

BEST-PRACTICE GUIDE · TRANSFORM

Phase 04 — Design

From requirement to a documented, decidable architecture

The Design phase transforms the requirements model into an agentic-ready target architecture: domain cuts, agent and tool architecture, data architecture and guardrails. Every architecture decision is explicitly documented and evaluated against quality requirements.

Goal of the phase

The phase delivers a target architecture documented per arc42, traceable through architecture decision records, with a defined agent topology, API contracts and a migration design from legacy to the target landscape.

Process model

- 01 Domain cut (domain-driven design)**
Derive bounded contexts and a ubiquitous language from the process and requirements model; strategic design with context mapping as the basis for the system and agent cut.
- 02 Agent & tool architecture**
Design the agent topology (orchestration, specialist agents, human-in-the-loop points), tool interfaces and context supply (retrieval, knowledge bases) along ReqPOOL's reference architecture.
- 03 System & data architecture**
Target landscape with API-first interfaces, event flows and data products; derivation of the migration architecture from the Analyse phase's horseshoe model.
- 04 Quality & guardrail design**
Attribute-driven design against prioritized quality scenarios (ISO 25010); definition of security, compliance and AI guardrails including authorization and audit concept.
- 05 Architecture decision & documentation**
Evaluate alternatives, fix them as architecture decision records and consolidate the arc42 documentation; review along ISO 42010 stakeholder viewpoints.

Methodology mix

METHOD	PURPOSE	REFERENCE
Domain-driven design (strategic)	Bounded contexts and context mapping as the functional cut for systems and agents; prevents monolithic agent landscapes.	Evans (2003)
arc42 & C4 model	Standardized architecture documentation and hierarchical visualization from context to component; ReqPOOL's documentation standard.	arc42; Brown
Attribute-driven design (ADD 3.0)	Iterative architecture development from prioritized quality attribute scenarios; anchors non-functional requirements in the design.	Bass, Clements & Kazman (SEI)

METHOD	PURPOSE	REFERENCE
ISO/IEC/IEEE 42010	Standard for architecture descriptions: stakeholders, concerns, viewpoints and correspondence rules.	ISO/IEC/IEEE 42010:2022
ISO/IEC 25010 quality model	Taxonomy of product quality (incl. reliability, security, maintainability) as the evaluation frame for quality scenarios.	ISO/IEC 25010:2023
ReqPOOL Agentic Reference Architecture	In-house reference architecture for agentic systems: orchestration, tool contracts, context supply, guardrails and evaluation loops.	ReqPOOL (internal); cf. Anthropic (2024)

Platform support

ReqPOOL Suite: **REAM** (Enterprise architecture) · **Requirements Manager** (Agentic requirements management)

REAM — AS-IS landscape as design basis	Processes, applications, interfaces and infrastructure from the inventory form the fact-based starting point of the architecture draft.
REAM — EAM decision board	AI-derived architecture recommendations are decided and documented in the board — every recommendation traceable down to the data basis.
REAM — target architecture & transformation roadmap	Decision-ready target landscape and roadmap per year, including the migration stages from the Analyse phase's gap analysis.
Requirements Manager — AI coding intent	The machine-readable implementation order translates the solution design (use cases, business objects, interfaces, permissions) into a directly implementable form for the Build phase.

Best-practice checklist

	Always decide AI-derived architecture recommendations in the EAM board — the decision stays with humans and is documented with its data basis.
	Record every accepted or rejected alternative as an ADR; architecture without a decision history counts as undocumented.
	Prioritize quality scenarios before drafting structure — ADD starts with drivers, not with technology.
	Explicitly mark human-in-the-loop points in the agent design and reconcile them with the governance frame from the Strategy phase.
	Review the AI coding intent against the target architecture and ADRs before it goes to implementation.
	Always think migration design backwards from target operations: every interim stage must be independently operable and rollback-capable.

Artifacts & outcomes

- Target architecture per arc42 with C4 diagrams
- Agent topology with tool contracts and context supply
- Architecture decision records (ADRs)
- Quality and guardrail concept (ISO 25010, EU AI Act)
- Migration design legacy → target architecture

Quality gate criteria

☐	Target architecture approved by the architecture board
☐	All critical decisions documented as ADRs
☐	Quality scenarios evaluated, residual risks accepted
☐	Migration stages defined with rollback capability

Scientific deep dives

Evans: Domain-Driven Design (2003) — Foundational work on domain-driven software design.
<https://www.domainlanguage.com/ddd/>

Bass, Clements & Kazman: Software Architecture in Practice (4th ed.) — SEI standard reference on quality attributes and architecture methods.
<https://insights.sei.cmu.edu/library/software-architecture-in-practice-fourth-edition/>

ISO/IEC/IEEE 42010:2022 — Architecture Description — Standard for architecture descriptions and stakeholder viewpoints.
<https://www.iso.org/standard/74393.html>

ISO/IEC 25010:2023 — Product Quality Model — Quality model for systems and software.
<https://www.iso.org/standard/78176.html>

arc42 — template and documentation — De-facto standard for architecture documentation.
<https://arc42.org/>

C4 Model — Simon Brown — Hierarchical model for architecture visualization.
<https://c4model.com/>

Anthropic: Building Effective Agents (2024) — Research-based design patterns for agentic systems.
<https://www.anthropic.com/research/building-effective-agents>