

Technische Praktiken: TDD-2



Im Kurs 'Technische Praktiken: TDD-2' setzen Sie Ihren Weg zur Perfektionierung agiler technischer Praktiken auf fortgeschrittenem Niveau fort, bauen auf den Grundlagen aus dem Grundkurs auf, lernen etwas über Outside-In Softwaredesign, üben fortgeschrittenere Formen des Pair Programming und wenden eine Mischung aus TDD-Techniken an.

Bitte beachten Sie: Dies ist ein Kurs für Fortgeschrittene, daher ist unser Kurs 'Technische Praktiken: TDD-1' oder ein ähnlicher Kurs Voraussetzung für die erfolgreiche Kursteilnahme!

Der Kurs richtet sich an Softwareentwickler, die ihre Kenntnisse über effektive technische Praktiken erweitern oder vertiefen möchten. Die Kurse können für die Demos und Übungen in verschiedenen Programmiersprachen angeboten werden – kontaktieren Sie uns für weitere Informationen. Bitte beachten: Es handelt sich um sehr praxisorientierte Kurse, bitte einen Computer mit der Entwicklungsumgebung Ihrer Wahl bereithalten.

Die Schulungsunterlagen sind auf Englisch.

Lernziele

- Vertiefung des Verständnisses moderner Konzepte agiler technischer Praktiken
- Beherrschung der Praktiken Testgetriebene Entwicklung (TDD), Paarprogrammierung, iterative und inkrementelle Entwicklung und emergentes Design
- Kenntnisse zur Erstellung flexibler, leicht modifizierbarer und wartbarer Codes
- praktische live Übungen zur Anwendung und Integration der im Unterricht erworbenen Fähigkeiten

Kursorganisation

Der Kurs mit einer Gesamtkontaktzeit von 16 Stunden wird in Präsenzform oder im interaktiven Online-Modus durchgeführt. Der Kurs ist in verschiedene Module unterteilt (keines länger als zwei Stunden), mit kurzen Pausen nach Bedarf und ausreichend langen Pausen zwischen den Sessions. Die genauen Zeiten für die Pausen werden zu Beginn des Kurses in der Gruppe vereinbart.

Vor- und Nachbereitungsaktivitäten sind Teil der Schulung und werden über unsere interaktive Online-Lernplattform in verschiedenen Formaten (Video, Text, Quiz, Arbeitsblätter, weiterführende Literatur usw.) bereitgestellt.

Kursinhalte

- Outside-In und Inside-Out Design
- Test-Doubles – Mocks, Stubs, Fakes
- Richtlinien für das Test-Case Design für Outside-In Design
- Ausgewogenes Outside-In und Inside-Out Design
- Outside-In Design mithilfe von Mikrotests
- Überprüfung der Softwaredesignprinzipien
- Häufige Code-Mängel und Abhilfemaßnahmen
- Festlegen einer Test-Baseline: Golden-Master Ansatz
- Festlegen einer Test-Baseline: Read-by-Refactoring Ansatz
- Identifizieren von Seams im Code
- Schreiben von Characterisation Tests
- Aufheben von Abhängigkeiten, um Mikrotesting zu ermöglichen
- Praktikum: