



Hardware in the Loop-Test in Kombination mit Continuous Integration

Dr.-Ing. Kristian Trenkel
SAEC Days 2020
22.07.2020

Agenda

- Einleitung
- Einführung Hardware-in-the-Loop-Testsysteme
- Verbindung der HIL-Systeme mit dem Continuous Integration
- Anwendungsbeispiel
- Zusammenfassung und Ausblick



Einleitung

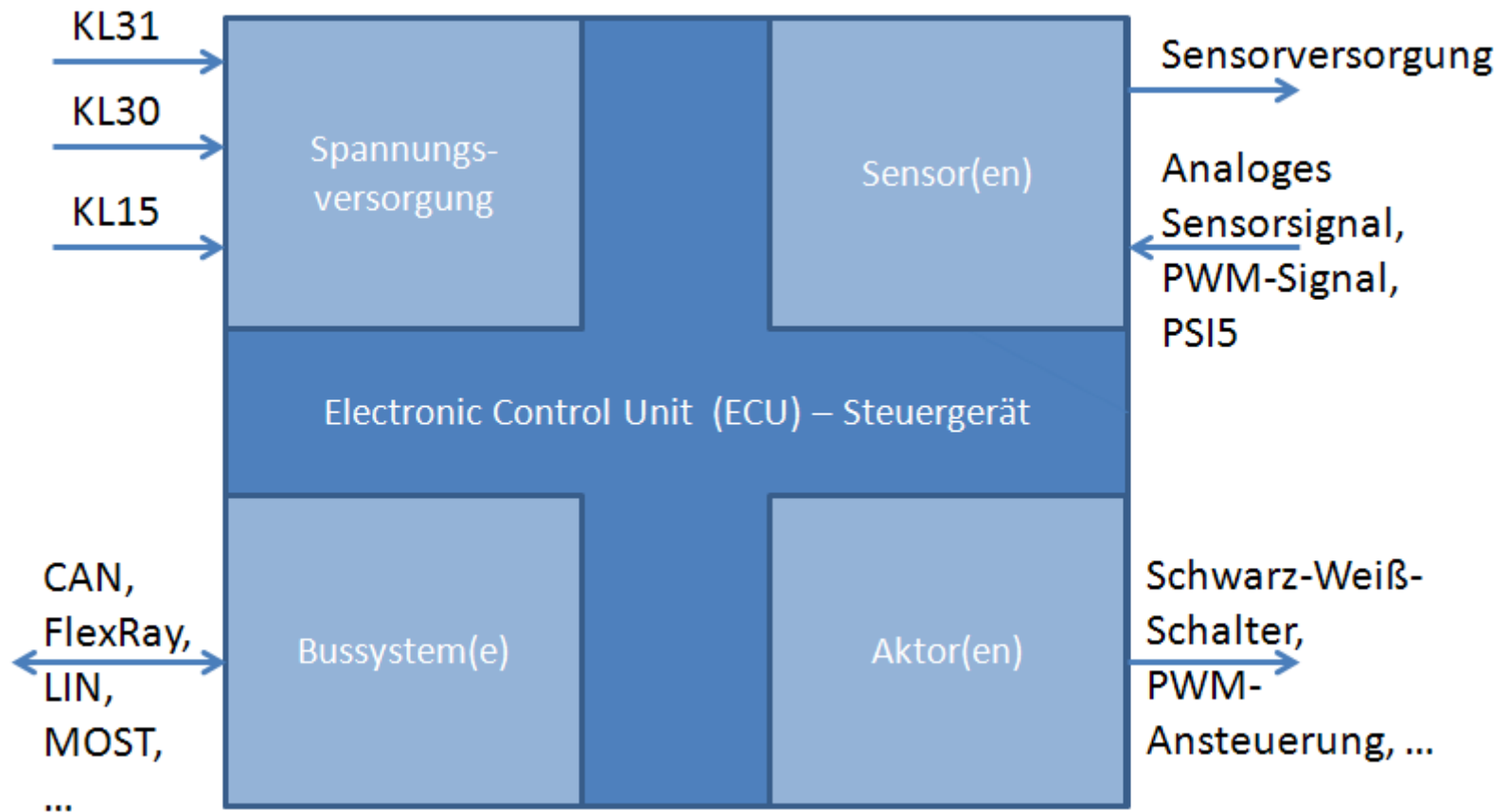
Embedded Systems

- Steigender Funktionsumfang
- Steigende Anzahl sicherheitskritischer Funktionen
- Immer stärkere Vernetzung der Systeme
- Verbreitung von Funkschnittstellen
- Direkte Verbindung zum Internet und zur „Cloud“



Einleitung

Eingebettetes System – Automotive



Agenda

- Einleitung
- Einführung Hardware-in-the-Loop-Testsysteme
- Verbindung der HIL-Systeme mit dem Continuous Integration
- Anwendungsbeispiel
- Zusammenfassung und Ausblick



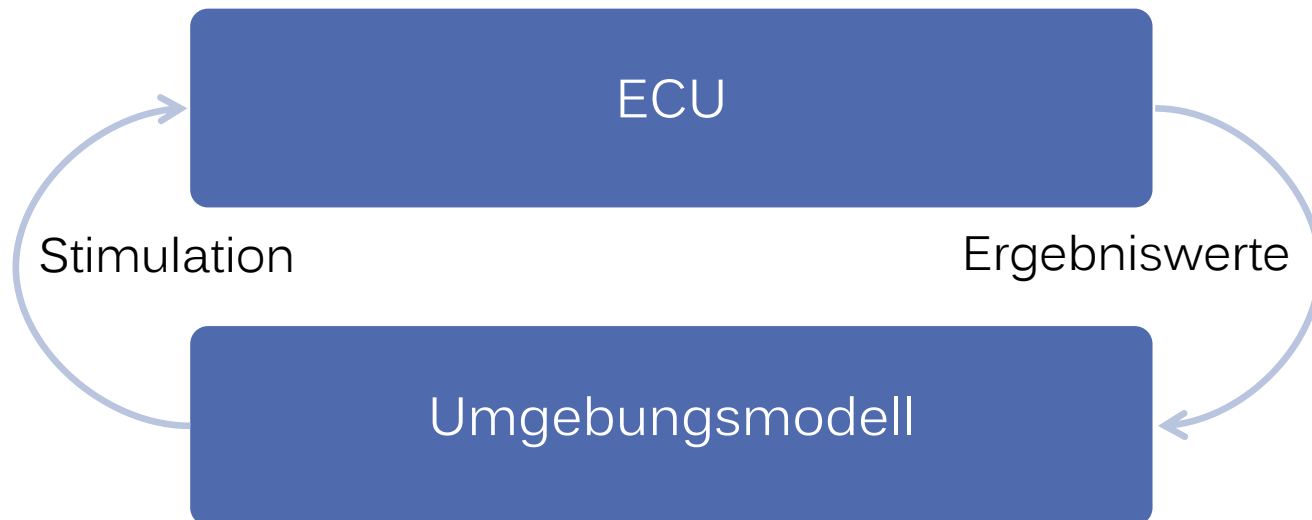
Einführung Hardware-in-the-Loop-Testsysteme

HIL-Test

- Begriffsklärung:

