

Testen von Legacy-Systemen

Qualität in gewachsenen Systemen auch ohne perfekte Testbasis beherrschbar machen

01 INTRO

Legacy-Systeme stellen Quality Engineering vor besondere Herausforderungen: unvollständige Dokumentation, starke Kopplung, unbekannte Abhängigkeiten, schwer reproduzierbare Datenzustände und wenige automatisierte Tests. Gleichzeitig sind gerade diese Systeme häufig geschäftskritisch.

Der Kurs behandelt Strategien, um ein bestehendes System schrittweise testbar und sein Verhalten nachvollziehbar zu machen. Der Schwerpunkt liegt auf Charakterisierung, Risikoanalyse, Regression, Schnittstellen und dem Aufbau eines belastbaren Sicherheitsnetzes für weitere Änderungen.

DAUER

3 Tage

FORMAT

Präsenz · Live Online · Inhouse

VORKENNTNISSE

Praktische Erfahrung im Softwaretest, in der Entwicklung oder im Betrieb bestehender Systeme

02 TRAININGSPROGRAMM

Inhalte

Was macht ein System zu Legacy?

- Alter versus Veränderbarkeit
- fehlendes Wissen
- technische Schulden
- Kopplung
- kritische Abhängigkeiten
- geschäftliche Bedeutung
- fehlende Tests als Risiko

Das System erschließen

- vorhandene Dokumentation
- Code und Architektur
- Logs
- Datenbanken
- Schnittstellen
- Nutzerwissen
- Betriebserfahrung
- tatsächliches Verhalten als Informationsquelle

Risiken identifizieren

- kritische Geschäftsprozesse
- häufig geänderte Bereiche
- Fehlerhistorie
- technische Hotspots
- externe Abhängigkeiten
- Daten
- Betrieb
- Priorisierung

Verhalten charakterisieren

- Characterization Tests
- Golden Master
- Snapshot-basierte Ansätze
- bestehendes Verhalten sichern
- unbekannte Nebenwirkungen
- Verhalten versus fachliche Korrektheit

Regression aufbauen

- kritische Pfade
- vorhandene Defects
- Produktionsszenarien
- Regressionstestbestand
- Priorisierung
- schrittweiser Ausbau
- Kosten und Nutzen

Testbare Schnittstellen finden

- APIs
- Datenbanken
- Dateien
- Messaging
- interne Schnittstellen
- UI als letzte Möglichkeit
- geeignete Testebenen auswählen

Abhängigkeiten kontrollieren

- externe Systeme
- Test Doubles
- Service Virtualization
- Stubs
- Simulatoren
- instabile Umgebungen
- Grenzen der Isolation

Testdaten in Legacy-Systemen

- komplexe Datenzustände
- produktionsnahe Daten
- Datenschutz
- anonymisierte Daten
- Setup
- Cleanup
- reproduzierbare Ausgangszustände

Automatisierung schrittweise einführen

- geeignete Einstiegspunkte
- API vor UI
- stabile Regression
- Wartbarkeit
- vorhandene Frameworks
- technische Schulden im Testcode
- realistische Automatisierungsziele

Änderungen sicher begleiten

- Impact Analysis
- betroffene Komponenten
- Regression
- Feature Flags
- Parallelbetrieb
- Monitoring
- Rollback
- kleine kontrollierbare Schritte

Wissen und Beobachtbarkeit verbessern

- Logs
- technische Metriken
- Tracing
- Dokumentation aus Tests
- bekannte Verhaltensweisen
- Fehlerwissen
- kontinuierlicher Wissensaufbau

Praktische Legacy-Teststrategie

- unbekanntes System analysieren
- Risiken bestimmen
- kritische Pfade identifizieren
- Charakterisierung entwickeln
- Regression priorisieren
- Automatisierungsstrategie planen
- Sicherheitsnetz schrittweise aufbauen

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 Teststrategie verwendet werden.

04 ZIELGRUPPE

Für wen ist der Kurs gedacht?

Tester, Quality Engineers, Entwickler und technische Verantwortliche, die bestehende, schwer testbare oder unzureichend dokumentierte Systeme absichern müssen.

Voraussetzungen

Praktische Erfahrung mit Softwareentwicklung oder Softwaretest.

05 LERNZIELE

Was Sie nach dem Kurs können

Die Teilnehmer können Legacy-Systeme schrittweise erschließen, kritische Risiken und Verhaltensweisen identifizieren und ein pragmatisches Regressionstest- und Qualitätssicherungsnetz aufbauen.

06 FAQ

Häufige Fragen

Muss der Quellcode verfügbar sein?

Nein. Viele Verfahren funktionieren auch auf Schnittstellen-, Daten- oder Systemebene. Verfügbarer Quellcode erweitert allerdings die Analysemöglichkeiten.

Bedeutet Characterization Testing, fehlerhaftes Verhalten festzuschreiben?

Characterization Tests dokumentieren zunächst beobachtetes Verhalten. Ob dieses fachlich korrekt ist, muss separat beurteilt werden.

Was unterscheidet den Kurs von Modernisierung und Migration?

Hier geht es um den laufenden Umgang mit einem bestehenden Legacy-System. Der Migrationskurs konzentriert sich auf die besondere Qualitätssicherung während einer geplanten Modernisierung oder Migration.

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

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