

Optimierungsstrategien beim Systemtest



Ein Vortrag für das



Forum Funktionale Sicherheit 2014
15.-17. Juli 2014
Kempinski Hotel Airport München

Das bin ich....



- ◆ Dipl.-Ing. (FH) Paul Huber, MBA
- ◆ **seit 2006 Ingenieurbüro Paul Huber**
 - Requirements-Engineering
 - Test-Engineering
 - Geschäftsprozesse
- ◆ **Weiterhin bin ich**
 - Autor eines Lehrbriefes zum Thema Requirements-Engineering
 - Fachgruppenleiter der regionalen ASQF Fachgruppe Software-Test Schwaben (www.ASQF.de)
 - Veranstalter des Neu-Ulmer Test-Engineering Days (www.NU-TED.de)



Voraussetzungen für wirtschaftliches autom. Testen

- ◆ Die Produkte und Produktversionen sollten sich ähnlich sein
- ◆ Es muss die Möglichkeit vorhanden sein, mit dem Prüfling automatisch zu interagieren
- ◆ Die Ergebnisse der Test müssen automatisch erfasst werden können
- ◆ Die Testfälle müssen modular aufgebaut sein
- ◆ Die Testfälle und Testumgebungen müssen wiederverwendbar sein

Strukturierung von automatischen Tests

Analog-Eingänge

- Klopf-Sensor
- Druck-Detektion
- Lambda-Sonde

Digital-Eingänge

- Motordrehzahl
- sonstige Digitale Eingänge

Stromversorgung

- eigene Stromversorgung
- ext. Sensoren

Prozessor

- CPU, RAM, Speicher
- Resistor, Dioden, Busse
- Schalter, Zündung
- Motor, Signal Output

High-Side Outputs

- High-Power

Low-Side Outputs

- Low-Power

DC-Motor-Bridge

- Steuerung der Drosselklappe

Systemarchitektur ↔ Testarchitektur
Softwarearchitektur ↔ Testarchitektur
sollten zueinander passen

Kommunikation nach

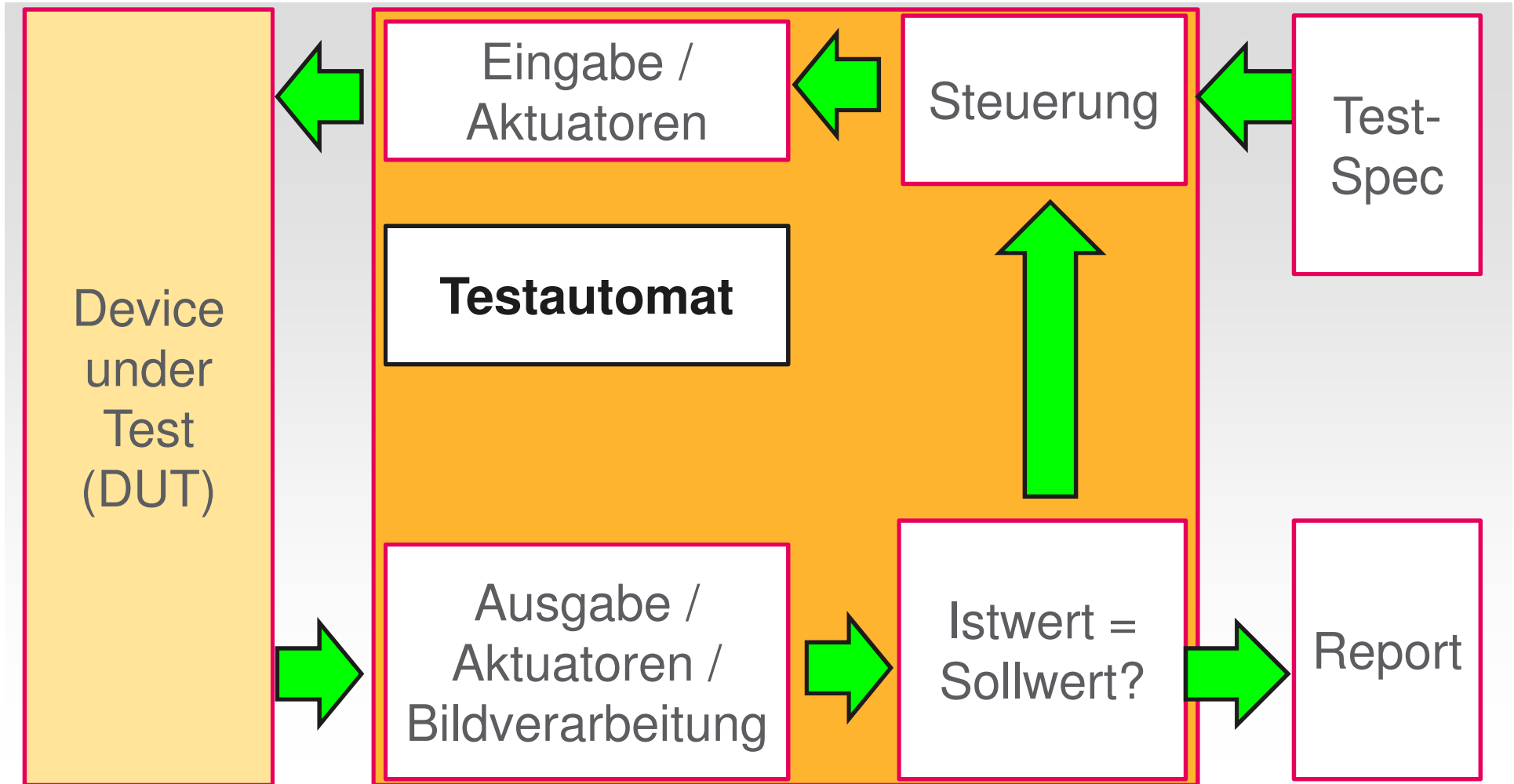
ebenen

- CAN-Bus
- Flexray
- LIN-Bus

Kriterien zur Analyse der Requirements

- ◆ Architektur
- ◆ Attribute
- ◆ Kritikalität
(Risiko-Analyse, FuSi und FMEA)
- ◆ Priorität
- ◆ Funktionalität

Modulare Architekturen - Testumgebung (Auszug)

