

BDD und Testautomatisierung mit Cucumber und Gherkin

Fachliche Beispiele von der Anforderung bis zum automatisierten Test nutzen

01 INTRO

Behavior Driven Development nutzt konkrete Beispiele, um fachliche Anforderungen gemeinsam zu präzisieren und ausführbares Verhalten zu beschreiben. Gherkin stellt dafür eine verständliche Sprache bereit; Cucumber verbindet diese Szenarien mit automatisiertem Testcode.

Der Kurs behandelt BDD nicht als bloße Schreibweise für Tests, sondern als kollaborativen Entwicklungsansatz. Die Teilnehmer entwickeln Beispiele aus fachlichen Regeln, formulieren wartbare Szenarien und verbinden sie mit einer technischen Automatisierung.

DAUER

3 Tage

FORMAT

Präsenz · Live Online · Inhouse

VORKENNTNISSE

Grundlagen des Softwaretests und grundlegende Programmiererfahrung

02 TRAININGSPROGRAMM

Inhalte

Behavior Driven Development

- Ursprung und Ziele
- gemeinsames Verständnis
- Beispiele statt abstrakter Anforderungen
- Discovery, Formulation und Automation
- BDD versus TDD
- BDD versus Testskripte

Fachliche Regeln und Beispiele

- Anforderungen analysieren
- Geschäftsregeln
- konkrete Beispiele
- Gegenbeispiele
- offene Fragen
- Example Mapping
- Three Amigos

Gherkin

- Feature
- Scenario
- Given
- When
- Then
- And und But
- fachliche Sprache
- deklarativ versus imperativ

Gute Szenarien entwickeln

- eine fachliche Aussage pro Szenario
- verständliche Namen
- relevante Details
- technische Details vermeiden
- Unabhängigkeit
- Lesbarkeit
- Anti-Patterns

Scenario Outline und Daten

- Examples
- Parametrisierung
- Tabellen
- Doc Strings
- mehrere Varianten
- Grenzen datengetriebener Szenarien

Cucumber-Architektur

- Feature Files
- Step Definitions
- Glue Code
- Runner
- Hooks
- Tags
- Ausführung

Step Definitions implementieren

- Matching
- Parameter
- Expressions
- Wiederverwendung
- fachliche Schritte
- technische Implementierung
- Duplikate vermeiden

Automatisierung anbinden

- API
- Web UI
- Service Layer
- Datenbanken
- vorhandene Testframeworks
- BDD-Schicht versus Automatisierungsschicht
- Verantwortlichkeiten trennen

State und Testdaten

- Scenario Context
- Setup
- Datenbereitstellung
- Isolation
- Cleanup
- Hooks
- Abhängigkeiten vermeiden

Größere BDD-Suites strukturieren

- Feature-Organisation
- Tags
- fachliche Domänen
- gemeinsame Schritte
- Wartbarkeit
- Step Explosion
- lebende Dokumentation

BDD in CI/CD

- automatisierte Ausführung
- Reports
- Tags
- Regression
- Quality Gates
- fehlgeschlagene Szenarien
- Dokumentation aktuell halten

Durchgängiges BDD-Projekt

- fachliche Regel analysieren
- Beispiele entwickeln
- Gherkin formulieren
- Steps implementieren
- Automatisierung anbinden
- Szenarien reviewen
- CI-Ausführung vorbereiten

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 und einer Cucumber-Implementierung

Inhouse-Schulungen

Eigene fachliche Regeln und Anforderungen können als Grundlage für die Beispiele und Szenarien dienen.

04 ZIELGRUPPE

Für wen ist der Kurs gedacht?

Tester, Entwickler, Business Analysts, Product Owner und Quality Engineers, die Anforderungen, Beispiele und Testautomatisierung mit BDD verbinden möchten.

Voraussetzungen

Grundlagen im Softwaretest und grundlegende Programmiererfahrung.

05 LERNZIELE

Was Sie nach dem Kurs können

Die Teilnehmer können BDD fachlich einordnen, Anforderungen mit Beispielen präzisieren, wartbare Gherkin-Szenarien formulieren und diese mit Cucumber automatisieren.

06 FAQ

Häufige Fragen

Ist BDD einfach Testautomatisierung in Given-When-Then?

Nein. Automatisierung ist nur ein Teil. Der eigentliche Nutzen entsteht durch die gemeinsame Konkretisierung fachlichen Verhaltens anhand von Beispielen.

Welche Sprache wird für Cucumber verwendet?

Die technische Umsetzung kann beispielsweise mit Java oder einer anderen zum Teilnehmerumfeld passenden Cucumber-Implementierung erfolgen.

Gehört Gherkin in jeden automatisierten Test?

Nein. Gherkin ist sinnvoll, wenn fachliche Beispiele und gemeinsame Verständlichkeit einen Mehrwert liefern.

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

<https://www.uc-it.de/coaching/bdd-cucumber/>