statt in Geschäftsfähigkeiten denkt, kann Fortschritt nicht an Strategie andocken.
- 3 **Der kritischste Einzel-Skill ist Übersetzung:** Michael Carducci nennt den Architekten den „*Organizational Rosetta Stone*“ – Kommunikationsversagen ist eine Hauptursache von Produktversagen. Ein Lagebild, das nur eine Sprache spricht (nur Business oder nur Technik), wird fehlgelesen oder gar nicht gelesen.
- 4 **Ein Lagebild ist nur so gut wie die Kompetenz seiner Ersteller und Leser.** Notwendige Skills (Capability-Denken, NFR-/Runway-Stewardship, Abhängigkeitskoordination, Trade-off-Disziplin, Systemdenken, Metrik-Interpretation) entscheiden, ob der Report *Wirksamkeit* oder bloß *Output* misst.

- 5 **Konsequenz für unsere Software:** Das Solution-Lagebild braucht neben den Flow-Metriken (vorhanden) vier EA-Dimensionen – **Capability-Health, NFR/Runway, Risiken (ROAM), Abhängigkeiten** – plus ein **Konfidenz-Flag je Datenquelle** und ein **Zwei-Lesarten-Layout**. Diese Dimensionen haben heute keine Quelle im Jira-Datenstrom; sie brauchen eigene, leichtgewichtige Eingabe-Artefakte ([Roadmap](#)).

Teil A – Die Quellen: vier EA-Werke im Profil

Alle vier liegen dem Kurator als Volltexte vor. Auswahl: die EA-/Architektur-Disziplin-Titel des 22-bändigen Apress-Korpus.

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

Das Werk: Mit ~194.000 Wörtern das umfangreichste des Korpus. Graves – langjähriger Vordenker der „Enterprise-as-a-whole“-Schule – versteht EA nicht als IT-Disziplin, sondern als **Change-Capability des gesamten Unternehmens**.

Kernideen:

- EA ist „*about effectiveness of outcomes, for the organization, and for the enterprise as a whole*“ und „*a bridge between strategy and change-management*“. Wirksamkeit (Effectiveness) ist das Leitmaß – nicht Vollständigkeit der Modelle.
- **Ebenen & Muster des Wandels** („Layers of Change“): Jede Veränderung bewegt sich von einem Ist (as-is) zu einem Soll (to-be) über mehrere Schichten (Business → Applikation → Technik) – wer nur eine Schicht optimiert, verschiebt Probleme.
- Begriffshäufigkeit im Volltext untermauert die Ausrichtung: *effectiveness* (250×), *governance* (198×), *capability/-ies* (209×), *stakeholder(s)* (117×) – EA als Steuerungs-, nicht als Zeichendisziplin.

Für uns: Das Solution-Lagebild ist genau diese Brücke in Berichtsform – es übersetzt Strategie (EA-Rahmen) in Umsetzungsrealität und zurück.

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

Das Werk: Ein Buch über die **Architektenrolle** und ein „Tailor-Made“-Architekturmodell – gegen Blaupausen-Dogma, für kontextspezifische Entscheidung.

Kernideen:

- **Capabilities als „Language of the Architect“** (eigenes Kapitel): Architektur wird über Fähigkeiten beschrieben, nicht über Produkte oder Frameworks.
- Der Architekt als „*Organizational Rosetta Stone*“ (nach Nate Schutta): Er übersetzt zwischen Business-Sprache, Marketing, Vision und Entwicklungssprache. „*Communication failures often account for a significant portion of product failures.*“
- „*Architecture Must Solve the 'Right' Problem*“ – illustriert am Laserdisc-Beispiel (technisch brilliant, falsches Problem). **Breadth of Knowledge** („die Breite ist die Superpower“) vor Tiefe; Trade-offs (111×) als Dauerzustand.

Für uns: Die zwei Lesarten des Lagebilds (Executive ↔ Engineering) sind keine Kosmetik, sondern die Rosetta-Stone-Anforderung in Berichtsform.

