

Institut für Softwaretechnologie

Universität Stuttgart  
Universitätsstraße 38  
D-70569 Stuttgart

Masterarbeit Nr. 91

# **Testabdeckungsmaße für den Integrationstest von E/E-Systemen im Fahrzeug**

Jaasiel Walter

<b>Studiengang:</b>	Softwaretechnik
<b>Prüfer/in:</b>	Prof. Dr. rer. nat. Stefan Wagner
<b>Betreuer/in:</b>	Asim Abdulkhaleq, M.Sc., Dipl.-Inf. Jens Herrmann
<b>Beginn am:</b>	18. April 2016
<b>Beendet am:</b>	18. Oktober 2016
<b>CR-Nummer:</b>	B.4.5



## **Danksagung**

An dieser Stelle möchte ich mich bei allen Personen bedanken, die mich durch das Studium begleitet und bei der Master-Thesis unterstützt haben.

Ich danke allen Kollegen der Daimler AG, die mir bei Fragen und Problemen stets kompetent und hilfreich zur Seite standen. Ein besonderer Dank gilt meinem Betreuer Dipl.-Inf. Jens Herrmann, der mich immer geduldig und hilfreich im Laufe meiner Masterarbeit unterstützt hat. Ebenfalls möchte ich mich bei meinem Betreuer an der Universität Stuttgart, Asim Abdulkhaleq, M.Sc., für die hervorragende Betreuung und Unterstützung und bei Prof. Dr. rer. nat. Stefan Wagner für die Übernahme der Prüfung bedanken.

Weiterhin danke ich meiner Familie und Azra, die stets zu mir standen, mich durch diese Masterarbeit begleitet und während des kompletten Studiums unterstützt haben.

## Kurzfassung

Die vorliegende Masterarbeit beschäftigt sich mit Testabdeckungsmaßen für den Integrationstest von E/E-Systemen. E/E-Systeme bestehen aus E/E-Komponenten (z.B. Steuergeräte, Sensoren und Aktoren) und sind fester Bestandteil bzw. Zukunftstrend in der Automobilindustrie. Beispielsweise befinden sich in einer modernen Mercedes-Benz S-Klasse je nach Ausstattung ca. 100 Steuergeräte. Komponenten der E/E-Systeme werden i.d.R. von Zulieferern isoliert entwickelt, getestet und als Black-Box an den Fahrzeughersteller (OEM) übergeben. Letzterer integriert die Komponenten zu Sub-/Systemen und testet deren Zusammenwirken. Oft wird zur Bewertung der Testgüte lediglich die Anforderungsabdeckung gemessen. Ziel dieser Arbeit war die Entwicklung von ergänzenden Testabdeckungsmaßen für den black-box-orientierten Integrationstest. Weiterhin sollte ein Konzept zur Nutzung der entwickelten Maße in einer erweiterten Teststrategie entworfen werden. Da es sich bei den E/E-Komponenten um Black-Boxes handelt, sind auch die beim Softwareintegrationstest einsetzbaren Testabdeckungsmaße wie Anweisungs-, Zweig- und Pfadüberdeckung nicht direkt einsetzbar. In der Literatur konnten keine veröffentlichten Testabdeckungsmaße zu diesem Thema ermittelt werden. Daher wurden aus dem Softwaretest bekannte Testabdeckungsmaße auf den Integrationstest von E/E-Systemen übertragen. Diese basieren auf Architektur- und Schnittstellenbeschreibungen, Zustandsautomaten und Kommunikationsmatrizen. Zur Evaluierung wurde der Einsatz der Testabdeckungsmaße anhand von Fallbeispielen demonstriert und durch Expertenbefragungen bewertet.

