

AR100-WS Software Engineering 2.0 Warum Entwicklung allein nicht mehr reicht

Kurzbeschreibung:

Die Zeiten, in denen Softwareentwicklung nur aus Coding bestand, sind vorbei. Erfolgreiche Softwareprojekte brauchen heute mehr als reines Programmieren – sie erfordern ein ganzheitliches Verständnis für Architektur, Security, Testing und Requirements Engineering. Unternehmen stehen vor der Herausforderung, ihre Teams mit den richtigen Skills auszustatten, um technische Schulden, Sicherheitsrisiken und Qualitätsprobleme zu vermeiden.

In unserem Web-Seminar **AR100-WS Software Engineering 2.0 – Warum Entwicklung allein nicht mehr reicht** zeigen wir, warum der Wandel vom klassischen Entwickler zum Software Engineer für den langfristigen Projekterfolg entscheidend ist – und wie Organisationen und Fachkräfte sich optimal darauf vorbereiten.

Zielgruppe:

Das Web-Seminar **AR100-WS Software Engineering 2.0 – Warum Entwicklung allein nicht mehr reicht** richtet sich vor allem an:

- Software Entwickler
- Software Engineer
- Software Developer
- Requirements Engineer
- Software Architect
- Software Tester
- Führungskräfte
- IT-Leiter
- Teamleiter
- HR-Leiter
- Qualitätsmanager

Voraussetzungen:

Grundlegende IT-Kenntnisse

Sonstiges:

Dauer: 1 Tage

Preis: 0 Euro plus Mwst.

Ziele:

Ziel des Web-Seminars **AR100-WS Software Engineering 2.0 – Warum Entwicklung allein nicht mehr reicht** ist es, den Teilnehmern zu helfen, ihre Fähigkeiten und Kenntnisse zu erweitern, um besser auf die Herausforderungen der modernen Softwareentwicklung vorbereitet zu sein.

Inhalte/Agenda:

- **Historische Betrachtung des Software Engineering**

- - ◆ Entwicklungsmodelle & Anwendungen in der Vergangenheit
 - ◆ Wie hat sich das Verständnis für Qualität verändert?

- - ◆

- **Die Rolle des Software Engineers heute**

- - ◆ Die Komplexität von Software
 - ◆ Der Einfluss von agiler Entwicklung, DevOps und Security
 - ◆ Die Relevanz von domainübergreifender Ausrichtung
 - ◆ Moderne Testing-Ansätze durch Testautomatisierung

- - ◆

- **Ein Blick in die Zukunft: Wie verändert sich Software Engineering in den nächsten Jahren?**

- - ◆ Abstraktion in der Entwicklung: Von Low-Level über Hochsprachen zu Modellen
 - ◆ Wie KI bei dem Verständnis von Anforderungen und dem Modellieren hilft
 - ◆ Welche Skills bleiben relevant und was bedeutet das für Sie?

- - ◆

- **Q & A Session**