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

Das Werk: Führt **A-ESA** ein (Agile Enterprise Solution Architecture) – eine Methode, die EA-Denken agil und leichtgewichtig auf die **Solution-Ebene** bringt.

Kernideen:

- **Acht Architektur-Profil** von Strategie über Lean-Business-, Referenz-, Integrations-, Applikations-, Microservice- bis Cloud-/Big-Data-Architektur – ein gestuftes Modellierungsraster statt eines Mega-Frameworks.
- Sehr starke **NFR-Behandlung** (*non-functional* 51×) und **Governance** (264× – Spitzenwert im Korpus): Solution-Modellierung ohne NFR-/Governance-Bezug ist unvollständig.
- „Lean Model Mode“: so viel Modell wie nötig, so wenig wie möglich – die agile Antwort auf klassische EA-Schwergewichte.

Für uns: Das direkteste methodische Vorbild – A-ESA zeigt, wie EA-Substanz (Profile, NFRs, Governance) agil-kompatibel auf Solution-Ebene gehoben wird, ohne in TOGAF-Schwere zu verfallen.

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

Das Werk: EA für **große soziotechnische Systeme (LSS)** im KI-Zeitalter – Unternehmen als gleichzeitig physische, digitale und soziale Systeme.

Kernideen:

- Die Diagnose: Stakeholder-Komplexität kann lähmen („*The Grand Ballet of Stakeholders*“), niemand kennt das Gesamtbild („*Who Knows What It Looks Like?*“), Information fließt schlechter als behauptet („*The Information Masquerade*“), Command-and-Control lebt heimlich weiter.
- Antwort: ein **einheitenorientiertes** Architekturparadigma – das Unternehmen aus klar geschnittenen, verantwortlichen Einheiten komponieren; explizit **Systems Thinking** (21×) und *decision-making* (46×) als Kernkompetenz.
- Als einziges Werk des Korpus verwendet es wörtlich **„situational awareness“** – die Brücke zu unserem Lagebild-Begriff und zur Team-of-Teams-Denkschrift.

Für uns: Bestätigt die Solution-Ebene als soziotechnische Einheit (nicht nur Technik-Scheibe) – und dass ein geteiltes Lagebild („*Shared Consciousness*“ bei McChrystal) das Gegenmittel gegen die „*Information Masquerade*“ ist.

A.5 Ergänzende Stützquellen (kompakt)

- **Ross/Weill/Robertson – *Enterprise Architecture as Strategy*** (bereits in der Lagebild-Liste): Operating Model & Core Diagrams – der klassische EA-Rahmen, den Graves/Milchman modernisieren.
- **Der technische Korpus** (Microservices/verteilte Systeme): liefert die *Realitäten*, an denen Solution-Entscheidungen scheitern, wenn das Wissen fehlt – CAP/Eventual Consistency/Idempotenz (*Immutable Architecture*), Saga/CQRS/Circuit Breaker (*Practical Patterns*), SLO/Error Budget (*SRE with Azure*), Coupling→Fehlerneigung (*Fault Detection*).
- **SAFe Enterprise Solution Delivery / Leffingwell** (Lagebild-Liste): Solution Intent, Capabilities, Architecture Runway als Framework-Vokabular der Solution-Ebene.

Teil A2 – Vier ergänzende Vordenker (neu in v2.0)

Außerhalb des Apress-Korpus, aber prägend für das Selbstverständnis der Disziplin – vier Praxis-Vordenker, deren Werke die Befunde aus Teil A unabhängig stützen und um je eine Flanke erweitern: **Hohpe** die Übersetzungs- und Integrationsflanke, **Fowler** die Evolutionsflanke, **Martin** die Struktur- und Qualitätsflanke, **Hohmann** die Geschäftsmodell-Flanke.

