

## **AR200 iSAQB Certified Professional for Software Architecture - Advanced Level (CPSA-A): Domain-Driven-Design**

### **Kurzbeschreibung:**

Teilnehmer erhalten eine praxisnahe Einführung in Domain-driven Design für Softwarearchitektur. Vermittelt wird, wie fachliches Wissen in eine Ubiquitous Language übertragen und mit Techniken wie Interviews, Domain Storytelling und Event Storming modelliert wird. Behandelt werden Bounded Contexts als Grundlage für sinnvolle vertikale Schnitte und die Entwicklung von Microservices, die Fachlichkeit wirksam abbilden.

### **Zielgruppe:**

Der Kurs **AR200 iSAQB Certified Professional for Software Architecture - Advanced Level (CPSA-A): Domain-Driven-Design** richtet sich an:

- Software-Architekten und Entwickler

### **Voraussetzungen:**

Die Zertifizierung im Advanced Level setzt den vorherigen Abschluss des Foundation Levels voraus.

### **Sonstiges:**

**Dauer:** 3 Tage

**Preis:** 2120 Euro plus Mwst.

### **Ziele:**

In diesem Kurs **AR200 iSAQB Certified Professional for Software Architecture - Advanced Level (CPSA-A): Domain-Driven-Design** lernen Sie, wie eine Fachsprache entwickelt, die DDD-Muster eingesetzt und die Verbindung zu anderen Anwendungen hergestellt wird, um eine an der Fachdomäne orientierte Anwendung im Kontext zu entwerfen. Im Workshop **AR200 iSAQB Certified Professional for Software Architecture - Advanced Level (CPSA-A): Domain-Driven-Design** wechseln sich Vorträge, Diskussion und Übungen ab. So kann das erlernte Wissen gleich in der Praxis vertieft werden. Gemeinsam üben wir an einer Beispielanwendung die einzelnen Schritte hin zu einer anwendungsorientierten und qualitativ hochwertigen Architektur. Sie werden erleben, wie einfach ein Entwurf sein kann, wenn man sich auf die Fachdomäne und die architektonischen Leitplanken von Domain-Driven Design einlässt!

## Inhalte/Agenda:

- ◆ Überblick: Was ist Domain-Driven Design?
- ◆
- ◆ Die Domäne kennenlernen: Event Storming
- ◆
- ◆ Strategisches Design: Die Domäne aufteilen, Modelle schneiden und voneinander abgrenzen
- ◆
- ◆ Ubiquitous Language
- ◆
- ◆ Der Weg zum Modell
- ◆
- ◆ Taktisches Design – Das Modell in Software ausdrücken