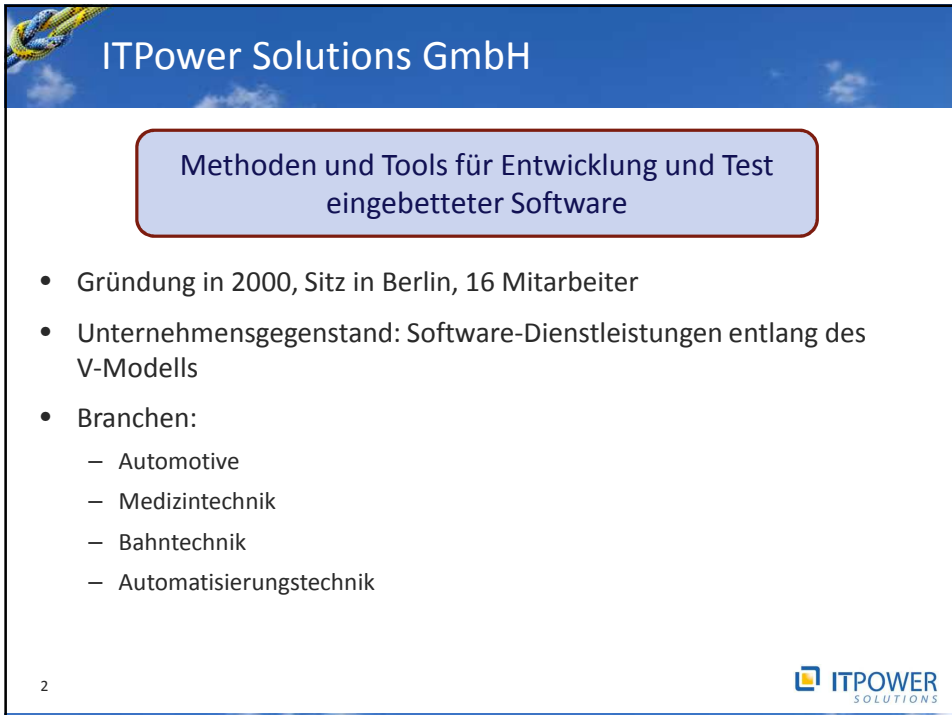




Gleitender Übergang vom manuellen zum automatisierten Test eingebetteter Software

40. Treffens der GI-Fachgruppe Test, Analyse & Verifikation von Software (TAV), 09.02.2017

Dr. Sadegh Sadeghipour
sadegh.sadeghipour@itpower.de




ITPower Solutions GmbH

Methoden und Tools für Entwicklung und Test eingebetteter Software

- Gründung in 2000, Sitz in Berlin, 16 Mitarbeiter
- Unternehmensgegenstand: Software-Dienstleistungen entlang des V-Modells
- Branchen:
 - Automotive
 - Medizintechnik
 - Bahntechnik
 - Automatisierungstechnik


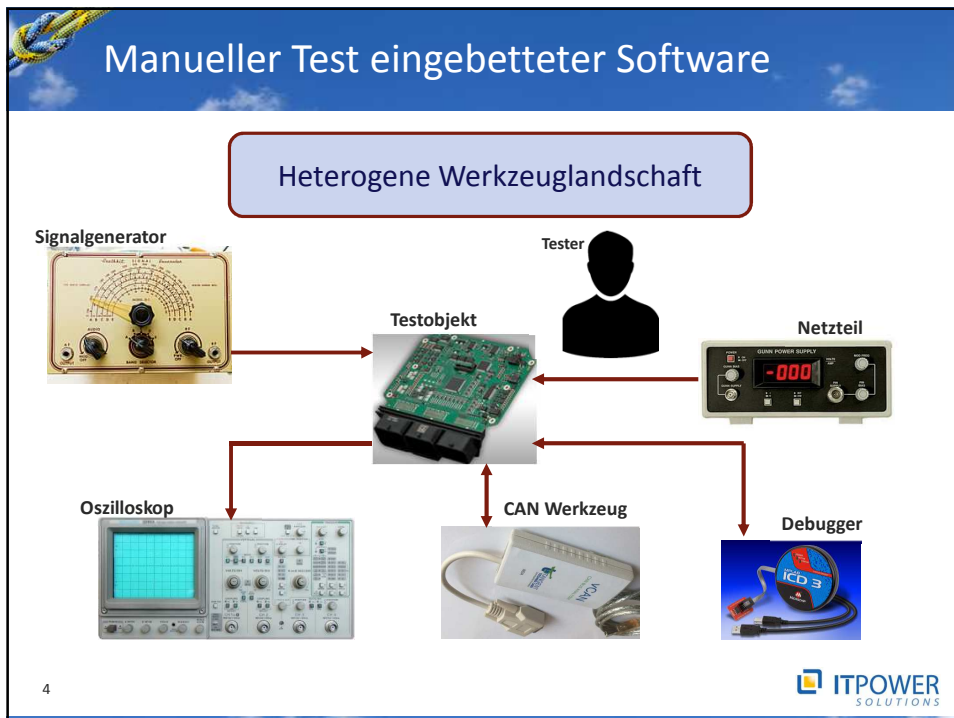
2




Inhalt

- 1 Ausgangspunkt
- 2 Architektur des Testframeworks
- 3 Tool und Beispiel
- 4 Fazit


