

Software-Architektur-Schulung für Embedded-Systeme und Echtzeitsysteme - Live-Online-Training

Ziele - Ihr Nutzen

Lernen Sie die Begriffe und Bedeutung von Software-Architektur, die Aufgaben und Verantwortungen von Software-Architekten und deren Rolle im Projekt sowie State-of-the-Art Methoden und Techniken zur Entwicklung von Software-Architekturen kennen.

In der Software-Architektur-Schulung erfahren Sie, wie Sie Software-Architekturen mit den Projektbeteiligten abstimmen, dokumentieren und kommunizieren. Anschließend können Sie die wesentlichen Schritte des Software-Architekturentwurfs selbstständig durchführen.

Teilnehmer

Die Software-Architektur-Schulung richtet sich an Software-Architekten, Software-Entwickler, Software-Entwicklungsleiter und Software-Teamleiter.

Voraussetzungen

Projekterfahrung in der Softwareentwicklung, Kenntnis einer höheren Programmiersprache; Grundlagen der UML-Notation zur Software-Modellierung sind von Vorteil.

Live Online Training

28.09. – 01.10.2026 2.600,00 € 4 Tage

* Preis je Teilnehmer, in Euro zzgl. USt.

Anmeldecode: L-EMB-AR

Präsenz-Training - Deutsch

Termin	Dauer
29.06. – 02.07.2026	4 Tage
07.12. – 10.12.2026	4 Tage

Live-Online - Englisch

Dauer
4 Tage

Präsenz-Training - Englisch

Termin	Dauer
29.06. – 02.07.2026	4 Tage

Software-Architektur-Schulung für Embedded-Systeme und Echtzeitsysteme - Live-Online-Training

Inhalt

Software-Architekturen: Einleitung und Grundbegriffe

- Begriffsdefinitionen
- Nutzen und Ziele einer Software-Architektur
- Elemente von Software-Architekturen
- Bezug zum Entwicklungsprozess
- Rolle und Verantwortlichkeiten des Software-Architekten
- Praxistipps

Vorgehen in der Software-Architektur-Entwicklung

- Unterschiedliche Vorgehensweisen in der Software-Architektur-Entwicklung: hierarchisch, iterativ, inkrementell, agil, modellgetrieben, domänengetrieben
- Abhängigkeiten und Zusammenhänge der unterschiedlichen Vorgehensweisen
- Darstellung eines typischen Vorgehens

Die Rolle des Software-Architekten

- Eigenschaften und Verantwortlichkeiten des Software-Architekten
- Software-Architekten-Teams
- Zusammenarbeit mit anderen Rollen

Basis und Voraussetzungen für die Software-Architektur-Entwicklung

- Typische Entwicklungsszenarien mit guten und schlechten Voraussetzungen
- Anforderungen: funktional und nicht-funktional
- Einflussfaktoren für die Software-Architektur
- Demonstration einer Einflussfaktor-Analyse
- Risikomanagement

Notation und Dokumentation von Software-Architekturen mit der UML (Unified Modeling Language)

- Statische und dynamische Sichten
- Qualitätsmerkmale der Dokumentation von Software-Architekturen
- Software-Architekturen stakeholdergerecht dokumentieren und kommunizieren
- Dokumentationsinhalte und -schwerpunkte
- Beschreibung und Kommunikation von Schnittstellen / Interfaces
- Definition und Nutzen verschiedener Architektursichten
- Dokumentenzentriertes versus modellzentriertes Vorgehen
- Demonstration: Von den Anforderungen zum Software-Architekturmodell
- Übung: Sie entwickeln auf Basis ausformulierter Anforderungen eine Embedded-Softwarearchitektur bestehend aus Struktur und interaktivem Verhalten

Entwurf von Software-Architekturen

- Funktionale und nicht-funktionale Anforderungen als Basis für eine qualitativ hochwertige Software-Architektur
- Positiver und negativer Einfluss von Qualitätsanforderungen auf die Software-Architektur
- Auswirkung von Sicherheit (Safety und Security), Zuverlässigkeit, Portierbarkeit, Performance und anderen Qualitätsanforderungen auf die Architektur

- Projektspezifische Einflussfaktoren auf die Software-Architektur
- Kontinuierliche Verfeinerung der Software-Architektur durch inkrementelles und iteratives Vorgehen
- Architekturbaukasten und Wiederverwendbarkeit
- Grundlegende Konzepte, Bausteine, Bausteinkopplung über Schnittstellen / Interfaces
- Entwurfsprinzipien
- Architektur-Entwurfsmuster (Pattern)
- Architekturrichtlinien
- Laufzeitarchitektur
- Architekturen für Multiprozessor- und Multicore-Systeme
- Hypervisor und Virtualisierung
- Plattform-Unabhängigkeit
- Verifikation von Software-Architekturen
- Übung: Sie ergänzen die bestehende Architektur um eine Laufzeitarchitektur

Qualitätsbewertung und Qualitätssicherung von Software-Architekturen

- Vorstellung von Qualitätsmodellen
- Zusammenhang und Wechselwirkung zwischen Qualitätsmerkmalen
- Vorgehensweisen zum Erreichen der spezifizierten Qualitätsmerkmale von Software-Systemen
- Bewertung von Software-Architekturen (Qualität und Umsetzbarkeit)
- ATAM (Architecture Tradeoff Analysis Method)
- Praxistipps für die Qualitätsbewertung und Qualitätssicherung

- Übung: Sie bewerten die Qualität verschiedener Software-Architekturen

Tools für den Software-Architekten

- Modellierung
- Statische und dynamische Analyse
- Generierung
- Anforderungsmanagement
- Dokumentation
- Versions- und Konfigurationsmanagement
- Build-Prozess und Build-Systeme
- Zur Produktauswahl erhalten Sie zu jedem Toolthema eine wertfreie Produktübersicht

Praxisbewährte Beispiele von Software-Architekturen

- Typische Softwarearchitekturen für Embedded-Systeme
- Softwareschichten/Software-Layer-Modelle
- Tipps aus der Praxis

Praktische Übung in der Software-Architektur-Schulung

- Von den Anforderungen bis zur Verifikation und Bewertung der Software-Architektur am Beispiel eines Embedded-Systems (komfortable Elektromotorsteuerung) auf Basis einer realen Hardware
- Zur Bearbeitung der durchgängigen Übung haben Sie die Wahl zwischen "Papier und Bleistift" oder der Verwendung des Modellierungstools 'Enterprise Architect' der Firma Sparx Systems.

MicroConsult PLUS:

- Sie erhalten von uns Ihre Übungsverzeichnisse und Lösungsbeispiele für alle Übungsaufgaben.
- Sie erhalten eine Tool- und Software-Komponentenübersicht für die Entwicklung von Embedded-Software-Architekturen.
- Ebenso erhalten Sie eine für Ihre Projekte anwendbare und modifizierbare Embedded-Softwarearchitektur-Checkliste als Zusammenfassung aller wichtigen Themenpunkte.
- Sie bekommen Notationsübersichten für die UML und SysML.