

Einleitung: Skalierbare Qualität durch intelligente Testautomatisierung

In modernen Entwicklungszyklen ist die manuelle Verifikation aufgrund der Systemkomplexität längst an ihre Grenzen gestoßen. Um Release-Zyklen zu verkürzen und gleichzeitig die Absicherungstiefe zu erhöhen, ist eine hochgradige Automatisierung unerlässlich.

Unsere Schulung im Bereich Test-Frameworks konzentriert sich auf den **Automated Test Verification Workflow (ATVW)**. Wir zeigen Ihnen, wie Sie weg von isolierten Einzeltests hin zu einer integrierten, toolgestützten Testumgebung gelangen. Dabei lernen Sie, wie Sie Hardware-Schnittstellen, Simulationssoftware und automatisierte Auswertungsprotokolle zu einer schlagkräftigen Einheit verbinden, die konsistente und reproduzierbare Ergebnisse liefert.

Der Schulungsplan: Test-Frameworks & ATVW Mastery

Modul 1: Strategien der Testautomatisierung im Engineering Wir starten mit der methodischen Einordnung: Wann lohnt sich Automatisierung und wie strukturiert man Testsuites für maximale Wiederverwendbarkeit? Wir analysieren die Schichten eines modernen Frameworks und wie diese mit der restlichen IT-Infrastruktur interagieren.

- **Kerninhalte:** Auswahlkriterien für Testfälle, Wartbarkeit von Skripten und die Architektur eines generischen Test-Frameworks.

Modul 2: Einführung in den ATVW (Automated Test Verification Workflow) In diesem Modul tauchen wir tief in die Logik von ATVW ein. Wir betrachten den gesamten Prozess von der Testspezifikation über die automatisierte Generierung von Testskripten bis hin zur physischen Ausführung am Prüfstand.

- **Kerninhalte:** Workflow-Design, Rollenverteilung im automatisierten Prozess und die Anbindung an Requirements-Management-Tools.

Modul 3: Integration von Hardware-in-the-Loop (HiL) Systemen Ein Framework ist nur so gut wie seine Anbindung an die reale Welt. Wir schulen die Teilnehmer darin, wie sie Testautomatisierungswerkzeuge mit HiL-Simulatoren (z.B. dSPACE, VT-System oder National Instruments) koppeln, um Signale zu manipulieren und Systemreaktionen in Echtzeit zu erfassen.

- **Kerninhalte:** I/O-Konfiguration, Echtzeit-Schnittstellen und die Synchronisation zwischen Software-Logik und Hardware-Signalen.

Modul 4: Skripting, Bibliotheken und Schnittstellen Hier geht es ans "Eingemachte": Wir vermitteln die Erstellung modularer Testbibliotheken. Teilnehmer lernen, wie sie universelle Testschritte entwickeln, die für verschiedene Projekte adaptiert werden können, ohne den Code jedes Mal neu schreiben zu müssen.

- **Kerninhalte:** Parametersteuerung, Nutzung von Python oder speziellen Framework-Sprachen zur Teststeuerung und Error-Handling in automatisierten Läufen.

Modul 5: Automatisierte Analyse und Reporting Daten ohne Auswertung sind wertlos. Wir zeigen, wie das Framework nach einem Testlauf automatisch Traces analysiert, Soll-Ist-Vergleiche durchführt und professionelle Testberichte generiert. So wird die Fehleridentifikation von Stunden auf Minuten reduziert.

- **Kerninhalte:** Automatisierte Log-File-Analyse, Generierung von PDF/HTML-Reports und die Rückführung von Testergebnissen in Fehlermanagement-Systeme (z.B. Jira).