

## **UML-Praxis-Workshop: Praktischer Einsatz modellbasierter Softwareentwicklung für Embedded- und Echtzeit-Systeme - Live-Online-Training**

### **Ziele - Ihr Nutzen**

Mit dem im UML-Praxis-Workshop erworbenen Wissen sind Sie in der Lage, Embedded- und Echtzeitsoftware mit der UML zu entwickeln und bestehende zu warten, zu erweitern und zu optimieren.

Sie können Vorgaben (UML Style-Guide) für den Einsatz der UML in Ihrem Unternehmen entwickeln und treffen bei Bedarf eine fundierte Toolentscheidung.

Sie erkennen den Gewinn an Qualität im Software-Projekt durch die Anwendung von UML-Modellen und -Diagrammen basierend auf einer kontinuierlichen Pflege.

### **Teilnehmer**

Der UML-Praxis-Workshop richtet sich an Software-Entwickler, Software-Architekten, Software-Entwicklungsleiter, Systemarchitekten.

### **Voraussetzungen**

Projekterfahrung mit Embedded-Softwareentwicklung und Basiskenntnisse der UML-Notationen.

## **Live Online Training**

\* Preis je Teilnehmer, in Euro zzgl. USt.

Anmeldecode: L-UMLPRAX

## **Präsenz-Training - Deutsch**

### **Dauer**

3 Tage

## **UML-Praxis-Workshop: Praktischer Einsatz modellbasierter Softwareentwicklung für Embedded- und Echtzeit-Systeme - Live-Online-Training**

### **Inhalt**

#### **Themeneinleitung**

- Nutzen der UML
- Kontext für den erfolgreichen Einsatz der UML (Unified Modeling Language)
- UML Best Practice: UML-Notation, praxisgerechter Entwicklungsprozess und erfolgreiche Entwicklungsmethodik
- Qualitätsverbesserung im gesamten Softwaredesign durch den strukturierten UML-Einsatz im Softwareentwurf
- Möglichkeiten der Effektivitätssteigerung und einhergehenden Kostenreduktion im Projekt mit dem Einsatz der UML
- Programmiersprache, Tools und Projekt
- Checkliste für das Aufsetzen eines Pilotprojektes

#### **Logisch aufeinander aufbauender Einsatz ausgewählter UML-Diagramme**

- Modellbasiertes Software Engineering
- Model Driven Architecture (MDA)
- Model Driven Development (MDD)

- Skalierung der Diagramme und Notationen
- Darstellungsformen und Sichten in den einzelnen Softwareanalyse-Aktivitäten:
- Anforderungsanalyse, Architekturanalyse, Klassen-/Modulanalyse
- Darstellungsformen und Sichten in den einzelnen Softwaredesign-Aktivitäten:
- Architekturdesign, Layer-/Subsystemdesign und Klassen-/Moduldesign
- Modellierung von speziellen Aspekten von Embedded- und Echtzeitsoftware
- Entwicklungsframework
- Die passende Abstraktion der Modellierung
- Nutzenaspekte
- Mögliche Alternativen und deren Bewertungen
- Berücksichtigung von Betriebssystem-Mechanismen
- Einbindung von Legacy-Software und Bibliotheken
- Praxistipps: Entscheidungshilfen bei der Sichten- und Diagrammauswahl
- Übung: Auf Basis von ausformulierten Anforderungen modellieren Sie mit der UML wichtige Sichten für eine konkrete Embedded-Software, erkennen deren Nutzen und bekommen bei einem Review ein direktes Feedback dazu.

#### **UML Case Tools**

- Funktionsweise
- Klassifizierung
- Automatische Programmcode-Generierung
- Synchronisation zwischen Modell und Programmcode
- Austausch von UML-Modellen
- Integration mit anderen Tools
- Checkliste mit Anforderungen zur Toolauswahl
- Aktuelle Tool-Übersicht
- Übung: Sie generieren C oder C++ Programmcode, führen diesen auf einem Embedded-Target aus und synchronisieren im Editor geänderten Programmcode ins Modell zurück.

#### **Projektorganisation im UML Case Tool**

- Organisation/ Strukturierung der UML-Diagramme
- Organisation/ Strukturierung der UML-Elemente
- Praxistipp: UML-Modell-Template

#### **UML Style-Guide**

- Nutzendarstellung
- Strukturierung
- Konkrete Inhalte
- Praxistipp: Style-Guide-Template

#### **Übersicht Embedded-Software-Test**

- Bezug zum Software-Modell und zur Software-Implementierung
- Testebenen, gezeigt im V-Modell
- Verfahren zum Softwaretest
- Testprozess
- Praxistipps mit speziellen Testaspekten für objektorientierte Software

#### **Praktische Übungen im UML-Praxis-Workshop**

- Durchgängige Modellierung der Embedded-Software einer realen Elektromotor-Steuerung, von den SW-Anforderungen bis hin zum SW-Design mit der UML.
- Dabei durchlaufen Sie die Entwicklungsschritte der Softwareanalyse und des Softwaredesigns.
- Sie führen die Übung wahlweise mit dem professionellen Modellierungstool Enterprise Architect der Firma Sparx Systems oder mit Papier und Bleistift durch.
- Sie haben die Möglichkeit, wahlweise in C oder C++, die Codegenerierung und Synchronisation mit Modellierungstool Enterprise Architect der Firma Sparx Systems durchzuführen.
- Den generierten Programmcode führen Sie auf einem Embedded-Target aus.
- Dazu nutzen Sie das Arm Keil MDK (Microcontroller Development Kit) und mit einer realen Hardware, basierend auf einem Arm Cortex® M3 Mikrocontroller.

#### **MicroConsult Plus**

- Kopien der Übungsverzeichnisse und Lösungsbeispiele für alle Übungsaufgaben stellen wir den Teilnehmern zum Download bereit.
- Sie bekommen das komplette UML-Softwaremodell der Elektromotor-Steuerung.
- Sie erhalten zudem eine Tool- und Software-Komponentenübersicht für die Entwicklung von

Embedded-Software.