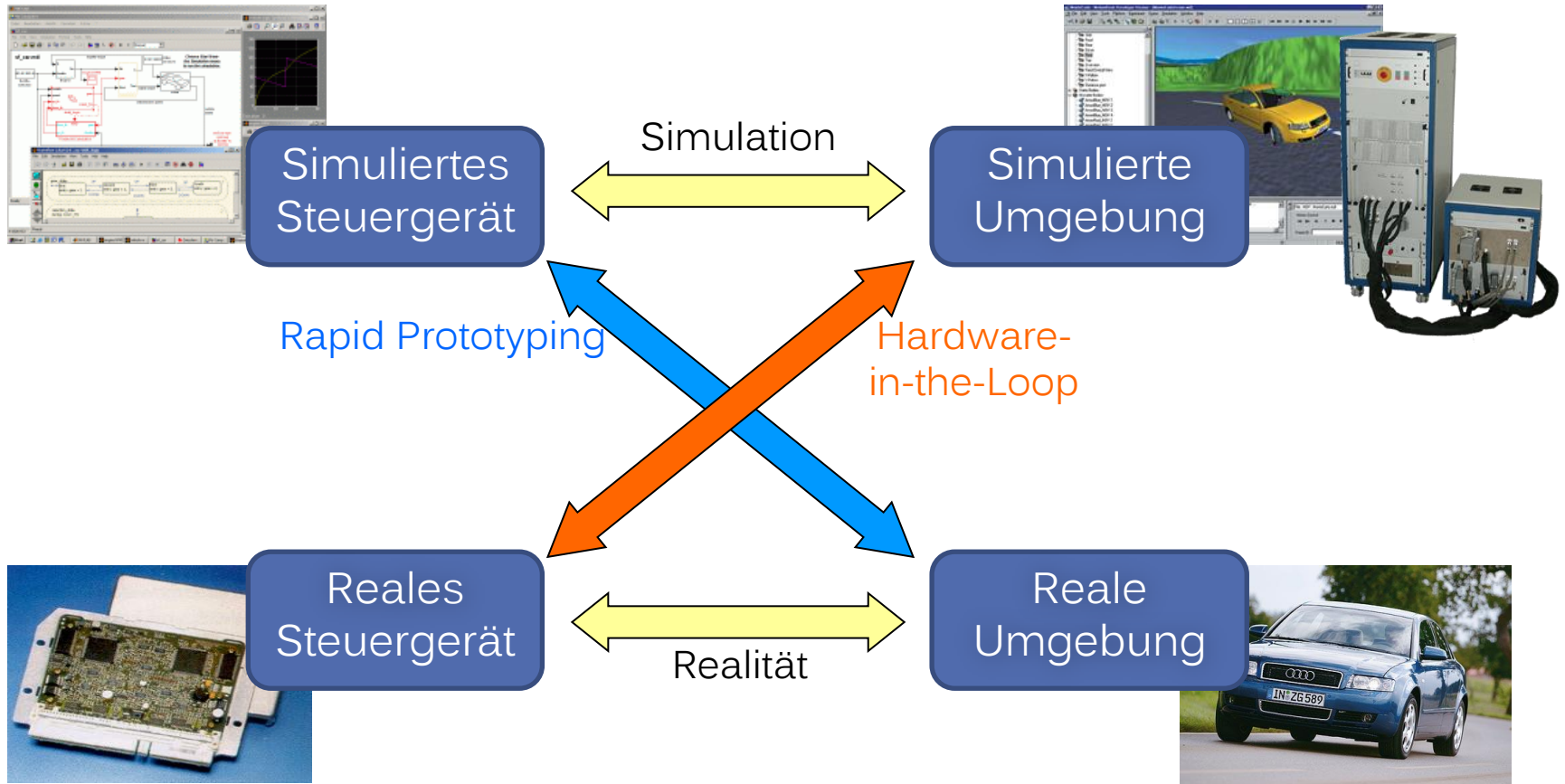
„bezeichnet ein Verfahren, bei dem ein eingebettetes System (z. B. reales elektronisches Steuergerät oder reale mechatronische Komponente) über seine Ein- und Ausgänge an ein angepasstes Gegenstück, das im allgemeinen HIL-Simulator genannt wird und als Nachbildung der realen Umgebung des Systems dient, angeschlossen wird.“

- Entwicklungsbegleitendes Testverfahren für Embedded Systems



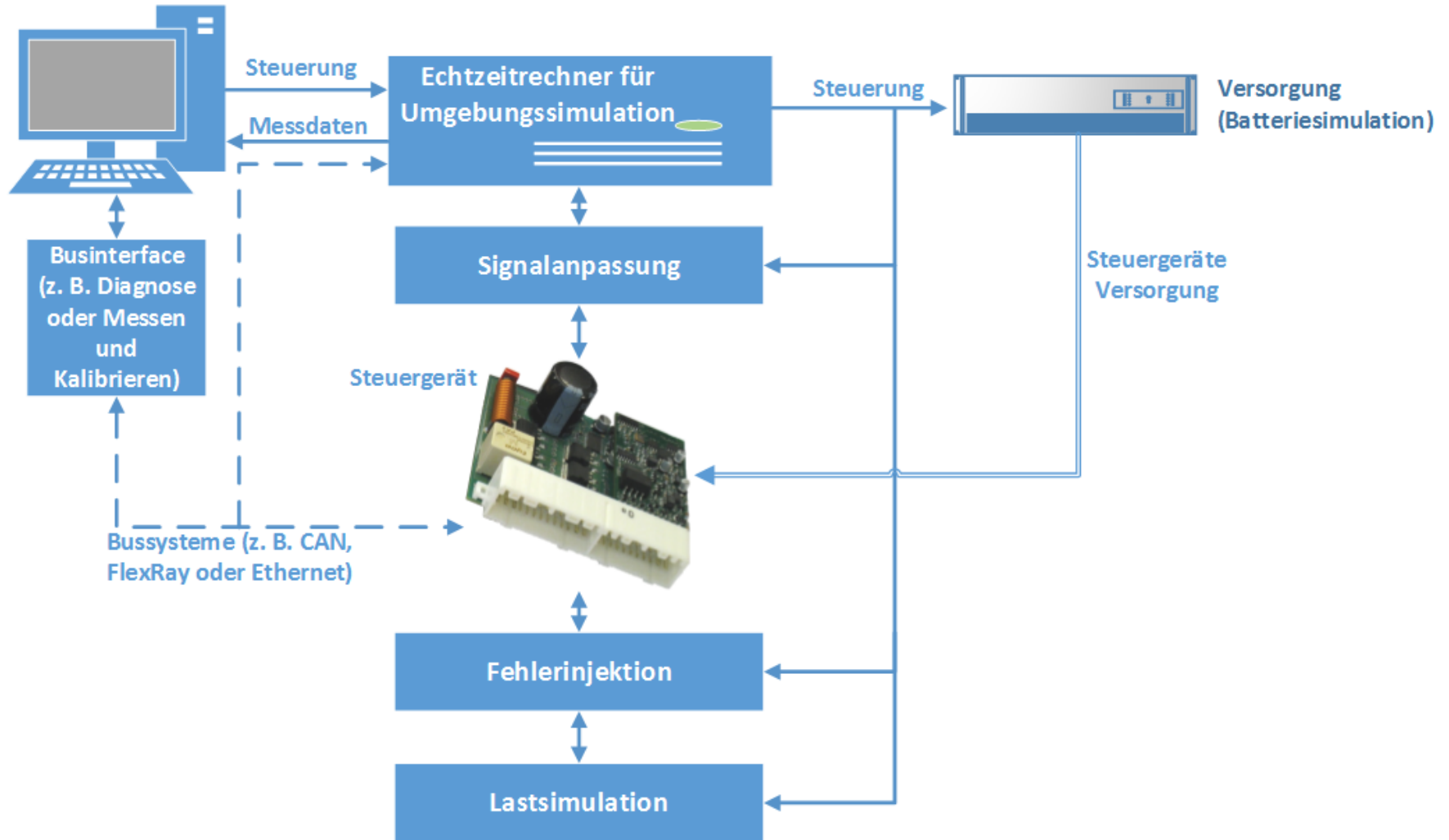
Einführung Hardware-in-the-Loop-Testsysteme

HIL-Test



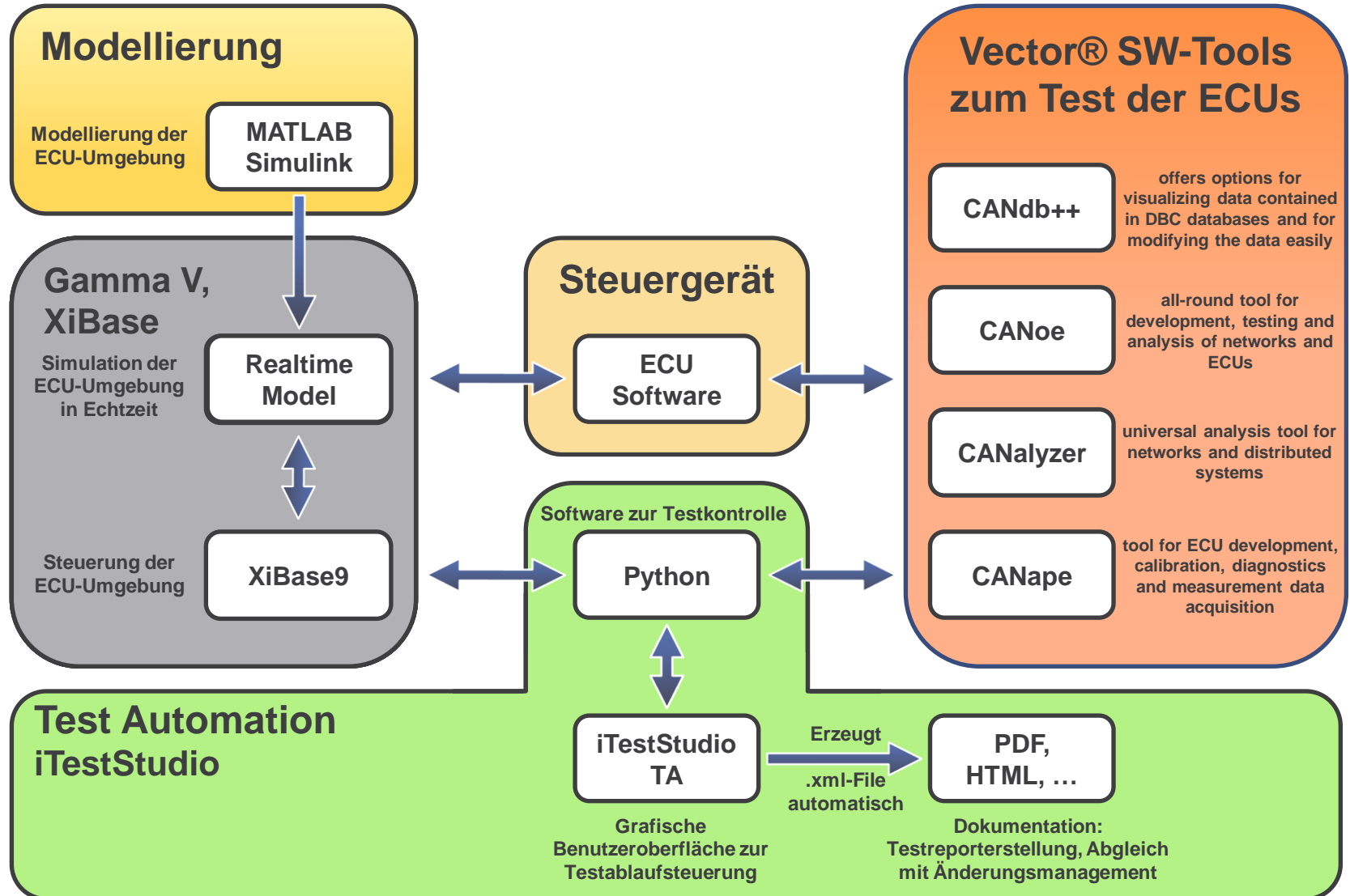
Einführung Hardware-in-the-Loop-Testsysteme

HIL-Testsystem – Aufbau – Hardware



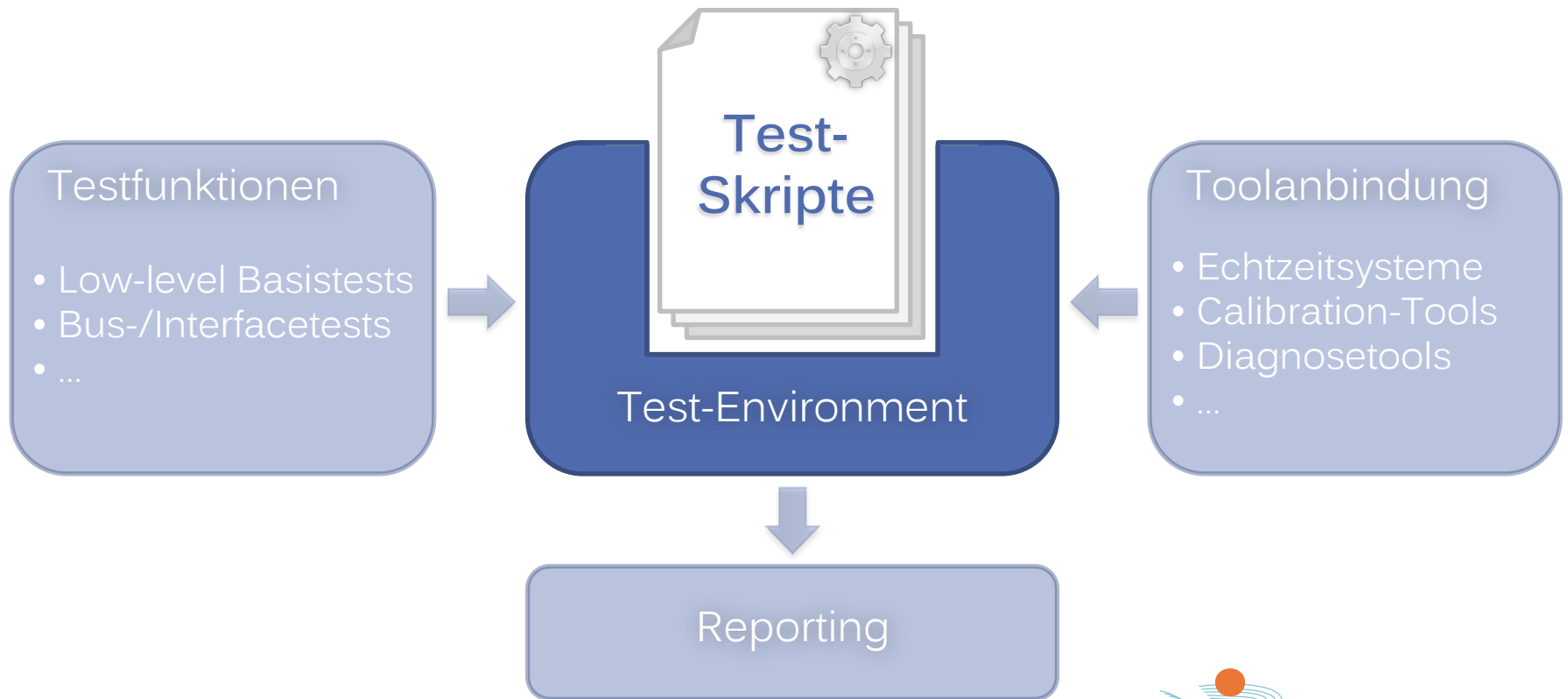
Einführung Hardware-in-the-Loop-Testsysteme