3



- 1 Ausgangspunkt
- 2 Architektur des Testframeworks
- 3 Tool und Beispiel
- 4 Fazit

5




Testautomatisierung: Hardware-in-the-Loop

Hardware-in-the-Loop: Reaktiver, oft echtzeitfähiger Test eines eingebetteten Systems anhand eines Nachbils der realen Umgebung (HiL-Simulator)

- + Große Mächtigkeit
- + Skalierbarkeit auf Testobjekte mit unterschiedlichen Schnittstellen
- + Echtzeitfähigkeit
 - Abschied von vertrauten Werkzeugen und dem erworbenen Know-how
 - Hohe Investitions- und Betriebskosten
 - Unterschiedliche Testumgebungen für verschiedene Testphasen

6

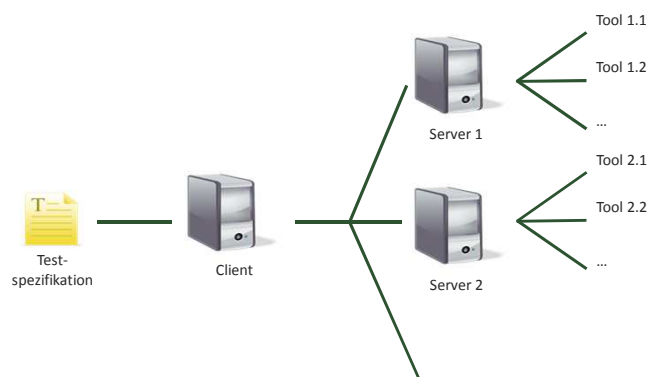


Anforderungen an ein alternatives Framework

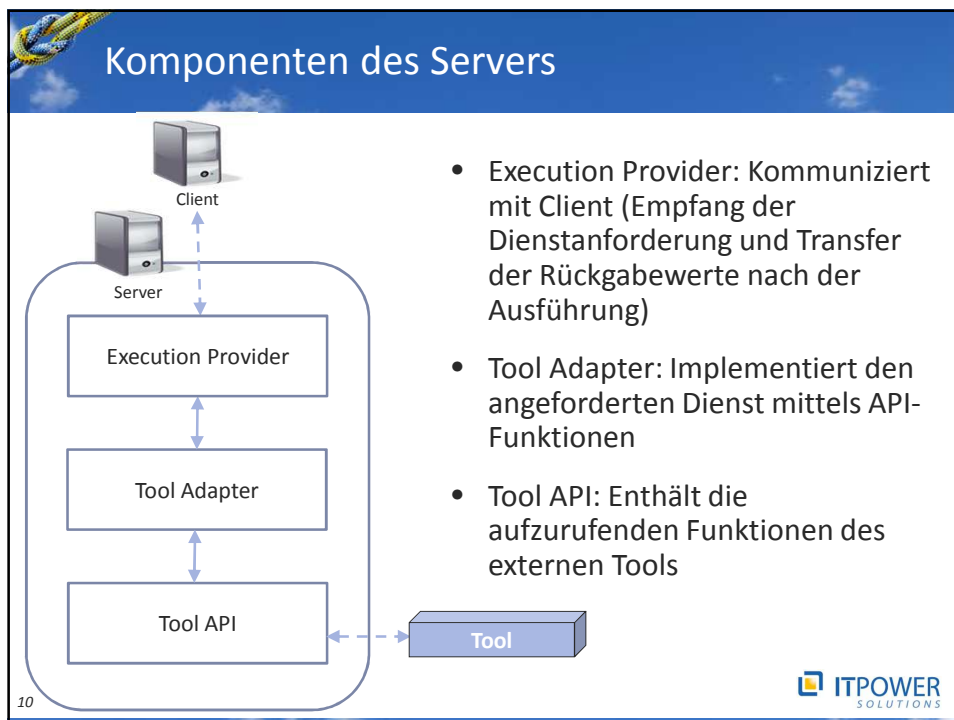
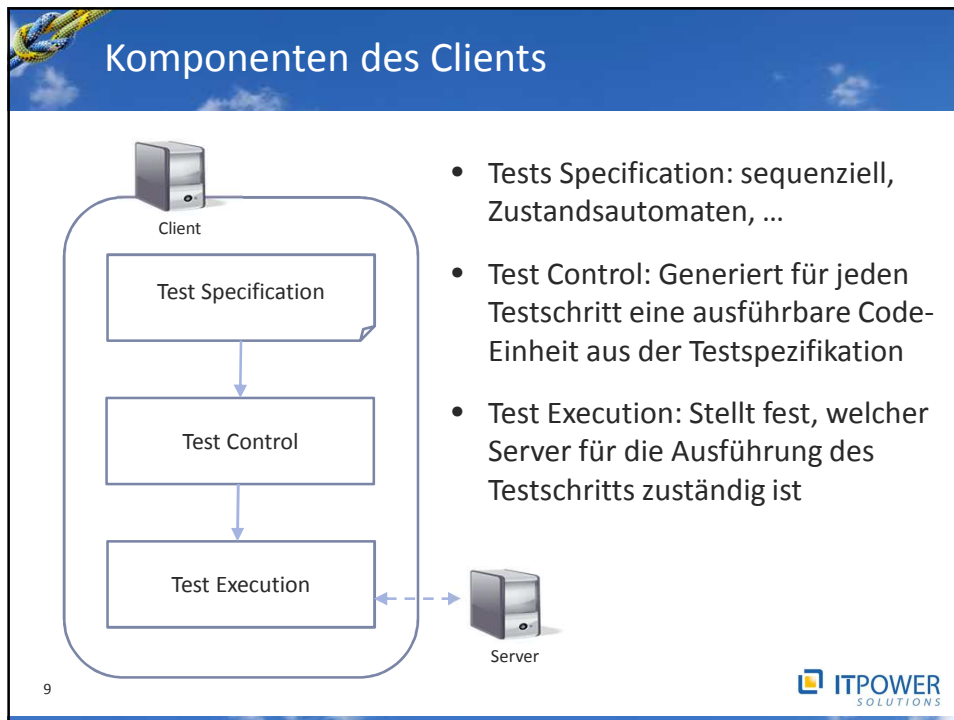
- Im Entwicklungsprozess verwendete und vertraute Werkzeuge sollen beim Test eingesetzt werden.
- Werkzeuge kommunizieren über ein Framework.
- Werkzeugübergreifende einheitliche Testspezifikation
- Einfache Bedienbarkeit