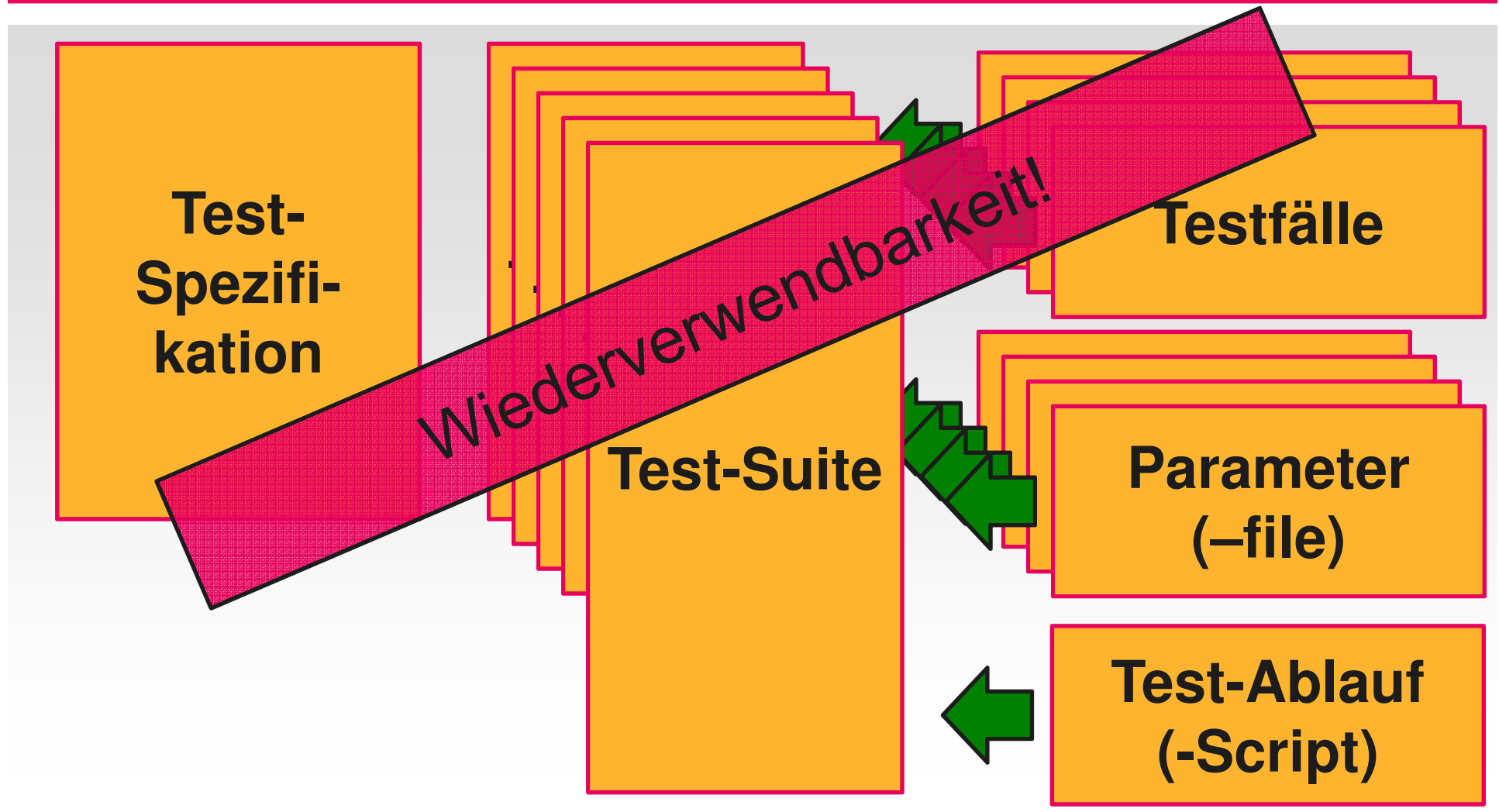


Modulare Architekturen

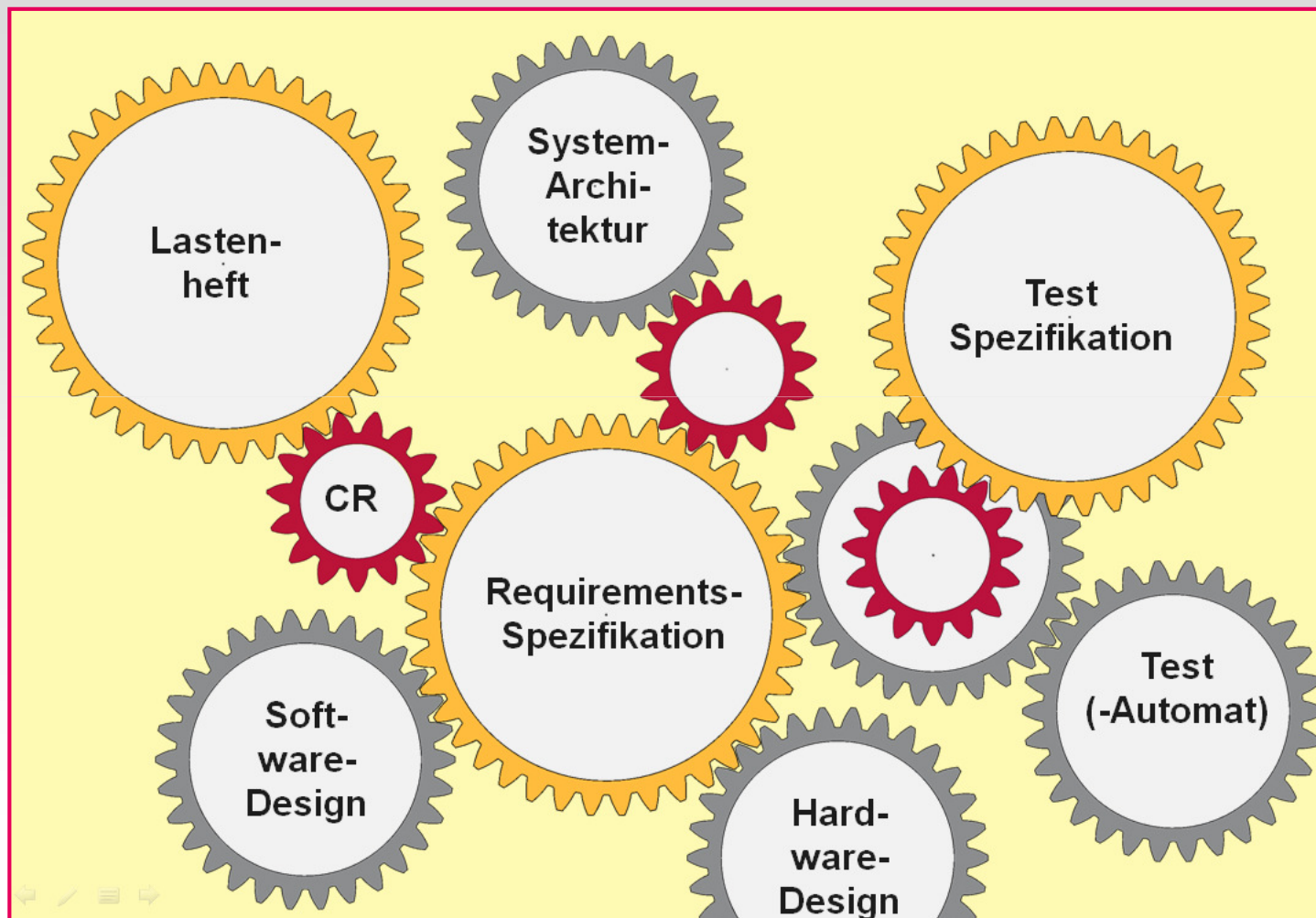
- Komponenten d. Testumgebung

- ◆ Test-Hardware
- ◆ Device under Test (DUT)
- ◆ Test-Software zur Durchführung der Tests
- ◆ Bei großen Datenmengen:
Software zur Auswertung der Testdaten
- ◆ Test-Suite für die Testfälle
- ◆ Requirements-Datenbank
- ◆ ...

