

Architektur von Testautomatisierungslösungen

Wartbare und skalierbare Testautomatisierung systematisch entwerfen

01 INTRO

Eine Testautomatisierungslösung ist ein Softwaresystem. Mit wachsender Anzahl automatisierter Tests werden Architektur, Schnittstellen, Abstraktionen, Testdaten, Konfiguration und Diagnosefähigkeit entscheidend für Wartbarkeit und Lebensdauer.

Der Kurs behandelt Testautomatisierung aus Architekturperspektive. Die Teilnehmer entwickeln Strukturen für größere Automatisierungslösungen, trennen fachliche Tests von technischer Interaktion und gestalten Lösungen, die sich an Änderungen von Anwendung, Technologie und Testumfang anpassen können.

DAUER

3 Tage

FORMAT

Präsenz · Live Online · Inhouse

VORKENNTNISSE

Praktische Erfahrung mit Testautomatisierung und Programmierkenntnisse

02 TRAININGSPROGRAMM

Inhalte

Testautomatisierung als Softwaresystem

- Architekturziele
- Wartbarkeit
- Erweiterbarkeit
- Stabilität
- Wiederverwendung
- technische Schulden
- Lebenszyklus einer Testlösung

Anforderungen an die Testarchitektur

- Testobjekt
- Testebenen
- Technologien
- Teamstruktur
- Ausführungsumgebung
- Testvolumen
- Änderungsfrequenz
- Qualitätsattribute

Schichten und Verantwortlichkeiten

- Testfälle
- fachliche Aktionen
- technische Adapter
- Treiber
- Daten
- Konfiguration
- Infrastruktur
- Separation of Concerns

Abstraktionsmuster

- Page Object
- Component Object
- Screenplay
- Service Layer
- Facades
- DSLs
- geeignete Abstraktionstiefe
- Overengineering vermeiden

Testcode strukturieren

- Module und Packages
- gemeinsame Bibliotheken
- Fixtures
- Helper
- Assertions
- Namenskonventionen
- Dependency Management

Mehrere Testebenen integrieren

- UI
- API
- Integration
- Datenbank
- End-to-End
- gemeinsame fachliche Komponenten
- Testpyramide in der Architektur

Testdatenarchitektur

- statische und dynamische Daten
- Builder und Factories
- Datenbereitstellung
- Setup und Cleanup
- Isolation
- Parallelisierung
- externe Datenquellen

Konfiguration und Umgebungen

- Environment Configuration
- Secrets
- Endpunkte
- Feature Flags
- Browser und Plattformen
- lokale und CI-Ausführung
- reproduzierbare Konfiguration

Stabilität und Synchronisation

- Wait-Strategien
- asynchrone Prozesse
- Retries
- Timeouts
- Race Conditions
- Isolation
- Flaky Tests architektonisch vermeiden

Diagnosefähigkeit und Observability

- Logging
- Screenshots
- Tracing
- Netzwerkdaten
- Reports
- Fehlerkontext
- Artefakte
- schnelle Ursachenanalyse

Skalierung und CI/CD

- parallele Ausführung
- Sharding
- Testselektion
- schnelle und langsame Suites
- Pipeline-Stufen
- Container
- Ressourcen
- Quality Gates

Architekturprojekt

- Anforderungen analysieren
- Testebenen festlegen
- Komponenten entwerfen
- Abstraktionen definieren
- Testdaten integrieren
- CI-Ausführung berücksichtigen
- Architekturentscheidungen 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 mit Entwicklungsumgebung für die Architekturübungen

Inhouse-Schulungen

Eine vorhandene Testautomatisierungslösung eignet sich besonders gut als Grundlage für Analyse und Architekturarbeit.

04 ZIELGRUPPE

Für wen ist der Kurs gedacht?

Erfahrene Testautomatisierer, Quality Engineers, Entwickler, Test Architects und technische QA Leads.

Voraussetzungen

Praktische Erfahrung mit Testautomatisierung und sichere Grundlagen in mindestens einer Programmiersprache.

05 LERNZIELE

Was Sie nach dem Kurs können

Die Teilnehmer können Anforderungen an größere Testautomatisierungslösungen analysieren, geeignete Architekturstrukturen und Abstraktionen entwickeln und Wartbarkeit, Skalierbarkeit und Diagnosefähigkeit systematisch berücksichtigen.

06 FAQ

Häufige Fragen

Ist der Kurs frameworkabhängig?

Nein. Architekturprinzipien werden anhand konkreter Technologien illustriert, sind aber nicht an ein bestimmtes Framework gebunden.

Was unterscheidet den Kurs von den Grundlagen?

Der Grundlagenkurs vermittelt Testautomatisierung als Gesamtprozess. Dieser Kurs richtet sich an erfahrenere Teilnehmer und konzentriert sich auf die Architektur größerer Automatisierungslösungen.

Werden bestehende Frameworks analysiert?

Ja. Bestehende Strukturen eignen sich besonders gut, um Architekturentscheidungen und ihre Folgen zu untersuchen.

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

<https://www.uc-it.de/coaching/testautomatisierung-architektur/>