HIL-Testsystem – Typische Softwarestruktur am Beispiel des iTestStudios



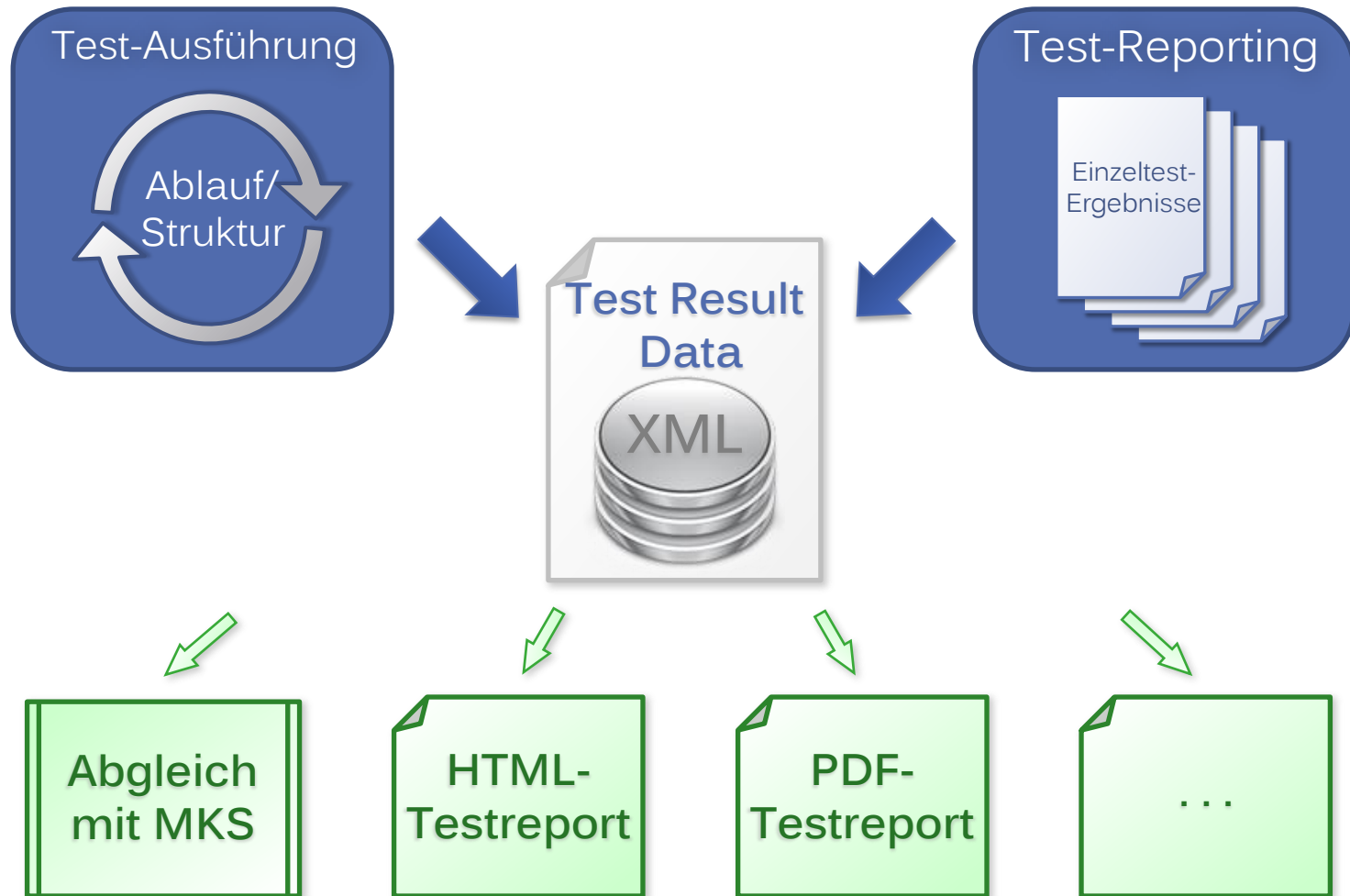
Einführung Hardware-in-the-Loop-Testsysteme

Testautomatisierung – iTestStudio



Einführung Hardware-in-the-Loop-Testsysteme

iTestStudio – Übersicht

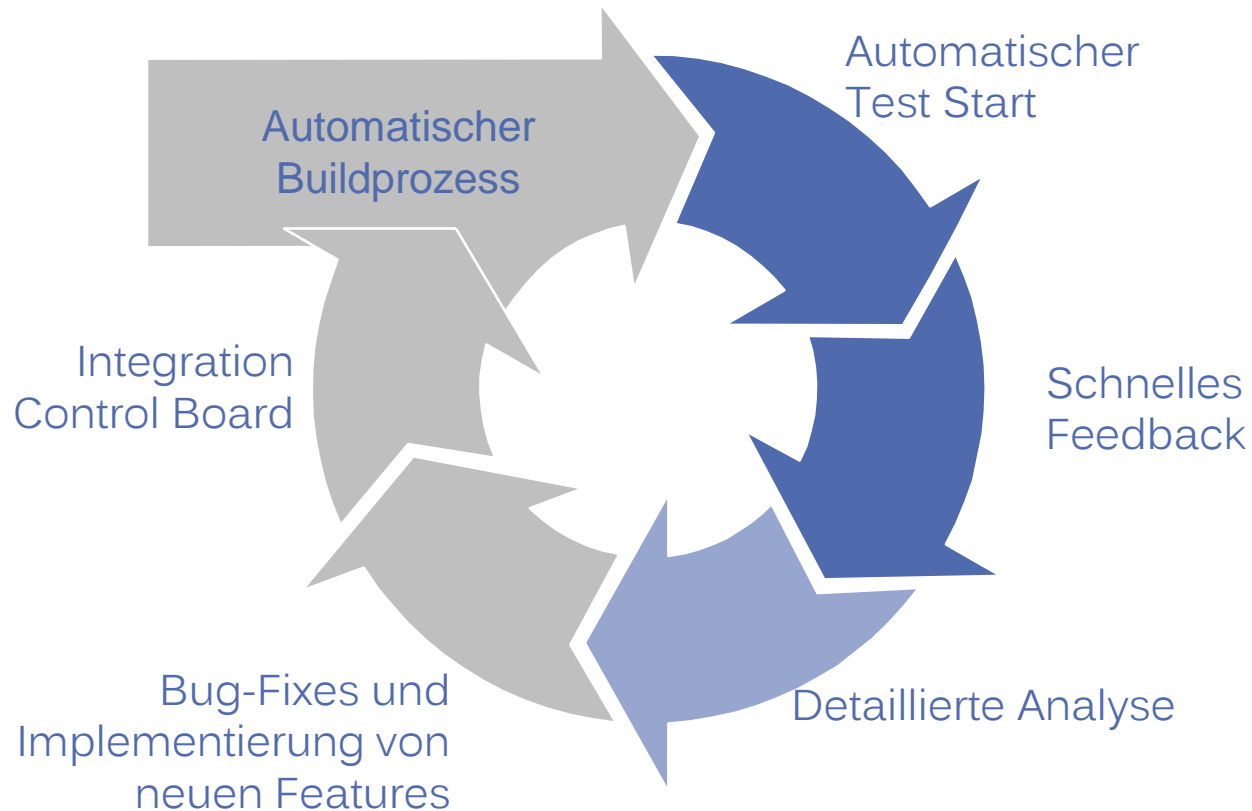


Agenda

- Einleitung
- Einführung Hardware-in-the-Loop-Testsysteme
- Verbindung der HIL-Systeme mit dem Continuous Integration
- Anwendungsbeispiel
- Zusammenfassung und Ausblick

