

Requirements Engineering Schulung: Requirements Engineering und Requirements Management für Embedded-Systeme - Live-Online-Training

Mit der steigenden Komplexität von Embedded-Systemen wächst ebenso die Bedeutung von Anforderungen und deren Verwaltung.

Ziele - Ihr Nutzen

Entwickeln und dokumentieren Sie qualitativ hochwertige Anforderungen und dazu passende Abnahmekriterien und treffen Sie fundierte Tool-Entscheidungen zur Verwaltung von Anforderungen.

Mit dem in der Requirements Engineering Schulung erworbenen Wissen sind Sie in der Lage, einen Requirements-Prozess in Ihrer Firma einzuführen, zu bewerten, zu optimieren, zu verstehen und ihn zu leben.

Teilnehmer

Die Requirements Engineering Schulung richtet sich an Produktmanager, Marketing Manager, Vertriebsleiter, Projektleiter, Anforderungsanalysten, Systemarchitekten, Hardware-Architekten, Software-Architekten, Hardware-Entwickler, Software-Entwickler, Leiter Qualitätssicherung, Ingenieure Qualitätssicherung, Test-Manager, Testingenieure, Product Owner, Scrum Master, Scrum-Team

Voraussetzungen

Keine; Projekterfahrungen sind von Vorteil.

Live Online Training

24.03. – 26.03.2025 1.950,00 €3 Tage

17.11. – 19.11.2025 1.950,00 €3 Tage

* Preis je Teilnehmer, in Euro zzgl. USt.

Anmeldecode: L-REQ-ENG

Präsenz-Training - Deutsch

Termin **Dauer**

07.07. – 09.07.2025 3 Tage

09.03. – 11.03.2026 3 Tage

Requirements Engineering Schulung: Requirements Engineering und Requirements Management für Embedded-Systeme - Live-Online-Training

Inhalt

Themeneinleitung

- Notwendigkeit und Nutzen von Anforderungen
- Anforderungen und Arten und Kategorien
- Funktionale und nicht-funktionale Anforderungen
- Zusicherungen als Anforderungen
- Produkt-, Produktgruppen-, Projekt- und Unternehmensanforderungen

- Trennung zwischen Anforderungen und deren Umsetzung
- Stakeholder als Anforderungsquelle
- Aussehen einer Anforderung und deren Bezug zu Abnahmekriterien
- Differenzierung: Feature und Anforderung
- Qualitätseigenschaften von Anforderungen und Anforderungsdokumenten
- Differenzierung: Requirements Engineering, Requirements Development, Requirements Management
- Umgang mit Projektrisiken (Risikomanagement)
- Systemverständnis
- Notwendigkeit des Entwicklungsprozesses
- Prozessqualität, innere und äußere Produktqualität

Entwicklungsprozess: Klassisch

- Anforderungsanalyse
- Vorgehen vor der Anforderungsidentifikation
- Vorgehen mit und nach der Anforderungsidentifikation
- Artefakte: Systemidee, Anforderungsdokument (Lastenheft), Anforderungsspezifikation (Pflichtenheft), Machbarkeitsstudie, Testfälle, Entwicklungsauftrag
- Rollen: Produktmanager, Projektleiter, Anforderungsanalyst, Testteam
- Stakeholder: Analyse, Interview, Ableitung von Anforderungen
- Qualitätssicherung und Freigabe der Anforderungen
- Umgang mit Änderungen von Anforderungen (Change Management und Impact Analyse)
- Ebenen von Anforderungen für Embedded-Systeme (im V-Modell)
- Kunden-, System-, Subsystem-, Hardware-, Software- und Konstruktionsanforderungen
- Anforderungsabstimmung zwischen den Domänen
- Schnittstellenanforderungen
- N-Augen Prinzip
- Prozess-Reifegradmodelle: CMMI® (Capability Maturity Model Integration) und SPiCE

Entwicklungsprozess: Agil

- V-Modell XT und agiles Vorgehen kombiniert
- Das Agile Manifest und seine Kernaussagen
- Scrum-Übersicht: Rollen, Meetings, Artefakte und Zusammenspiel
- Integration von Requirements Engineering in Scrum
- Anforderungen mit Scrum-Artefakten beschreiben
- Artefakte und Beispiele: Epic, User Story, Akzeptanzkriterium, Technical Story, Definition of Ready, Definition of Done
- Scrum-Vorgehen mit klassischen Artefakten adaptieren

Identifikation, Dokumentation und Verifikation von Requirements

- Textquelle versus modellbasierte Anforderungserfassung
- Aussehen von Anforderungen
- Attribute und deren Nutzen zu Anforderungen
- Referenz in Anforderungstexten zu Begriffsdefinition, Liste, Dokument, Standard, Norm
- Methoden und Praktiken zur Identifikation und Erfassung
- Aspekte: Sprache, Psychologie
- Transformationen und Regelwerk zu Rücktransformationen
- Anforderungstemplates anwenden
- Ableitung von Testfällen und Modellen
- Besonderheiten bei der Formulierung
- Verifikation und Qualitätsverbesserung
- Wiederverwendungskonzepte von Requirements
- Plattform-Anforderungen
- Besonderheiten bei nicht-funktionalen Requirements bzw. Qualitätsmerkmalen
- Anforderungsbeispiele
- Übung: Sie entwickeln Systemanforderungen für eine real existierende Elektromotorsteuerung. Auf Basis der vorgegebenen Systemarchitektur spezifizieren Sie entweder ein Subsystem, eine Hardware, eine Embedded-Software (Firmware) oder ein Konstruktionsteil mit weiteren domänenspezifischen Anforderungen. Im gemeinsamen Review wird bei Bedarf die Anforderungsqualität optimiert.

Modellierung von Anforderungen mit UML und SysML

- Grundgedanke objektorientierter Ansatz
- Relevante Diagramme und Notationen der UML (Unified Modeling Language): Use-Case-, Klassen-, Paket-,

Aktivitäts-, Zustandsfolge- und Sequenzdiagramm

- Relevante Diagramme und Notationen der SysML (Systems Modeling Language): Requirements-Diagramm
- Methode zum konsistenten Einsatz der Diagramme
- Kontextsicht, funktionale Anforderungssicht, Szenarien-Sichten, generische Verhaltenssicht
- Fehleranalyse
- Ausführbare Anforderungen und Simulationsmodelle
- Nutzen der unterschiedlichen Sichten
- Toolunterstützung
- Praxistipp: Eine mehrfach bewährte Vorgehensweise
- Vorführung: Mit dem Übungsbeispiel der Elektromotorsteuerung wird die Umsetzung im SysML-/UML-Modell mit einem Case-Tool gezeigt. (Das Modell erhält der Teilnehmer als Teil der Übungslösung)

Abnahmekriterien und Test

- Synonyme und Notwendigkeit
- Bezug zwischen Anforderungen und Abnahmekriterien
- Abnahmekriterien im Vertrags-/Auftragskontext
- Qualitätseigenschaften von Abnahmekriterien
- Arten von Abnahmekriterien
- Formulierungstemplates
- Ableitung und Dokumentation der Abnahmekriterien aus den Anforderungen
- Validieren und verifizieren
- Abnahmekriterien im V-Modell-Kontext
- White-Box-, Black-Box- und Gray-Box-Tests im V-Modell-Kontext
- Wichtige Verfahren zur Identifikation von Abnahmekriterien: Funktionsabdeckung, Äquivalenzklassenbildung und

Grenzwertanalyse

- Testszenarioszenarien
- Testdokumentation, Standards und Templates
- Richtlinien zum Schreiben von Abnahmekriterien
- Übung: Sie formulieren zu einem Teil der von Ihnen geschriebenen Anforderung die Abnahmekriterien. Im gemeinsamen Review wird bei Bedarf die Qualität optimiert.

Management von Requirements, Abnahmekriterien und mehr

- Versions- und Variantenverwaltung
- Baselines
- (Bidirektionale) Traceability und Verlinkung
- Praxisbeispiele
- Anforderungsdokumente, Standards und Templates
- Verteilte Entwicklung
- Erzeugung von Reports und Statistiken für die Projektleitung
- Anforderungabdeckung in Umsetzungsmodellen
- Wissenskonservierung
- Toolkategorien
- Anforderungen an Tools, Toolauswahl und -integration
- Text-Tools versus Datenbank-Systeme (ALM-/PLM-Tools)
- Toolintegration, ReqIF-Standard
- Praxistipp: Aktuelle Toolübersicht

Tipps zur Etablierung (von Veränderungen) im Unternehmen

- Veränderungen: Wahrheit, Auslöser, Initiatoren und Planung
- 5-Phasen-Modell
- Erfolgsfaktoren für Veränderungen im Unternehmen
- Umsetzungskontext: Ist- und Zielsituation
- Pilotprojekt richtig auswählen
- Konkretes Vorgehen zur Etablierung der Veränderung
- Prozessverbesserungsmodell
- Praxistipps zur Durchführung von Workshops

Praktische Übung in der Requirements Engineering Schulung

- In der durchgängigen Übung entwickeln und formulieren Sie Anforderungen und Abnahmekriterien zu einem real existierenden Embedded-System (Elektromotorsteuerung) auf unterschiedlichen Entwicklungsebenen/Entwicklungsdomänen und erfahren im Review ein direktes Feedback.

MicroConsult Plus:

- Sie erhalten von uns Ihre Übungsverzeichnisse und Lösungsbeispiele für alle Übungsaufgaben.

- Sie erhalten eine aktuelle Produktübersicht zu Requirements Management Tools und UML-/SysML-Tools
- Sie erhalten zudem Template-Dokumente zur spezifischen Anpassung und Verwendung.
- Sie bekommen hilfreiche Notationsübersichten für UML und SysML.