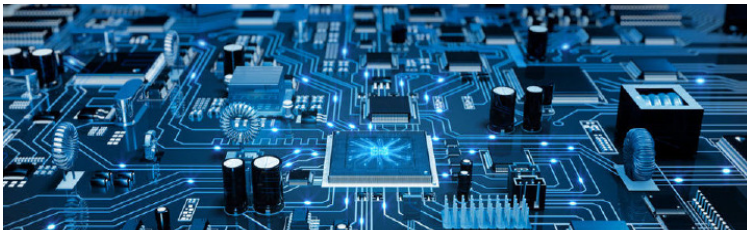


Embedded Software Testing

- Testautomatisierung für Embedded Software
- Robuste Systeme schaffen statt debuggen
- Wartbare, zukunftsträchtige und testbare Software-Architektur
- Die richtigen Tools auswählen
- Hands-On mit erfahrenen Experten



Beschreibung

Testautomatisierung ist Garant für Produktqualität. Testen bedeutet Fehler frühzeitig erkennen und beheben, bevor sie zum Projektrisiko werden. Während Testautomatisierung verbreitet Einsatz in der modernen Software-Entwicklung findet, stellt Embedded Software in dieser Hinsicht noch die Ausnahme dar. Weit verbreitet ist die Ansicht, automatisiertes Testen sei zu kompliziert und man komme ohne strukturiertes Testen besser zum Ziel. Das Gegenteil ist richtig! „Embedded Software Testing“ zeigt in Theorie und Praxis, wie mit einfachen Techniken eine maximale Qualitätssteigerung zu erzielen ist. Mittels Hands-On Übungen wird die vermittelte Theorie vor Ort erprobt.

Schwerpunkte:

Techniken: Komponententests, Integrationstests, Platzhalter; sowohl am Entwicklungs-PC als auch auf der Zielplattform

Testbare Architekturen: Komponentenmodell, Schichtmodell, Event-basierte Programmierung

Weitere Kursinhalte:

Testtheorie: Metriken, Abdeckung

Werkzeuge: Build-Tools, CI-Server

Best Practises: Codierungs-Standards, Code Reviews, Legacy Code

Die Kursinhalte überschneiden sich mit ISTQB Foundation Level, sind aber im Hinblick auf Embedded Software neu aufbereitet.

Ziele

Teilnehmende können maßgebliche Testtechniken in Embedded Projekten anwenden. Sie kennen die Charakteristika von testbaren Architekturen. Sie sind befähigt Testautomatisierung in ihren Projekten einzuführen.

Teilnehmer

Der Kurs wendet sich an Entwicklerinnen und Entwickler im Bereich Firmware, Treiber und sonstiger Embedded Software. Die Beispiele im Kurs sind in C gehalten, gelten eingeschränkt aber auch für C++. Vorkenntnisse im Bereich Software Testing werden keine vorausgesetzt.

Dauer

2 Tage

Termine und Preise

www.zuehlke.com/academy

Level

Beginner

Inhalt

- **Einführung in das Software Testen**
Warum? Wieviel? Effizienzbetrachtungen
Komponententest vs. Integrationstest
Testautomatisierung vs. manuelles Testen
- **Handwerkliches**
Komponententests, Platzhalter (Mocks, Test Doubles)
Was ist ein guter Testfall
Testabdeckung, Metriken
Umgang mit Legacy Code
Code Reviews
- **Test-Umgebungen**
Toolchain-Problematik
Testen am PC
Testen am Target
- **Testbare Software Architekturen (Embedded)**
Schichtenmodell Komponentenmodell
Asynchronität: Interrupts, Tasks, ...
Event-Bus Modell
Embedded Software Patterns
- **Best Practices**
- **Hands-on**
2 Beispiele (HW-Simulator)
Demonstration mit Hardware (target-based testing)

Sprache

Deutsch

Organisation

Öffentlich oder als Firmenkurs bei Zühlke oder beim Kunden.