

Softwarearchitektur für gewachsene Systeme

Bestehende Architekturen rekonstruieren, stabilisieren und evolutionär weiterentwickeln

01 INTRO

Bei gewachsenen Systemen entspricht die reale Architektur häufig weder der ursprünglichen Dokumentation noch dem gewünschten Zielbild. Abhängigkeiten haben sich verschoben, Grenzen sind erodiert und technische Entscheidungen wurden unter unterschiedlichen Rahmenbedingungen getroffen.

Der Kurs behandelt Architektur deshalb nicht als Neuentwurf auf der grünen Wiese, sondern als Arbeit an einem existierenden System. Die Teilnehmer rekonstruieren Ist-Architekturen, identifizieren strukturelle Probleme und entwickeln realistische Schritte zu einer besser beherrschbaren Architektur.

DAUER

3 Tage

FORMAT

Präsenz · Live Online · Inhouse

VORKENNTNISSE

Erfahrung in Softwareentwicklung und grundlegendes Verständnis von Softwarearchitektur

02 TRAININGSPROGRAMM

Inhalte

Architektur im gewachsenen System

- geplante Architektur
- reale Architektur
- Architekturdrift
- Erosion
- Constraints
- historische Entscheidungen
- technische Schulden

Ist-Architektur rekonstruieren

- Code
- Module
- Build-Struktur
- Abhängigkeiten
- Laufzeit
- Deployment
- Daten
- Dokumentation

Abhängigkeiten analysieren

- statische Abhängigkeiten
- Laufzeitabhängigkeiten
- Zyklen
- Kopplung
- gemeinsame Daten
- externe Systeme
- Dependency Graphs

Grenzen untersuchen

- Module
- Komponenten
- fachliche Grenzen
- technische Grenzen
- Verantwortlichkeiten
- Kohäsion
- Boundary Erosion

Architekturprobleme erkennen

- Big Ball of Mud
- Layer Violations
- Distributed Monolith
- Shared Database
- God Components
- versteckte Kopplung
- unklare Ownership

Qualitätsziele für die Weiterentwicklung

- Änderbarkeit
- Testbarkeit
- Deployability
- Performance
- Zuverlässigkeit
- Security
- Observability
- Priorisierung

Zielbilder entwickeln

- realistische Zielarchitektur
- Modularisierung
- Ports and Adapters
- fachliche Grenzen
- Services
- Datenhoheit
- Trade-offs

Architektur schrittweise verändern

- Architectural Refactoring
- Branch by Abstraction
- Strangler Fig
- Adapter
- Facades
- Parallelbetrieb
- Feature Flags

Datenarchitektur entkoppeln

- Shared Database
- Schema Ownership
- Zugriff kapseln
- Replikation
- Migration
- Konsistenz
- Übergangslösungen

Architektur absichern

- Architekturtests
- Dependency Rules
- Fitness Functions
- Continuous Integration
- Reviews
- Architecture Decision Records
- technische Governance

Evolutionäre Architektur

- kleine Entscheidungen
- Feedback
- Optionen offenhalten
- irreversible Entscheidungen
- Architekturmetriken
- kontinuierliche Bewertung
- Dokumentation

Architektur-Sanierungsprojekt

- Ist-Architektur visualisieren
- Probleme identifizieren
- Qualitätsziele definieren
- Zielbild entwickeln
- Übergangsschritte planen
- Risiken bewerten
- Architekturregeln festlegen

03 DURCHFÜHRUNG

Rahmen und Organisation

Dauer	3 Tage
Durchführung	Präsenz oder Live Online, vorzugsweise als Inhouse-Schulung
Schwerpunkt	Inhouse-Schulungen für Unternehmen und Teams
Sprache	Deutsch, englische Fachbegriffe und Dokumentation
Unterlagen	Kursunterlagen, Beispiele und Übungsaufgaben
Technik	Eigener Rechner für die praktischen Übungen

Inhouse-Schulungen

Die eigene Systemarchitektur kann als Grundlage für Rekonstruktion und Sanierungsplanung dienen.

04 ZIELGRUPPE

Für wen ist der Kurs gedacht?

Softwarearchitekten, Senior-Entwickler und technische Leads, die bestehende Architekturen weiterentwickeln müssen.

Voraussetzungen

Praktische Softwareentwicklung und grundlegende Architekturkenntnisse.

05 LERNZIELE

Was Sie nach dem Kurs können

Die Teilnehmer können reale Architekturen rekonstruieren, Architekturdefizite analysieren, realistische Zielbilder entwickeln und strukturelle Veränderungen schrittweise absichern.

06 FAQ

Häufige Fragen

Worin unterscheidet sich der Kurs von den Architekturgrundlagen?

Der Grundlagenkurs vermittelt Architekturprinzipien und Entwurf. Hier ist der Ausgangspunkt ausdrücklich eine bereits vorhandene, gewachsene Architektur mit ihren Constraints.

Wie grenzt sich der Kurs gegen die Legacy-Modernisierung ab?

Der Modernisierungskurs behandelt ein Legacy-System insgesamt. Hier liegt der Schwerpunkt auf dessen architektonischen Strukturen und deren evolutionärer Veränderung.

Geht es um die Migration zu Microservices?

Nur wenn Microservices für das konkrete Problem sinnvoll sind. Eine saubere Modularisierung kann die bessere Zielarchitektur sein.

UC IT Service · Ulrich Cuber · kontakt@uc-it.de
<https://www.uc-it.de/coaching/architektur-gewachsene-systeme/>