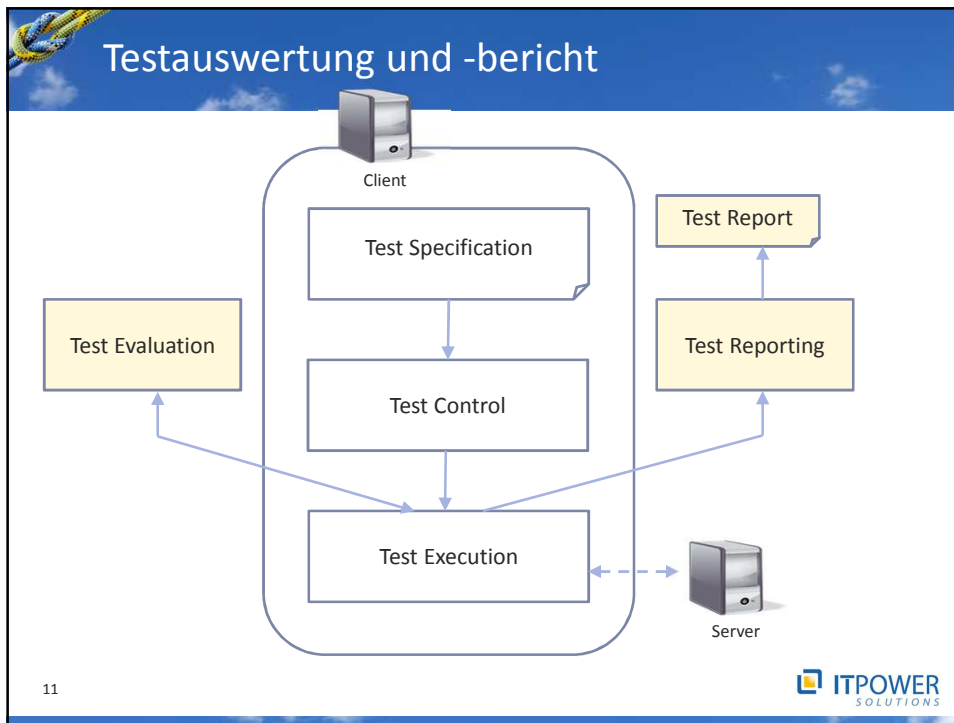
7

Client-Server Architektur



8





- 1 Ausgangspunkt
 - 2 Architektur des Testframeworks
 - 3 Tool und Beispiel
 - 4 Fazit
- 12
- ITPOWER SOLUTIONS