Modulare Architekturen - Testmodularisierung



Modulare Architekturen - in den F&E Prozessen



Modulare Architekturen

- Einsatz der Testautomatisierung

Klassische Einsatzgebiete:

- ◆ Software-Tests
- ◆ Hardware-Test
- ◆ Systemintegrations-Tests
- ◆ System-Tests
- ◆ Umwelt-Tests

Modulare Architekturen

- Testplanung im Vorfeld

- ◆ Planung Testfallerstellung / Anpassung der Testfälle bei Wiederverwendung
- ◆ (Weiter-) Entwicklung Software für Testumgebung (Initialisierung)
- ◆ Spezielle Software für die Prüflinge notwendig??
- ◆ Entwicklung der Testsysteme
- ◆ Anzahl der Prüflinge
- ◆ Durchführung der Tests

Modulare Architekturen

- Testplanung im Vorfeld

- ◆ Planung der Ressourcen zur Testdurchführung
- ◆ Bereitstellung der Prüflinge (DUTS)
- ◆ Bereitstellung der Testsysteme
- ◆ Personal zur Vorbereitung / Durchführung Tests
- ◆ Personal zur Ausführung Tests
- ◆ Räume / Makelräume / Test-Ort
- ◆ Netzanbindung notwendig?
(z.B. Remote-Control / Abfrage akt. Teststatus usw.)

Es ist sinnvoll, die Testdurchführung und Entwicklung des automatisierten Tests aufzutrennen!

- ◆ Risikoklassen
Risiko-Analyse, FMEA, FuSi
- ◆ Priorität
- ◆ Geänderte
Module

Priorisierung von Fehlern

		Schadenshöhe		
		kein Schaden, Schönheitsfehler	Schaden	Großer Schaden
Auftreten von Problemen	wahrscheinlich	2	1	1
	möglich	3	2	1
	unwahrscheinlich	3	2	2

1	hohe Priorität
2	mittlere Priorität
3	niedrige Priorität

Methodenkompetenz

- Typische Testfälle aus der FuSi

Nachstellung von Fehlerszenarien aus FuSi, FMEA und Risiko-Analysen wie z. B.

- ◆ Fehlbedienung
- ◆ Fehler in der Buskommunikation (Flexray / CAN)
- ◆ Fehlerhafte Bestückung HW
- ◆ Fehlerhaft compilierte Software
- ◆ ...

Klassisch:

