

Legacy-Systeme verstehen und modernisieren

Gewachsene Softwaresysteme analysieren und kontrolliert weiterentwickeln

01 INTRO

Legacy-Systeme sind häufig geschäftskritisch, schlecht dokumentiert und über Jahre durch technische und organisatorische Veränderungen geprägt worden. Gerade deshalb ist ein vorschneller Rewrite oft riskanter als eine kontrollierte Weiterentwicklung.

Der Kurs vermittelt Methoden, um unbekannte oder gewachsene Systeme systematisch zu verstehen, Risiken sichtbar zu machen und geeignete Modernisierungsschritte zu entwickeln. Dabei werden Code, Daten, Schnittstellen, Abhängigkeiten und Betrieb gemeinsam betrachtet.

DAUER

3 Tage

FORMAT

Präsenz · Live Online · Inhouse

VORKENNTNISSE

Mehrjährige praktische Erfahrung in Softwareentwicklung ist hilfreich

02 TRAININGSPROGRAMM

Inhalte

Was macht ein System zu Legacy?

- Alter versus Änderbarkeit
- Geschäftskritikalität
- fehlendes Wissen
- Technologien
- Abhängigkeiten
- fehlende Tests
- technische Schulden

Systematisch verstehen statt raten

- vorhandene Dokumentation
- Code
- Build
- Deployment
- Logs
- Datenbanken
- Laufzeitverhalten
- Stakeholder

Systemstruktur rekonstruieren

- Module
- Komponenten
- Schnittstellen
- Abhängigkeiten
- Datenflüsse
- externe Systeme
- Visualisierung

Verhalten erfassen

- fachliche Prozesse
- Characterization Tests
- Golden Master
- Produktionsbeobachtung
- Logs
- Beispiele
- unbekannte Regeln

Daten verstehen

- Datenmodell
- Datenhoheit
- gemeinsame Tabellen
- implizite Verträge
- historische Daten
- Datenqualität
- Migration

Schnittstellen und Abhängigkeiten

- interne APIs
- externe APIs
- Dateien
- Messaging
- Batch
- Datenbanken
- zeitliche Kopplung
- versteckte Integrationen

Risiken bewerten

- Änderungsrisiko
- Geschäftskritikalität
- Single Points of Failure
- Wissen
- Security
- Betrieb
- technologische Risiken

Modernisierungsstrategien

- Maintain
- Refactor
- Replatform
- Replace
- Rewrite
- Strangler Fig
- schrittweise Ablösung

Grenzen schaffen

- Module
- Facades
- Adapter
- Anti-Corruption Layer
- APIs
- Datenzugriff kapseln
- Seams
- neue Komponenten

Migration planen

- kleine Schritte
- Parallelbetrieb
- Datenmigration
- Schnittstellen
- Rollback
- Kompatibilität
- Abschaltkriterien

Betrieb während der Modernisierung

- Observability
- Monitoring
- Deployment
- Feature Flags
- Fehlerdiagnose
- Regression
- sichere Releases

Modernisierungs-Workshop

- Legacy-System analysieren
- Risiken kartieren
- Ziel definieren
- Strategie auswählen
- Schritte planen
- Abhängigkeiten berücksichtigen
- Entscheidung begründen

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

Ein eigenes Legacy-System kann als Grundlage für Analyse und Modernisierungsplanung dienen.

04 ZIELGRUPPE

Für wen ist der Kurs gedacht?

Erfahrene Entwickler, Softwarearchitekten und technische Leads, die gewachsene Systeme übernehmen oder modernisieren.

Voraussetzungen

Praktische Erfahrung mit größeren Softwaresystemen.

05 LERNZIELE

Was Sie nach dem Kurs können

Die Teilnehmer können Legacy-Systeme strukturiert untersuchen, technische und fachliche Risiken sichtbar machen, Modernisierungsoptionen gegeneinander abwägen und schrittweise Migrationsstrategien entwickeln.

06 FAQ

Häufige Fragen

Ist ein Rewrite nicht häufig einfacher?

Am Anfang kann er einfacher erscheinen. Bei geschäftskritischen Legacy-Systemen steckt jedoch oft jahrelang gewachsenes, teilweise undokumentiertes Verhalten im bestehenden System.

Worin unterscheidet sich der Kurs vom Refactoring-Kurs?

Der Refactoring-Kurs arbeitet primär am bestehenden Code. Hier wird das gesamte System einschließlich Daten, Schnittstellen, Betrieb und Migrationsstrategie betrachtet.

Wie grenzt sich der Kurs gegen die Legacy-Kurse im Quality Engineering ab?

Die Kurse zum Testen von Legacy-Systemen und zur Qualität bei Modernisierung und Migration betrachten das Thema aus Sicht der Qualitätssicherung. Hier geht es um die technische Analyse und Modernisierung des Systems selbst.

UC IT Service · Ulrich Cuber · kontakt@uc-it.de

<https://www.uc-it.de/coaching/legacy-systeme-modernisieren/>