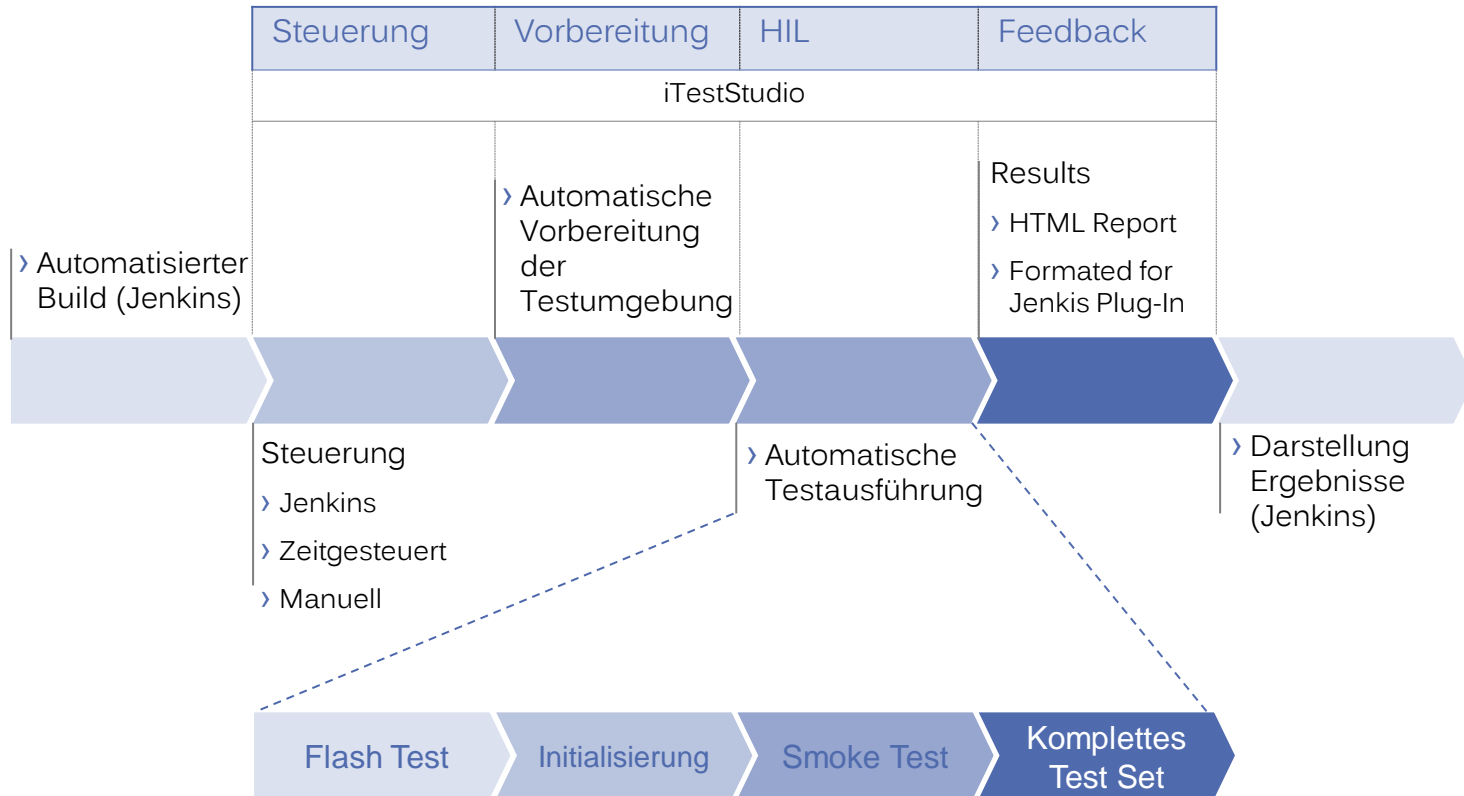


Verbindung der HiL-Systeme mit dem Continuous Integration Entwicklungszyklus



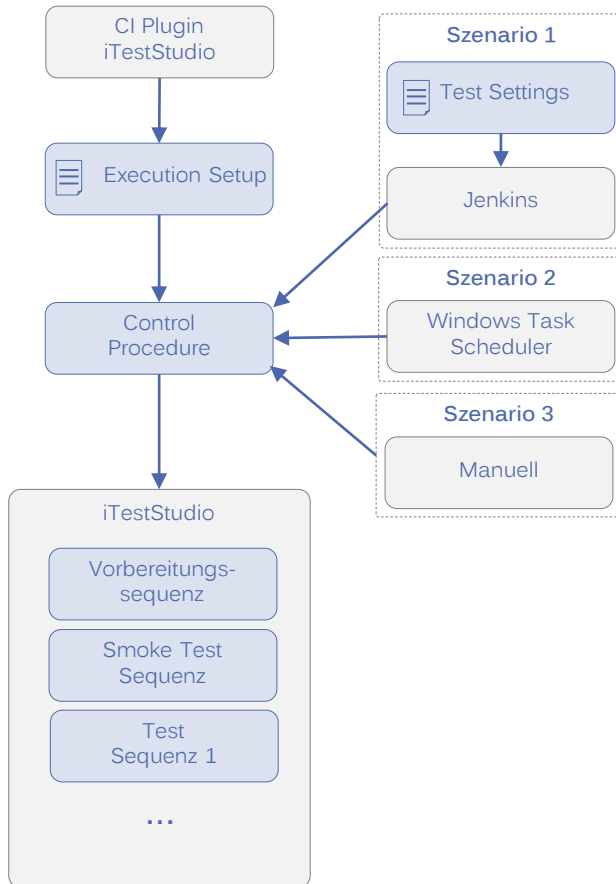
Verbindung der HIL-Systeme mit dem Continuous Integration

Integrationskonzept



Verbindung der HIL-Systeme mit dem Continuous Integration

Steuerung



Test Start durch Jenkins

- Nach erfolgreichem Software Build Prozess
- Smoke Tests mit direkter Rückmeldung
- Kompletter Regressionstest wenn der Smoke Test bestanden wurde