A2.1 Gregor Hohpe – Der Architekt im Aufzug (architectelevator.com)

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

- Der **Architect Elevator**: Wert entsteht nicht in einem Stockwerk, sondern durch die *Fahrt* zwischen Penthouse (Vorstand/Strategie) und Maschinenraum (Engineering). Je größer die Organisation, desto mehr Stockwerke – und desto teurer, wenn niemand fährt: Strategie und Umsetzung driften unbemerkt auseinander.
- „*Architects live in the first derivative*“: Nicht der Zustand zählt, sondern die Änderungsrate. Architekturarbeit heißt, unter Unsicherheit **Optionen zu schaffen und zu verkaufen** (Real-Options-Denken).
- **Enterprise Integration Patterns**: 65 Messaging-Muster als gemeinsame Sprache für asynchrone Integration – bis heute der Katalog (das Vokabular hinter Kafka-, ESB- und iPaaS-Architekturen). Integration ist die Kernrealität jeder Mehr-ART-Solution.
- **Cloud Strategy / Platform Strategy**: Migration und interne Plattformen sind Strategie-, nicht Technikfragen; Plattformen erzeugen „Innovation durch Harmonisierung“, verschieben die Build/Buy-Linie und operationalisieren den Architecture Runway.

Für uns: Hohpe liefert die Betriebsanleitung für die Rosetta-Stone-Rolle (Befund 3) – der Aufzug ist das Bild hinter dem Zwei-Lesarten-Layout. Und „first derivative“ begründet, warum das Lagebild **Trends und Änderungsraten** zeigen muss, nicht nur Momentaufnahmen.

A2.2 Martin Fowler – Evolution statt Masterplan (martinfowler.com)

Die Werke: *Patterns of Enterprise Application Architecture* (2002) · *Refactoring – Improving the Design of Existing Code* (2. Aufl. 2018) · martinfowler.com/bliki als laufend gepflegtes Referenzwerk.

- **PoEAA**: Muster mit Namen (Domain Model, Data Mapper, Unit of Work ...) schaffen ein gemeinsames Vokabular, das zusammengewürfelte Teams sofort arbeitsfähig macht – das zivile Pendant zur Doktrin-These der Team-of-Teams-Denkschrift („alle lernen dieselbe Sprache“).
- **Refactoring & Design-Stamina-Hypothese**: Innere Qualität ist kein Selbstzweck – vernachlässigtes Design drosselt messbar die Featuregeschwindigkeit. Der **Technische-Schulden-Quadrant** (bewusst/unbewusst × klug/leichtsinnig) macht Schulden diskutier- und steuerbar statt moralisch.
- **Strangler Fig**: Modernisierung als schrittweises Umwachsen des Altsystems statt Big-Bang-Rewrite – das Referenzmuster für As-is→To-be-Übergänge.

Für uns: Die ökonomische Begründung von Prinzip 2 (*Schulden sind Erstklass-Bürger*) – und mit Strangler Fig das Standardmuster hinter „Modernisierungs-Roadmaps als Epics steuerbar“ (B.2). Der bliki ist die lebende Referenz für die Engineering-Lesart des Lagebilds.

A2.3 Robert C. Martin – Grenzen, Regeln, Handwerksethos

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

- **Dependency Rule:** Quellcode-Abhängigkeiten zeigen nur nach innen, zur Geschäftslogik; Frameworks, Datenbank und UI sind austauschbare Details. Gute Architektur **maximiert die Zahl der noch nicht getroffenen Entscheidungen** – Optionen offen halten statt früh zementieren.
- **Screaming Architecture:** Die Struktur eines Systems soll seinen Zweck „herausschreien“, nicht sein Framework.
- **Craftsmanship-Ethos:** Code wird um ein Vielfaches öfter gelesen als geschrieben; „*The only way to go fast is to go well*“ – Disziplin ist Geschwindigkeitsvoraussetzung, nicht Bremse.
- *Einordnung:* bewusst normativ formuliert – als gemeinsame Doktrin wertvoll, als Blaupause mit Carduccis „Tailor-Made“-Korrektiv (A.2) zu lesen.

Für uns: „Boundaries“ sind das Handwerkszeug für saubere Solution- und ART-Schnitte; das Ethos begründet, warum NFR- und Qualitätsdimensionen ins Lagebild gehören (Befund 4, Prinzip 2) – wer innere Qualität nicht ausweist, steuert blind in den Geschwindigkeitsverlust.