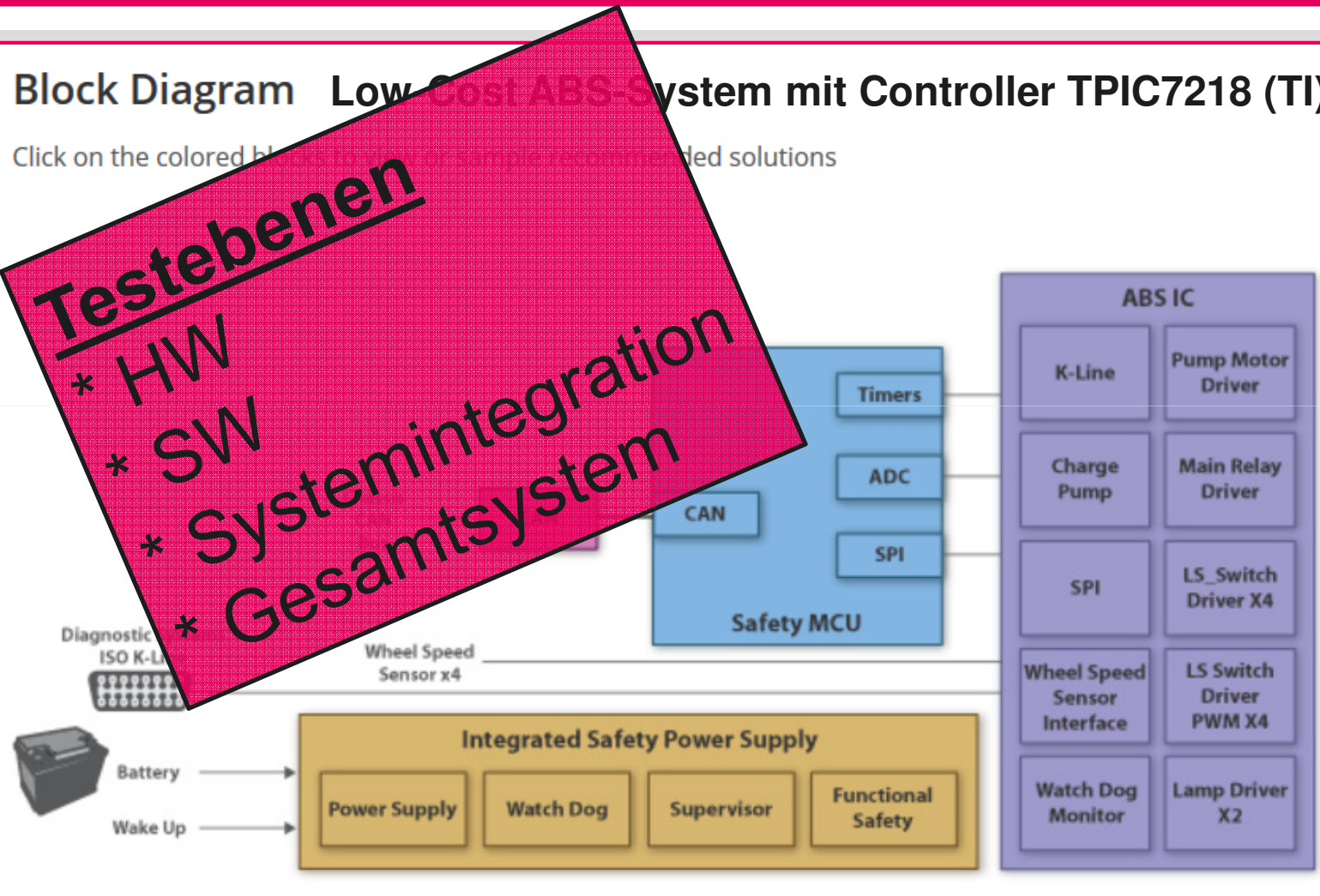
- ◆ Resetnetzwerk
- ◆ Openload
- ◆ Short to U_{Batt}
- ◆ Short to GND
- ◆ Jump Start
- ◆ Temperaturen
- ◆ EMV

Methodenkompetenz

- Typische Testfälle aus der FuSi

Block Diagram Low Cost ABS-System mit Controller TPIC7218 (TI)

Click on the colored blocks to see sample recommended solutions



 TEXAS INSTRUMENTS

Quelle:

http://www.ti.com/solution/antilock_braking_system?keyMatch=antiblockiersysteme&tisearch=Search-EN

Methodenkompetenz

- Testfallreduktion

- ◆ Klassifikations- Entscheidungsbaume
- ◆ Kombinatorik
- ◆ Zustandsbasiertes Testen
- ◆ Modellbasiertes Testen
- ◆ Auswahl der getesteten Fehlerarten
- ◆ Auswahl der Testebene
- ◆ Fehlerinjektion
- ◆ Kritische Punkte im System identifizieren

Darf ich noch Fragen beantworten?

Ingenieurbüro Paul Huber

Marlene-Dietrich-Str. 5

D-89231 Neu-Ulm

www.ing-buero-ph.de



Paul Huber

Mobil: 0176 / 208 46 334

Tel.: 0731 / 985 88 545

Mail: paul.huber@ing-buero-ph.de

Vorteile von automatisierten Tests

◆ Einsparung von Zeit

- Automatisierte Tests können unbeaufsichtigt ablaufen und binden während der Ausführung keine Mitarbeiter
- Automatisierte Tests sind beliebig oft wiederholbar. Bei Regressions- und Requalifizierungstests wird hier viel Zeit eingespart.
- Automatisierte Tests können außerhalb der üblichen Geschäftszeiten und rund um die Uhr ausgeführt werden..
- Die Dokumentation der Tests kann ebenfalls automatisiert erfolgen.

Vorteile von automatisierten Tests

- ◆ Einsparung von Entwicklungskosten
 - Die Test müssen nur einmal entwickelt werden
 - Wiederverwendung möglich

- ◆ Hohe örtliche Flexibilität
 - Test kann ortsunabhängig erfolgen

- ◆ Traceability
 - Automatisches einpflegen der Testergebnisse in die Datenbankstrukturen