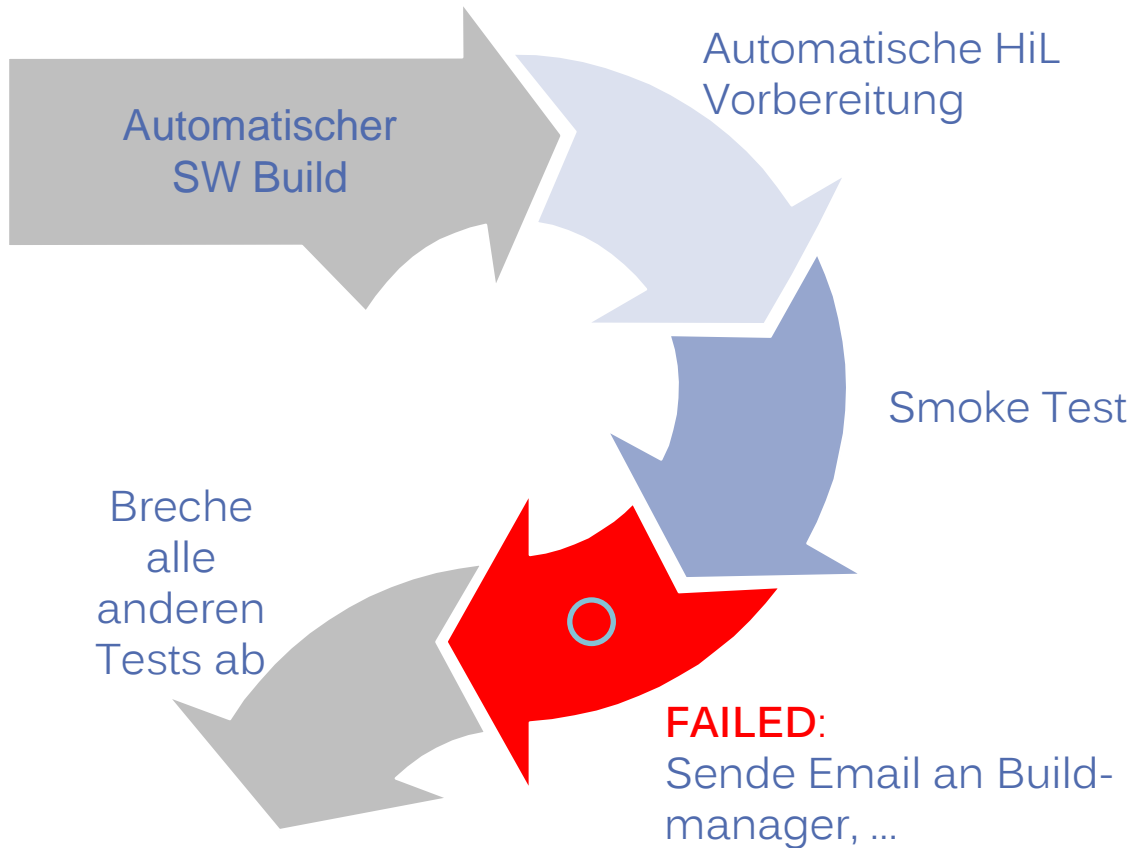
Test Start durch Test Team

- Jede Software zu jeder Zeit
- Verwende nur Tests die notwendig sind
- Exportiere die Ergebnisse nach Jenkins oder verwende sie lokal



Verbindung der HiL-Systeme mit dem Continuous Integration

Hardware-in-the-Loop

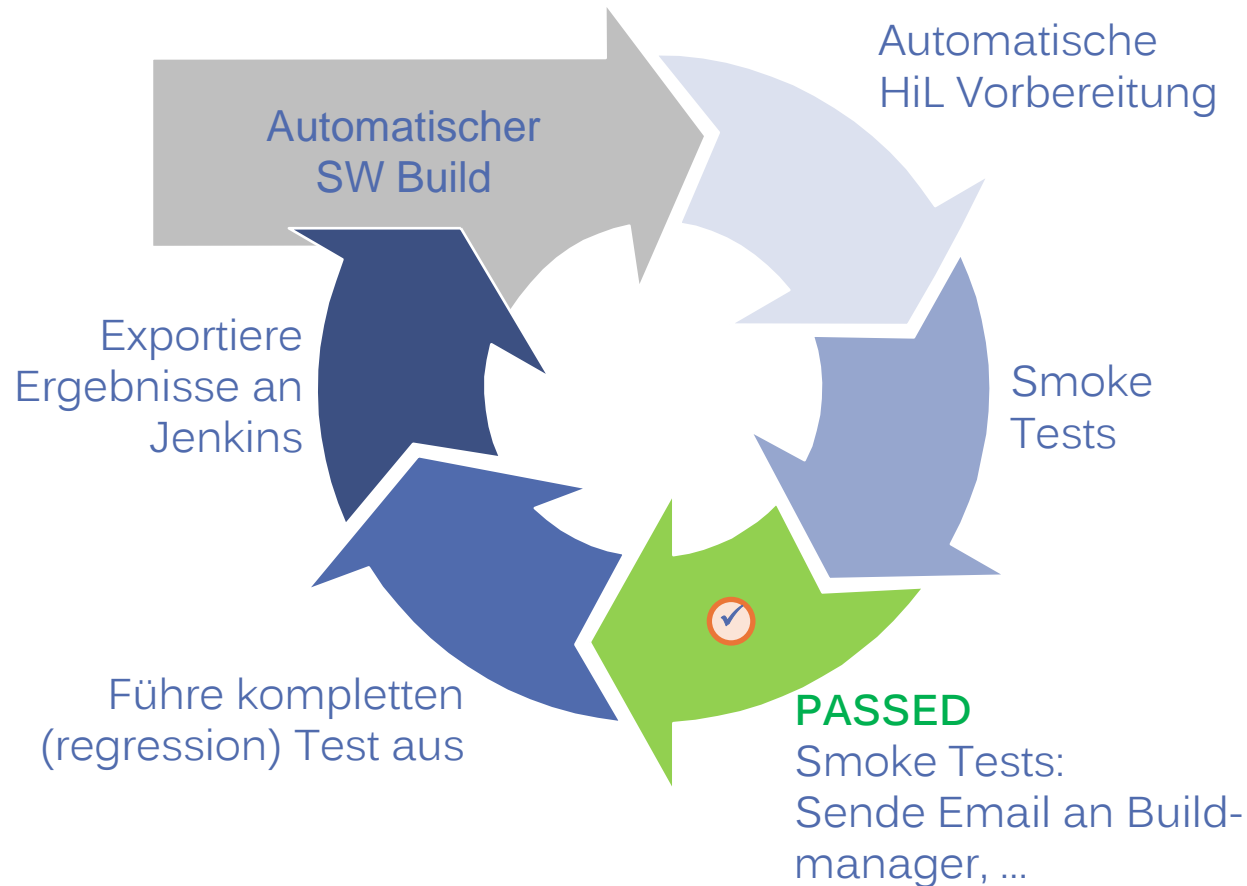


Grundidee:

- Schnelle Rückmeldung an Entwicklungsteam
- Weniger Aufwand für das Testteam um Ergebnisse bereitzustellen
- Mehr Zeit für Testentwicklung
- Spare Ressourcen für effizientes Testen