A2.4 Luke Hohmann – Architektur trifft Geschäftsmodell

Die Werke: *Beyond Software Architecture – Creating and Sustaining Winning Solutions* (2003) · *Software Profit Streams™* (mit Jason Tanner, 2023) · *Innovation Games* (2006) · Participatory Budgeting für Portfolio-Priorisierung (über seine Firma Conteneo; von SAFe ins Lean Portfolio Management übernommen).

- **Tarchitecture vs. Marketecture:** Die technische Architektur und die Markt-/Geschäftsarchitektur (Lizenzierung, Deployment-Modell, Upgrade-Pfade, Support, Branding) müssen *zusammen* entworfen werden – sonst gewinnt die Technik und verliert das Produkt.
- **Software Profit Streams:** Pricing, Packaging und Lizenz-Compliance als Designdisziplin (Profit Stream Canvas) – nachhaltige Profitabilität wird entworfen, nicht erhofft.
- **Innovation Games / Participatory Budgeting:** ernsthafte Spiele (Buy a Feature, Prune the Product Tree, Participatory Budgeting) als strukturierte Priorisierungs- und Alignment-Rituale, die viele Stimmen in eine Entscheidung bündeln.

Für uns: Die Business-Lesart des Zwei-Lesarten-Layouts hat hier ihr Standardwerk (Befund 3 und 5): Ein Solution-Lagebild, das Runway und Schulden zeigt, aber Profit-/Kostenströme je Capability ausblendet, erzählt nur die halbe Geschichte. Participatory Budgeting ist zugleich das konkrete Ritual für Portfolio-Priorisierung – die Brücke zur Team-of-Teams-Denkschrift (Rituale, Planspiele).

Teil B – Wie die EA-Disziplin arbeitet (und was davon trägt)

Analog zu Teil B der Team-of-Teams-Denkschrift („wie Militärs Stäbe ausbilden“) hier: wie die EA-Profession ihr Handwerk organisiert – als Steinbruch für die Solution-Ebene.

B.1 Was Enterprise-Architekten eigentlich tun

Jenseits der Framework-Folklore besteht EA-Arbeit aus vier wiederkehrenden Tätigkeiten:

- 1 Kartieren**
Capability-Landkarten, Ist-/Soll-Landschaften, Abhängigkeits- und Informationsflüsse. (Graves: as-is/to-be über Change-Layers; Milchman: Einheiten-Komposition.)
- 2 Rahmen setzen**
Referenzarchitekturen, Prinzipien, Guardrails, NFR-Standards – Autonomie *ermöglichen* durch klare Leitplanken (A-ESA-Governance; *Pro Azure Governance*: Policy-as-Code als gelebte Form).
- 3 Entscheidungen stützen**
Optionen und Trade-offs explizit machen, Annahmen dokumentieren (Carducci: Trade-offs als Dauerzustand; ADR-Praxis).
- 4 Rückkoppeln**
Architektur-Realität (Schulden, Risiken, Runway) in Strategie- und Investitionsentscheidungen zurückspielen.

B.2 Die Werkzeuge im Klartext

EA-Werkzeug	Was es leistet	Solution-Ebenen-Gegenstück
Capability-Map	Gemeinsame Business-Sprache; Investitionen adressierbar	Solution-Capabilities → ARTs mappen; Capability-Health berichten
Referenzarchitektur / Profile (A-ESA)	Wiederverwendbare Muster je Ebene	Solution Intent + Architecture Runway
NFR-Katalog	Qualitäten messbar machen	NFR-Register mit Ziel/Ist je Solution
Prinzipien & Guardrails	Autonomie in Leitplanken	Policy-as-Code, Compliance-Nachweis im Report
ADR / Decision Log	Entscheidungen + Annahmen nachvollziehbar	Decision-/Assumption-Log im Lagebild
As-is/To-be über Ebenen	Wandel vollständig denken	Modernisierungs-Roadmaps als Epics steuerbar

B.3 Was die EA-Profession *falsch* machen kann (Lehren aus den Quellen)

- **Elfenbeinturm:** Modelle ohne Umsetzungskontakt – Graves' Gegenmittel ist die Effectiveness-Fixierung; Milchmans Diagnose („Information Masquerade“) beschreibt das Scheitern.
- **Framework-Dogma:** Carducci argumentiert explizit gegen Blaupausen-Kopieren („Tailor-Made“); A-ESA gegen Modell-Schwergewichte („Lean Model Mode“).
- **Governance als Bremse statt Befähigung:** Die moderne Linie (A-ESA, Azure-Governance) dreht Governance in automatisierte, dienende Leitplanken.

