

Integration-, System- und End-to-End-Tests

Zusammenspiel komplexer Systeme gezielt und beherrschbar prüfen

01 INTRO

Viele kritische Fehler entstehen nicht innerhalb einzelner Komponenten, sondern an ihren Grenzen: unterschiedliche Datenmodelle, Protokolle, Zustände, Timing und Annahmen treffen aufeinander. Je größer der betrachtete Systemausschnitt wird, desto realistischer, aber auch aufwendiger und instabiler werden Tests.

Der Kurs behandelt Integration-, System- und End-to-End-Tests als unterschiedliche Ebenen mit jeweils eigenen Zielen. Die Teilnehmer entwickeln eine Testarchitektur, die Fehler möglichst nah an ihrer Ursache erkennt und gleichzeitig zentrale Geschäftsabläufe über Systemgrenzen hinweg absichert.

DAUER

3 Tage

FORMAT

Präsenz · Live Online · Inhouse

VORKENNTNISSE

Grundkenntnisse in Softwaretest und Verständnis verteilter Softwaresysteme

02 TRAININGSPROGRAMM

Inhalte

Teststufen und Systemgrenzen

- Komponenten
- Integration
- System
- End-to-End
- unterschiedliche Testziele
- Verantwortlichkeiten
- sinnvolle Abgrenzung

Integrationsrisiken analysieren

- Schnittstellen
- Datenformate
- Protokolle
- Versionen
- Zustände
- Timing
- Fehlerbehandlung
- externe Systeme

Integrationsstrategien

- inkrementelle Integration
- Bottom-up und Top-down
- Big Bang
- serviceorientierte Integration
- Contract-first
- Auswahl nach Architektur

Schnittstellen systematisch testen

- Verträge
- Request und Response
- Events
- Daten
- Fehlerfälle
- Kompatibilität
- fachliche Integrationsregeln

Test Doubles und Service Virtualization

- Mocks
- Stubs
- Fakes
- simulierte Services
- externe Abhängigkeiten
- kontrollierte Fehler
- Grenzen der Simulation

Contract Testing

- Consumer und Provider
- Verträge
- Schemaänderungen
- Kompatibilität
- Consumer-driven Contracts
- Position in der Testarchitektur

Systemtests

- vollständiges System
- funktionale Anforderungen
- nichtfunktionale Eigenschaften
- realistische Konfiguration
- Testdaten
- Umgebungen
- Systemzustände

End-to-End-Tests

- Geschäftsprozesse
- mehrere Systeme
- reale Integrationen
- kritische Pfade
- Aussagekraft
- Kosten
- Instabilität
- bewusste Begrenzung

Daten über Systemgrenzen

- IDs
- Transformationen
- Mapping
- Konsistenz
- asynchrone Verarbeitung
- Eventual Consistency
- Datenzustände verifizieren

Verteilte und asynchrone Systeme

- Messaging
- Events
- Queues
- Timing
- Retries
- Duplikate
- Reihenfolgen
- Timeouts

Testarchitektur und CI/CD

- Testpyramide
- Integrationstestschicht
- schnelle versus langsame Tests
- Umgebungsabhängigkeit
- Parallelisierung
- Quality Gates
- Diagnosefähigkeit

Praktisches Integrationsszenario

- Systemlandschaft analysieren
- Integrationsrisiken identifizieren
- Teststufen verteilen
- Doubles auswählen
- End-to-End-Pfade bestimmen
- Fehlerfälle entwickeln
- Testarchitektur begründen

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 Systemlandschaft mit ihren Schnittstellen und Abhängigkeiten kann als Grundlage für die Testarchitektur dienen.

04 ZIELGRUPPE

Für wen ist der Kurs gedacht?

Tester, Quality Engineers, Testautomatisierer, Entwickler und technische Testmanager, die komplexe Systemintegrationen absichern müssen.

Voraussetzungen

Grundkenntnisse in Softwaretest sowie grundlegendes Verständnis von APIs und Systemarchitekturen.

05 LERNZIELE

Was Sie nach dem Kurs können

Die Teilnehmer können Integrations-, System- und End-to-End-Tests klar voneinander abgrenzen, Integrationsrisiken analysieren und eine effiziente Testarchitektur für komplexe Systeme entwickeln.

06 FAQ

Häufige Fragen

Warum nicht einfach alles End-to-End testen?

End-to-End-Tests sind wertvoll, aber langsam, teuer und häufig schwer zu diagnostizieren. Viele Fehler lassen sich auf niedrigeren Teststufen schneller und gezielter erkennen.

Werden Microservices behandelt?

Ja, wo verteilte Architekturen, Contracts, asynchrone Kommunikation und Service Virtualization relevant sind. Der Kurs ist jedoch nicht auf Microservices beschränkt.

Überschneidet sich der Kurs mit API Testing?

API Testing ist ein wichtiger Baustein der Integration. Der API-Kurs untersucht Schnittstellen selbst; hier geht es um das Zusammenspiel mehrerer Komponenten und Systeme und die Architektur der entsprechenden Tests.

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

<https://www.uc-it.de/coaching/integration-system-e2e/>