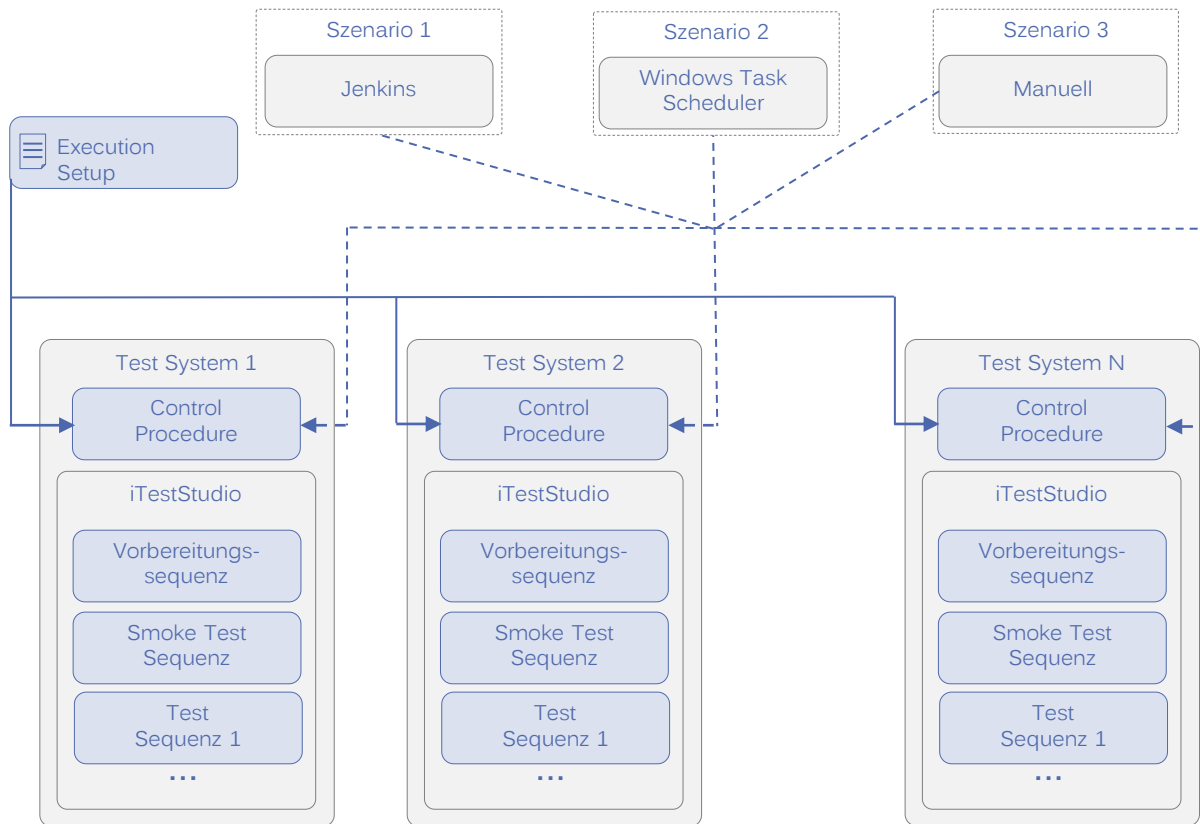
Verbindung der HiL-Systeme mit dem Continuous Integration

Hardware-in-the-Loop



Verbindung der HiL-Systeme mit dem Continuous Integration

Parallele Testausführung auf unterschiedlichen Systemen



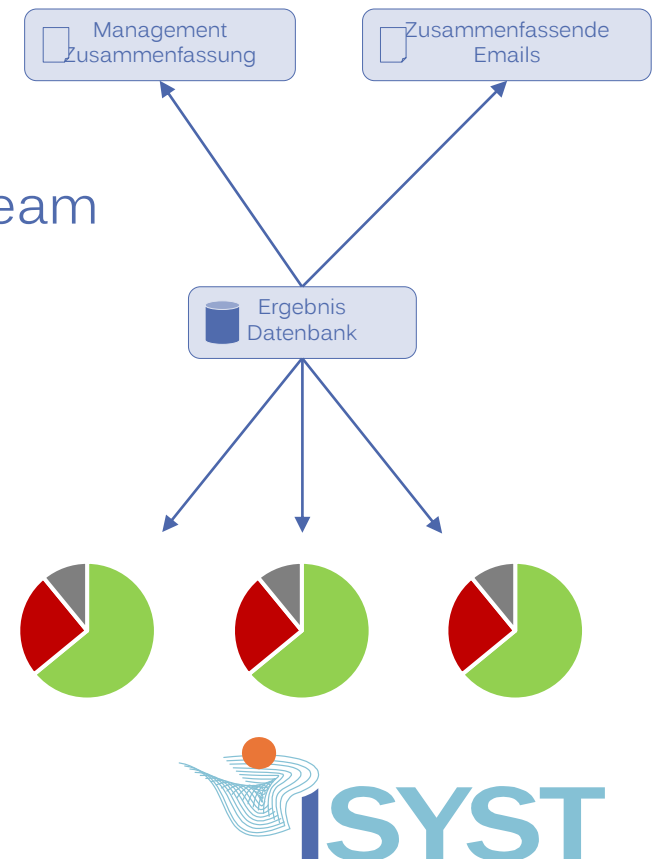
Features

- Globale Steuerung für mehrere Testsysteme
- Parallele Testausführung
- Zusammenfassendes Feedback
- Schnelles Feedback für das Projektmanagement

Verbindung der HiL-Systeme mit dem Continuous Integration

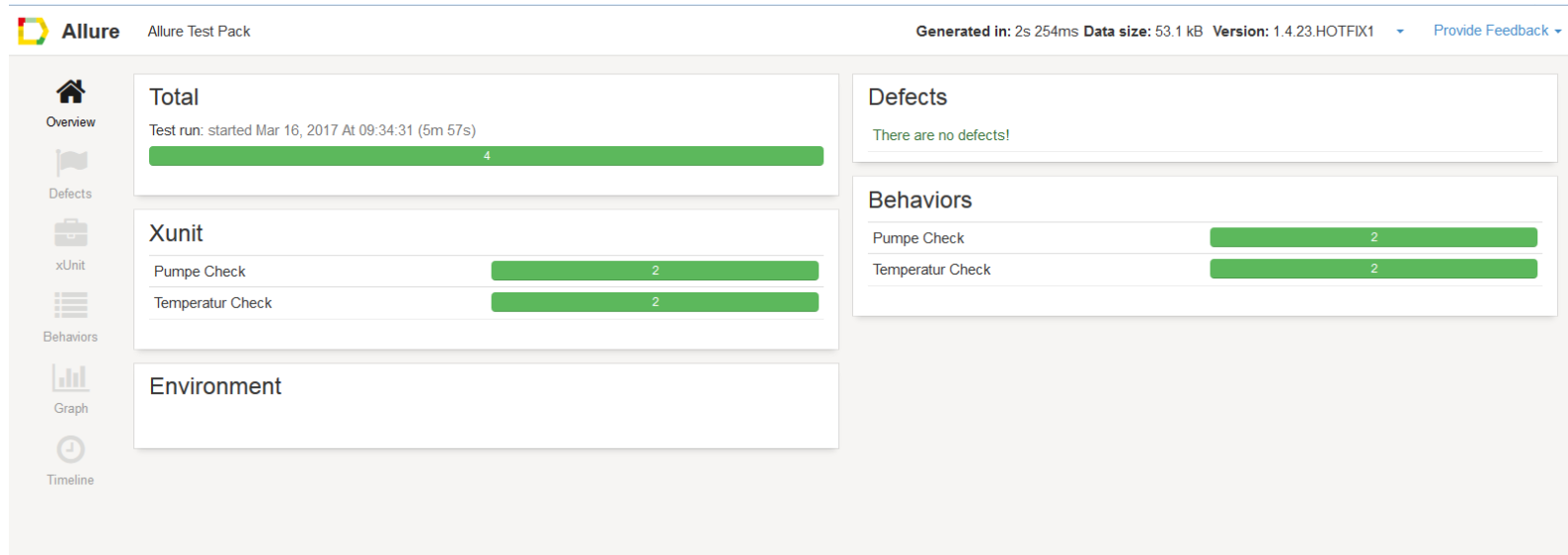
Rückmeldung

- Schnelle Rückmeldung für das Management
 - Management Zusammenfassung (Datenbank)
 - Visualisiere Ergebnisse und Trends
- Schnelle Rückmeldung für Entwicklungsteam
 - Automatische Emails mit Zusammenfassung
 - Verschiedene Stufen detaillierter Informationen
 - Verschiedene Empfänger (SW Manager, ...)
- Darstellung der Ergebnisse
 - Evaluierung
 - Jenkins (Allure)
 - Statistiken