Übertragung: Die Solution-Ebene übernimmt die vier Tätigkeiten aus B.1 in *leichter* Form – kartieren, rahmen, Entscheidungen stützen, rückkoppeln – und meidet die drei Fehlformen. Das Lagebild ist ihr Rückkopplungsinstrument.

Teil C – Übertragung: Wissen, Skills und die Lagebild-Konsequenz

Kurzfassung; vollständige Herleitung mit Rollen-Matrix im Werkstatt-Archiv des Kurators.

C.1 Der Regelkreis als Ordnungsrahmen

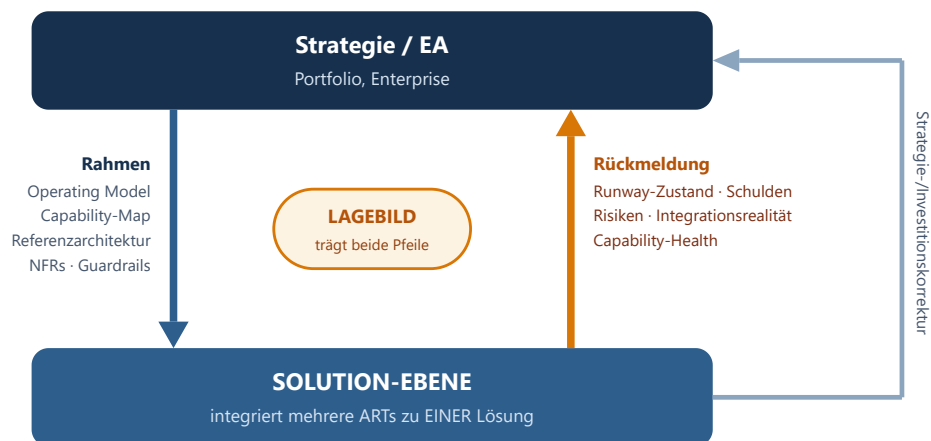


Abb. 1 – Der Regelkreis: EA gibt den Rahmen nach unten, die Solution-Ebene meldet die architektonische Realität zurück – das Lagebild trägt beide Richtungen.

Das **Lagebild ist der Träger beider Pfeile**: Es macht den Rahmen und die Rückmeldung sichtbar – oder der Regelkreis existiert nur auf Folien.

C.2 Grundlegendes Wissen (alle Beteiligten)

Capability-Denken & Operating Model · Solution Intent/NFRs/Architecture Runway · Realitäten verteilter Systeme (CAP, Eventual Consistency, Idempotenz, Saga/CQRS) · As-is/To-be über Change-Ebenen · Governance-als-Befähigung · Mess-/Metrikgrundlagen (Flow + SLO + Qualität; GQM; Goodhart) · ggf. Regulatorik.

C.3 Skills – notwendig vs. dienlich (Trennkriterium)

Notwendig (ohne sie bricht die Funktion oder das Lagebild wird irreführend)

ART-übergreifende
Abhängigkeits-/Integrationskoordination · NFR-/Runway-Stewardship · Capability-/Value-Stream-Modellierung · Trade-off-Entscheidung unter Unsicherheit · Übersetzung Business↔Architektur↔Entwicklung (Rosetta Stone) · Risiko-Facilitation (ROAM) · korrekte Metrik-Interpretation · Systemdenken.

Dienlich (Verstärker)

tiefe Stack-Expertise · Wardley Mapping · Set-Based Design · Red-Teaming/Premortem · Storytelling mit Daten · MLOps-/KI-Bewusstsein · kalibriertes Schätzen/Forecasting (Hubbard, Monte-Carlo).

