

Technische Praktiken: TDD-1



In diesem Kurs geht es um die grundlegenden Konzepte agiler technischer Praktiken, mit besonderem Schwerpunkt auf den Techniken des eXtreme Programming. Der Kurs vermittelt die notwendigen Grundlagen, damit das Team die Qualität seiner Arbeit langfristig nachhaltig verbessern kann, indem es flexiblen, leicht modifizierbaren und gut zu wartenden Code erstellt. Im Kurs wechseln sich theoretische Präsentationen mit praktischen Übungen zur Anwendung und Integration des Gelernten ab.

Lernen Sie agile technische Praktiken in einem kompakten, intensiven, praxisorientierten Format kennen. Unter Anleitung erfahrener Praktiker lernen Sie die Grundlagen von Test-Driven Development (TDD), Pair Programming, iterativer und inkrementeller Entwicklung, Emergent Design, der Anwendung robuster Software-Design-Prinzipien und der Arbeit mit Frameworks und Libraries.

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

- Verständnis der grundlegenden Konzepte agiler technischer Praktiken
- Kenntnisse der Grundlagen von Test-Driven Entwicklung (TDD), Paarprogrammierung, iterativer und inkrementeller Entwicklung und emergentem Design
- Kenntnisse für die Erstellung von flexiblem, leicht modifizierbarem und gut zu wartendem Code
- praktische 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

- TDD als Softwaredesigntechnik und Testwerkzeug
- Wo TDD in das Gesamtbild der Softwaretests passt
- Geschichte von TDD und wann es eingesetzt wird
- Konzepte von Unit-Tests und Mikrotests
- Codeisolierung
- Detroit School oder klassisches TDD-Praktikum
- Inside-Out-Design
- Tests-Coverage-Metriken – Bedeutung und Verwendung
- Merkmale effektiver Test Cases
- Überlegungen zur Organisation und Benennung von Testsuiten
- Test-Driving Code, der mit Frameworks und Bibliotheken interagiert
- IDE- und VCS-Integration
- Wert der Zusammenarbeit
- Entwicklung mit künstlicher Intelligenz als Pairing Partner
- Grundlagen des Pair Programming, Anti-Patterns, verschiedene Stile
- Testsuiten als Designdokumentation
- Vorab- und emergentes Design und wie man sie ausbalanciert
- Prinzipien des Software-Designs
- Verhaltensorientierte vs. implementierungsorientierte Tests
- Design durch Mikrotests – Praktikum
- Branching-Strategien