

Softwarequalität und Quality Engineering – Grundlagen

Softwarequalität systematisch verstehen, planen und absichern

01 INTRO

Softwarequalität entsteht nicht erst beim Testen eines fertigen Systems. Anforderungen, Architektur, Entwicklung, Test, Betrieb und die Zusammenarbeit im Projekt beeinflussen gemeinsam, ob Software ihre fachlichen und technischen Qualitätsziele erreicht. Quality Engineering betrachtet Qualität deshalb als durchgängige Engineering-Aufgabe.

Der Kurs vermittelt die grundlegenden Konzepte moderner Softwarequalität und des Softwaretests. Die Teilnehmer lernen Qualitätsmerkmale, Teststufen und Testarten kennen, entwickeln einfache Testfälle und ordnen präventive und analytische Maßnahmen in einen durchgängigen Qualitätsprozess ein.

DAUER

3 Tage

FORMAT

Präsenz · Live Online · Inhouse

VORKENNTNISSE

Keine Testkenntnisse nötig; Grundverständnis von Softwareentwicklung ist hilfreich

02 TRAININGSPROGRAMM

Inhalte

Softwarequalität und Quality Engineering

- Qualität aus unterschiedlichen Perspektiven
- Produkt- und Prozessqualität
- Qualitätssicherung und Qualitätskontrolle
- Quality Assurance, Testing und Quality Engineering
- präventive und analytische Maßnahmen
- Verantwortung für Qualität

Qualitätsmerkmale von Software

- funktionale Eignung
- Zuverlässigkeit
- Benutzbarkeit
- Performance
- Sicherheit
- Wartbarkeit
- Kompatibilität
- Qualitätsmodelle als Orientierung

Grundlagen des Softwaretests

- Ziele des Testens
- Fehlerzustand, Fehlerwirkung und Fehlerursache
- Testen und Debugging
- grundlegende Testprinzipien
- Grenzen des Testens
- risikoorientiertes Denken

Qualität im Entwicklungslebenszyklus

- sequenzielle und iterative Entwicklung
- agile Entwicklung
- Shift Left und Shift Right
- frühe Qualitätssicherung
- Reviews
- statische und dynamische Verfahren
- Feedbackzyklen

Teststufen

- Komponententest
- Integrationstest
- Systemtest
- Abnahmetest
- Verantwortlichkeiten
- Ziele der einzelnen Stufen
- sinnvolle Übergänge

Testarten

- funktionale Tests
- nichtfunktionale Tests
- Regression
- Smoke und Sanity Tests
- exploratives Testen
- positive und negative Tests
- Auswahl nach Risiko und Ziel

Anforderungen als Testbasis

- testbare Anforderungen
- Akzeptanzkriterien
- Mehrdeutigkeiten
- Vollständigkeit
- Konsistenz
- Traceability
- frühes QA-Feedback

Grundlagen des Testdesigns

- Testbedingungen
- Testfälle
- Äquivalenzklassen
- Grenzwerte
- Entscheidungstabellen
- Zustände
- erfahrungsbasierte Verfahren

Testdurchführung und Fehlerbehandlung

- Testvorbereitung
- Testdaten
- Testumgebung
- erwartete und tatsächliche Ergebnisse
- Defects
- gute Fehlerberichte
- Retest und Regression

Testautomatisierung einordnen

- Ziele der Automatisierung
- geeignete Tests
- Testpyramide
- API- und UI-Tests
- Nutzen und Kosten
- Wartbarkeit
- Grenzen der Automatisierung

Qualität messen und kommunizieren

- Testfortschritt
- Coverage
- Defect-Metriken
- Qualitätsindikatoren
- Aussagekraft und Grenzen von Metriken
- Reporting
- Exit-Kriterien

Praktisches Quality-Engineering-Szenario

- Qualitätsziele bestimmen
- Risiken identifizieren
- Testbasis analysieren
- Tests entwerfen
- Maßnahmen priorisieren
- Ergebnisse bewerten
- Qualitätsvorgehen 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

Eigene Systeme, Anforderungen und Qualitätsziele können in die Übungen einbezogen werden.

04 ZIELGRUPPE

Für wen ist der Kurs gedacht?

Tester, Entwickler, Business Analysts, Projektmitarbeiter und Quereinsteiger, die Softwarequalität und Softwaretest systematisch verstehen möchten.

Voraussetzungen

Keine besonderen Vorkenntnisse erforderlich.

05 LERNZIELE

Was Sie nach dem Kurs können

Nach dem Kurs können die Teilnehmer Softwarequalität als durchgängige Engineering-Aufgabe einordnen, wichtige Teststufen und Testarten unterscheiden, einfache Tests systematisch entwickeln und geeignete Qualitätssicherungsmaßnahmen auswählen.

06 FAQ

Häufige Fragen

Ist der Kurs eine ISTQB-Prüfungsvorbereitung?

Nein. Zahlreiche grundlegende Konzepte überschneiden sich naturgemäß mit etablierten Teststandards. Der Kurs ist jedoch praxisorientiert und nicht auf das Bestehen einer Zertifizierungsprüfung ausgerichtet.

Ist Testautomatisierung Bestandteil des Kurses?

Sie wird eingeordnet. Die technische Umsetzung ist Gegenstand der spezialisierten Kurse zur Testautomatisierung.

Ist der Kurs auch für Entwickler geeignet?

Ja. Quality Engineering betrachtet Softwarequalität ausdrücklich nicht als alleinige Aufgabe einer Testabteilung.

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

<https://www.uc-it.de/coaching/softwarequalitaet-grundlagen/>