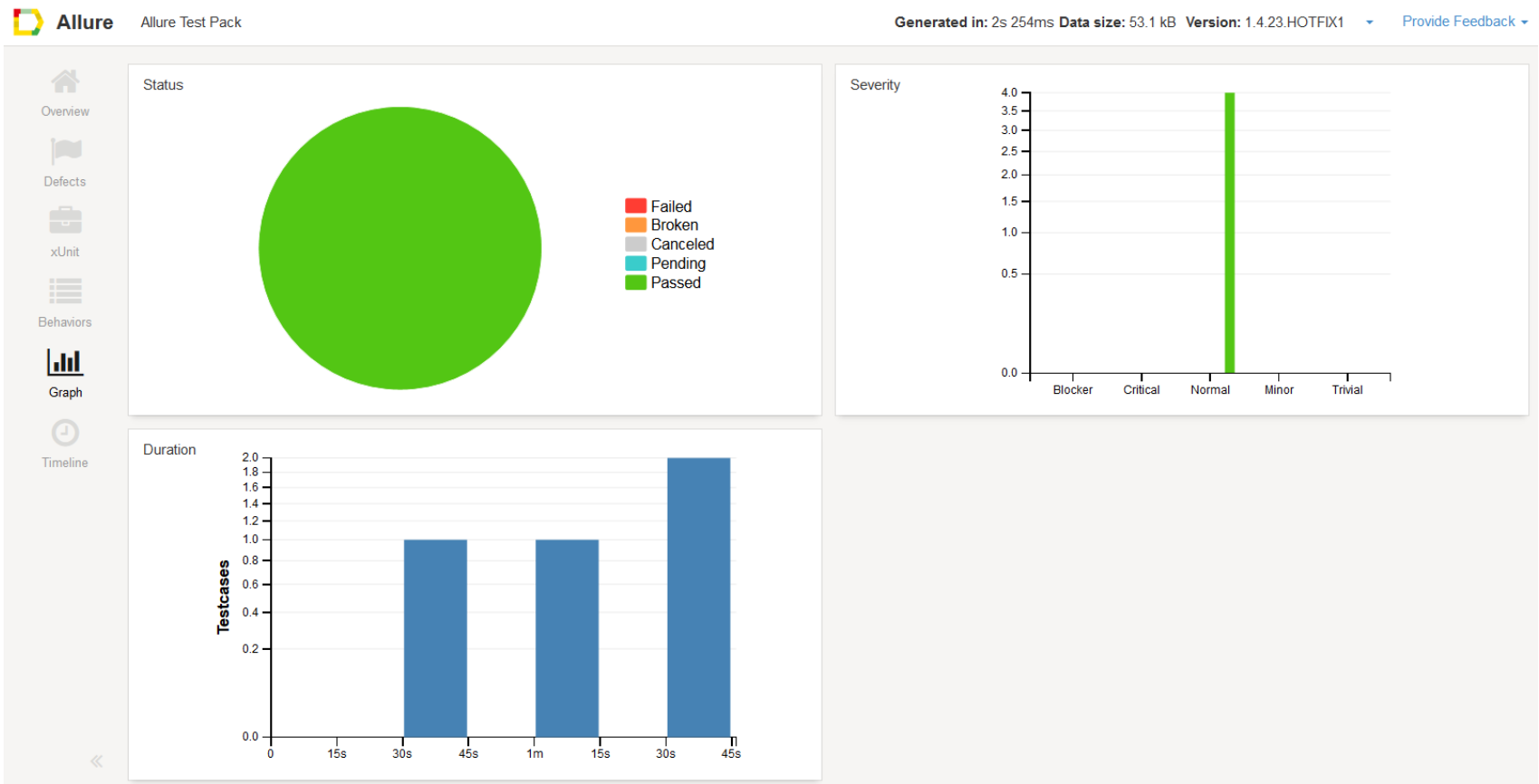
Verbindung der HiL-Systeme mit dem Continuous Integration

Rückmeldung – Allure Report



Verbindung der HiL-Systeme mit dem Continuous Integration

Rückmeldung – Allure Report



Agenda

- Einleitung
- Einführung Hardware-in-the-Loop-Testsysteme
- Verbindung der HIL-Systeme mit dem Continuous Integration
- Anwendungsbeispiel
- Zusammenfassung und Ausblick



Anwendungsbeispiel

Fahrwerkssteuergerät – Ausgangspunkt

- Steuergerät für die Regelung eines aktiven Fahrwerkes
- 3 Varianten
- Ca. 3000 Anforderungen
- Einige 100 Safety-Anforderungen bis ASIL D
- 5 HiL-Testsysteme parallel im Einsatz
- Sehr kritischer Projektverlauf bei der Softwareentwicklung
- Testlaufzeit ca. 40h – 500.000 Bewertungen



Anwendungsbeispiel

Fahrwerkssteuergerät – Problemstellungen

- Daily Build der Software – Fertigstellung meist nach 18 Uhr
- Manuelle Einrichtung und Start der Test dauert zum Teil bis 22 Uhr
- Ab 6Uhr musste die Auswertung der Ergebnisse für Management-Meeting (8 Uhr) erfolgen
 - große Belastung der Mitarbeiter
 - Hohe Fehlerrate beim Testsetup
- Basierend auf vorhandenen Skripten konnte das HIL-Setup und der Teststart zu 100% automatisiert werden (Aufwand ca. 2 MM)
- Frei konfigurierbare Verteilung der Test auf die HILs



Anwendungsbeispiel

Vorgehen und Notwendige Rahmenbedingungen

- Test Systeme
 - Initialisiere Echtzeit Modell (Neu laden des Modells)
 - Konfiguration der Tools und Software (z.B. dSpace Control Desk NG, Vector Canape)
- Umgebung
 - Möglichkeit automatische Emails zu versenden (Test System / Benutzer Account)
 - Zugriff auf Software Ordner im Netzwerk
 - Zugriff auf Repository
- Entwicklungsprozess
 - Automatischer Build Prozess
 - Testbare Software zur Verfügung stellen



Anwendungsbeispiel

Ergebnisse

- Entlastung der Mitarbeiter
- 100%ige Automatisierung des Setups der HIL-Systeme, des DUT und des Teststarts
- Vollständige Beseitigung der Fehler bei Testsetup
- Automatisierte Information an beteiligte Mitarbeiter – Auch Fehlermeldung (E-Mail)
- Automatisierte Rückmeldung der Testergebnisse – Aufbereitung für Management erfolgt automatisiert



Anwendungsbeispiel

Offene Punkte

- Statistische Übersicht
 - Smoke Test
 - Komplettes Test Sets
 - Alle HiLs (Einheitlichkeit und Reproduzierbarkeit der Testergebnisse)
 - Alle Software Versionen (Regression, Trend, Vergleich, Abhängigkeiten)



Agenda

- Einleitung
- Einführung Hardware-in-the-Loop-Testsysteme
- Verbindung der HIL-Systeme mit dem Continuous Integration
- Anwendungsbeispiel
- Zusammenfassung und Ausblick



Zusammenfassung und Ausblick

- Vollständige Automatisierung der Ausführung der HIL-Tests in Verbindung mit CI
- Umfangreiches Softwareprojekt (Automobil) in Wartungsphase kann durch ½ (Software) Testmanager betreut werden
- Stabile Testausführung in “Nightly” Testläufen
- Nächtliche Testausführung möglich ohne Nachtschicht
- Reduziert Teststart Zeit auf 20%
- Test Entwickler können sich auf das Wichtige konzentrieren
- Software Entwickler bekommen direkte Rückmeldung vom HIL
- Verringert Fehler durch komplexe manuelle Test Starts
- iTestStudio (iSyst) ist komplett in den CI Prozess einbindbar





Vielen Dank für die
Aufmerksamkeit!

Fragen oder Anregungen?