ContinoProva

Zeitsteuerung

Teststruktur →

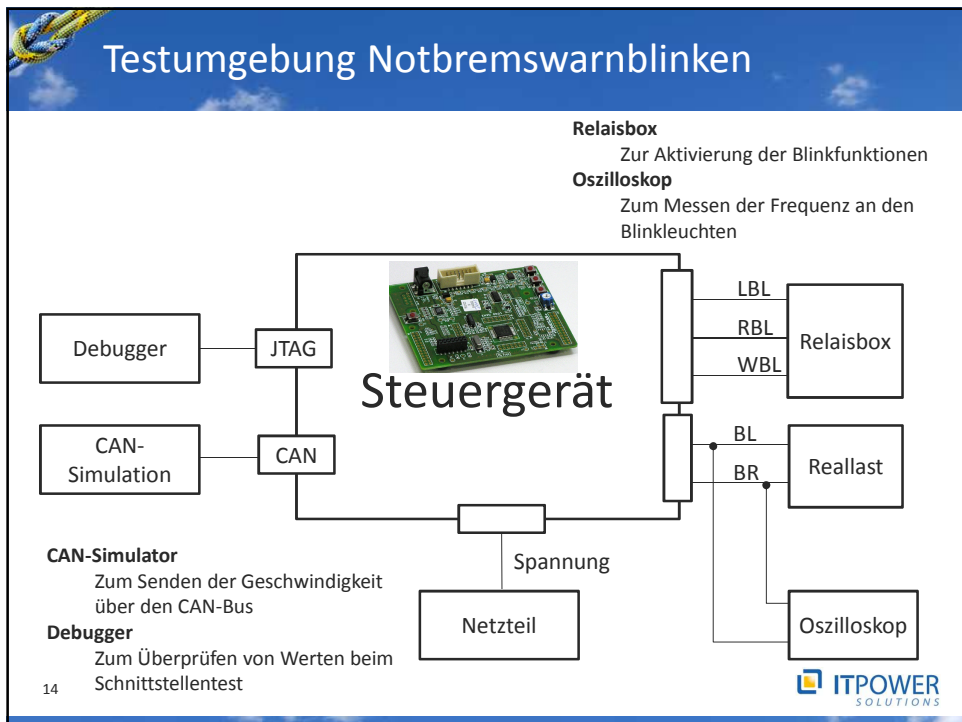
- ▼ Testgruppe
- ≡ Testsequenz
- Testschritt
- Testtask

Service

Operation

ITPOWER SOLUTIONS

13



14

TestszENARIO


- TestszENARIO: Nach der Aktivierung des linken Blinkers und bei einer Geschwindigkeit größer als v_{\min} wird mit einer Verzögerung größer als a_{kritisch} gebremst.
- Erwartetes Verhalten: Beide Blinker blinken mindestens für die Dauer von t_{ab} mit einer Frequenz f_n . Nach Ablauf der Zeit wird das Notbremswarnblinken zurückgesetzt und der Zustand Richtungsblinken wieder eingenommen.

15

Testspezifikation



Testschritt	Tool	Operation	Zeitverhalten
Init			
SetHScale	Oszilloskop	SetHorizontalScale(0.00025)	WaitAfter(500)
SetVerticalScale CH1	Oszilloskop	SetVerticalScale(CH1, 1)	
SetTriggerType	Oszilloskop	SetTriggerType(EDGE)	
ConnectToCANmodul	CAN	Connect()	
OpenWorkspace	Debugger	OpenWorkspace(c:\Nbw.ws)	WaitAfter(2000)
Testsequence			
LoadVelocityScale	CAN	LoadScale(vs, c:\VelocityScale.txt)	
SetLeftIndicator	RelaisBox	WriteOutput(LB, 1)	
ExecuteVelocityScale	CAN	ExecuteScale(vs)	
MeasureLeftIndicator == 5kHz	Oszilloskop	ReadAutoMeasurement(1) == 5kHz	WaitUntil(10000)
MeasureRightIndicator == 5kHz	Oszilloskop	ReadAutoMeasurement(2) == 5kHz	CheckWhile(5000)
MeasureLeftIndicator == 1kHz	Oszilloskop	ReadAutoMeasurement(1) == 1kHz	

16



- 1 Ausgangspunkt
- 2 Architektur des Testframeworks
- 3 Tool und Beispiel
- 4 **Fazit**


17



Fazit

- Automatisierter Test auf allen Teststufen
- Verbindung und Fernsteuerung verschiedener Werkzeuge und flexible Implementierung neuer Werkzeuganbindungen
- Werkzeugübergreifende einheitliche Testspezifikationen
- Test-Debugging
- Tests mit harten Echtzeitanforderungen nicht möglich!

18



Danke für Ihre Aufmerksamkeit!