Die Tierung ist **rollen-relativ** (STE ↔ Solution Architect ↔ Solution Management ↔ System Team): siehe Matrix im Vorarbeitspapier.

C.4 Die fünf Lagebild-Prinzipien (destilliert)

1 Capability vor Feature

Der Report zeigt Gesundheit von Geschäftsfähigkeiten, nicht Ticket-Zählung – sonst misst er Output statt Wirksamkeit (Graves).

2 Schulden sind Erstklass-Bürger

NFR-Erfüllung und Runway-Zustand stehen gleichberechtigt neben Flow – unsichtbare Architekturschuld ist das teuerste Risiko.

3 Zwei Lesarten, ein Bild

Executive-Sicht und Engineering-Sicht aus denselben Daten (Rosetta Stone, Carducci) – nie zwei getrennte Wahrheiten.

4 Konfidenz gehört zur Zahl

Jede gepoolte Kennzahl trägt Herkunft und Vertrauensgrad – gegen die „Information Masquerade“ (Milchman).

5 Das Lagebild ist Governance-Instrument

ROAM-Risiken, Entscheidungen und Annahmen sind Teil des Reports – er *stützt* Entscheidungen, statt sie nur zu illustrieren (EA-Tätigkeit 3 und 4 aus B.1).

Teil D – Konkrete Programme & Umsetzung

D.1 Befähigung der Beteiligten (einkaufbar/lernbar)

- **EA-/Architektur-Ausbildung:** kontextnah statt Framework-Zertifikatssammlung – Lektürepfad: Carducci (Rolle) → Graves (Enterprise-Sicht) → A-ESA (Solution-Methode) → Milchman (soziotechnisch); flankierend Ross/Weill. *Vertiefung (v2.0, Teil A2):* Hohpe (*Software Architect Elevator*) als Einstieg in die Vermittlerrolle, Fowler (*Refactoring*, PoEAA) fürs Evolutions- und Pattern-Handwerk, Martin (*Clean Architecture*) für Grenzen und Abhängigkeitsregeln, Hohmann (*Beyond Software Architecture, Software Profit Streams*) für die Geschäftsmodell-Flanke. TOGAF-Kurse allenfalls als Vokabeltraining.
- **Entscheidungs- & Stabskompetenz:** bereits in der Denkschriften-Reihe verankert – 12-Monats-Curriculum, Red-Teaming, Kalibrierung (Hubbard), Superforecasting: siehe Team-of-Teams-Denkschrift Teil D. Die EA-Skills docken dort als Modul an (Empfehlung: Capability-Mapping + NFR-Workshop als eigener Baustein im Curriculum).
- **Technisches Grundwissen:** Lesefade im Werkstatt-Archiv des Kurators (z. B. „verteilte Daten/Kommunikation“: Immutable Architecture → Go & NATS; „Betrieb/Metrik-Brücke“: Cloud-Native DevOps → SRE with Azure → Fault Detection).

D.2 Einstieg für Eilige (90 Tage)

Woche 1–2 Kartieren beginnen

Capability-Map der Solution skizzieren (Workshop, 1 Tag) + NFR-Register anlegen (die zehn wichtigsten).

Woche 3–6 Risiken & Entscheidungen strukturieren

ROAM-Board etablieren; PI-Planning um Dependency-Register erweitern; Decision-Log beginnen (ADR-leichtgewichtig).

Woche 7–12 Lagebild erweitern

Lagebild um die vier EA-Dimensionen ergänzen (zunächst manuell gepflegt, Werkzeug folgt); Zwei-Lesarten-Gliederung einführen; erste Retro auf das Lagebild selbst („misst es Wirksamkeit?“).

D.3 Software-Umsetzung

Die technische Roadmap für das `portfolio`-Modul steht in `docs/portfolio_roadmap_solution-lagebild.md` – gegliedert nach Datenherkunft: **Phase A** (aus vorhandenen Jira-Daten: Konfidenz-Flag, erweiterte Summary), **Phase B** (neue leichte Eingabe-Artefakte: Capability-Map, NFR-/Runway-Register, ROAM-Board, Decision-Log, Dependency-Register), **Phase C** (externe Integrationen: SLO/Error-Budget, DORA/Qualität, Jira-REST).

KERNAUSSAGE DER SOFTWARE-ROADMAP

Die EA-Dimensionen haben heute **keine Quelle im Datenstrom** – sie brauchen eigene Eingaben; **genau dort entsteht der Mehrwert**.

Anschlussfähigkeit an die bestehende Reihe (Merkposten)

1

Team-of-Teams-Denkschrift

Dort Führung/Prozess (Shared Consciousness, Empowered Execution, Stabsausbildung) – hier Struktur/Architektur (Capabilities, Runway, Governance). Zusammen ergeben sie die zwei Hälften der Solution-Befähigung: *wie wir führen und worüber wir sprechen*.

2

Lagebild-Literaturliste

Diese Denkschrift vertieft deren „Ebene 3 – Large Solution“ (A–D) architektonisch; die Skills-Liste dort (C) wird durch die Notwendig/Dienlich-Tierung geschärft.

3

Agile-Metriken-Reihe

Das BGS-Prinzip (Beobachtet/Geschätzt/Systemgeneriert) strukturiert die Software-Roadmap; die Flow-Metriken bleiben das ökonomische Rückgrat, die EA-Dimensionen ergänzen das strukturelle.

Quellenverzeichnis

Primär (EA-Korpus; Volltexte liegen dem Kurator vor): 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.

Ergänzende Kernquellen (Teil 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*, 2. Aufl. 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.

Stützend (Auswahl aus Korpus & Lagebild-Liste): Perry, Michael L.: *The Art of Immutable Architecture*, 2. Aufl. 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 (→ eigene Denkschrift der Reihe).

Interne Bezüge: Team-of-Teams-Denkschrift „Team of Teams & Stabsausbildung“ · Lagebild-Literaturliste, EA-Vorarbeit und EPUB-Arbeitsdokumente im Werkstatt-Archiv des Kurators · Software-Roadmap: [docs/portfolio_roadmap_solution-lagebild.md](#).

Versionshistorie

Version	Datum	Änderungen
1.0	06.08.2026	Erstfassung: vier EA-Quellen im Profil (Graves, Carducci, A-ESA, Milchman), EA-Handwerk & Übertragung, fünf Lagebild-Prinzipien, 90-Tage-Einstieg, Verzahnung mit Team-of-Teams-Reihe und Software-Roadmap.
1.1	07.08.2026	Grafische Neuausstattung im Reihen-Design der Team-of-Teams-Denkschrift (Cover, Karten, Nummernkreise, Regelkreis-Abbildung statt ASCII-Diagramm); „Wichtigster Einzelsatz des Dokuments“ auf Seite 1 als neuer Reihen-Standard; Umbruch-/Leseflussregeln (Überschriften bleiben beim Folgetext). Inhalt unverändert gegenüber 1.0.
2.0	07.08.2026	Vertiefung: neuer Teil A2 mit vier ergänzenden Vordenkern – 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-Lektürepfad erweitert; Literaturlisten-Ergänzung v3.
2.1	14.08.2026	Verweispflege für die Online-Veröffentlichung: Querverweise auf die Reihe verlinken die Online-Übersicht, Werkstatt-Pfade entfernt; Querverweise nun ohne Versionsnummern. Inhalt unverändert.
2.2	18.08.2026	Transparenzhinweis nach dem Reihen-Standard (KI-Unterstützung, Prüfung, redaktionelle Verantwortung) in der Kopfbox ergänzt; Versionierungs-Grundsätze 5/6 der Reihe. Inhalt unverändert.

Quellen-Briefing „Enterprise-Architektur & Solution-Lagebild“ · Version 2.2 · Solution-/Portfolio-Management · Stand 18.08.2026 · kuratiert von Robert Seebauer, erstellt mit Claude (Anthropic) · Aktuelle Fassungen online: <https://jaegerfeld.github.io/situation-report/de/denkschriften/>