# Inhaltsverzeichnis

<b>Glossar und Abkürzungsverzeichnis</b>	<b>XI</b>
<b>1. Einleitung</b>	<b>1</b>
1.1. Motivation . . . . .	3
1.2. Problemstellung . . . . .	3
1.3. Zielsetzung . . . . .	4
1.4. Aufbau der Arbeit . . . . .	4
<b>2. Grundlagen</b>	<b>7</b>
2.1. E/E-Systeme in Fahrzeugen . . . . .	7
2.2. Was ist Testen? . . . . .	9
2.3. Integrationstest . . . . .	10
2.3.1. Der Integrationstest im V-Modell . . . . .	14
2.3.2. E/E-Integrationstest in Fahrzeugen . . . . .	15
2.4. Was sind Testabdeckungsmaße? . . . . .	16
2.5. Basen zur Ableitung von Testabdeckungsmaßen . . . . .	18
2.5.1. Zustandsautomaten . . . . .	18
2.5.2. Kommunikationsmatrix . . . . .	19
2.5.3. Architekturmodelle . . . . .	20
2.6. Anforderungsspezifikation . . . . .	21
<b>3. Analyse</b>	<b>23</b>
3.1. Stand der Wissenschaft und Technik . . . . .	23
3.2. Stand der Technik beim Original Equipment Manufacturer . . . . .	26
3.2.1. Testprozess beim Original Equipment Manufacturer . . . . .	26
3.2.2. Basen für die Ableitung von Testabdeckungsmaßen . . . . .	28
<b>4. Testabdeckungsmaße für den Integrationstest von E/E-Systemen im Fahrzeug</b>	<b>31</b>
4.1. Erweiterte Teststrategie . . . . .	31
4.2. Testabdeckungsmaße basierend auf Architektur- und Schnittstellenbeschreibungen . . . . .	33
4.3. Testabdeckungsmaße basierend auf Zustandsautomaten . . . . .	35
4.4. Testabdeckungsmaße basierend auf Kommunikationsmatrizen . . . . .	37
4.5. Abschließende Bemerkungen . . . . .	38

<b>5. Evaluierung</b>	<b>41</b>
5.1. Fallbeispiele: Anwendung der Testabdeckungsmaße . . . . .	41
5.2. Bewertung durch Experten . . . . .	47
<b>6. Zusammenfassung, Wertung und Ausblick</b>	<b>49</b>
<b>Anhang</b>	<b>52</b>
A. Fragebogen . . . . .	52
<b>Literaturverzeichnis</b>	<b>53</b>

# Abbildungsverzeichnis

1.1.	Vernetzte Steuergeräte im Fahrzeug . . . . .	1
1.2.	Beteiligte Komponenten einer typischen Kommunikationsstruktur in Fahrzeugen	2
1.3.	Zeitungsausschnitt zum Thema Takata-Airbags . . . . .	3
2.1.	Trendvergleich des Anteils der E/E-Systeme im Fahrzeug in den Jahren 2005 und 2015 . . . . .	8
2.2.	E/E-Komponenten und deren Funktion . . . . .	8
2.3.	Ablauf funktionaler Test (Black-Box-Sicht) . . . . .	10
2.4.	Ablauf struktureller Test (White-Box-Sicht) . . . . .	10
2.5.	Klassifikation von Integrationstestverfahren . . . . .	13
2.6.	E/E-V-Modell inklusive Prüftätigkeiten . . . . .	14
2.7.	Beispiel: HiL-Prüfstand . . . . .	15
2.8.	Funktionsweise eines HiL-Prüfstands . . . . .	16
2.9.	Zustandsautomat des Autonavigationssystems . . . . .	17
2.10.	Zustandsautomat eines Telefons . . . . .	18
2.11.	Aufbau einer K-Matrix . . . . .	19
2.12.	Modellierung einer E/E-Architektur auf vier Ebenen . . . . .	21
3.1.	Von der Analyse zur Integration . . . . .	26
3.2.	Testprozess nach ISTQB . . . . .	27
3.3.	Allgemeiner Testprozess . . . . .	28
4.1.	Erweiterung der allgemeinen Teststrategie . . . . .	32
4.2.	Komponentenpaare . . . . .	34
4.3.	Schnittstellen . . . . .	34
4.4.	Abdeckung der Zustände und Zustandsübergänge in einem Zustandsautomaten	36
5.1.	Architekturmodell zum Fallbeispiel „Einsatz der Testabdeckungsmaße auf Basis von Architektur- und Schnittstellenbeschreibungen“ . . . . .	43
5.2.	Zustandsautomat zum Fallbeispiel „Einsatz der Testabdeckungskriterien auf Basis von Zustandsautomaten“ . . . . .	44
5.3.	K-Matrix zum Fallbeispiel „Einsatz der Testabdeckungsmaße auf Basis von K-Matrizen“ . . . . .	46





# Tabellenverzeichnis

2.1.	Bekannte inkrementelle Integrationsstrategien . . . . .	12
2.2.	Übersicht über typische Integrationsstufen . . . . .	13
2.3.	Zustandsübergangstabelle des Autonavigationssystems . . . . .	17
3.1.	Testabdeckungsmaße aus der Literatur . . . . .	25
4.1.	Testabdeckungsmaße basierend auf Architektur- und Schnittstellenbeschreibungen . . . . .	35
4.2.	Testabdeckungsmaße basierend auf Zustandsautomaten . . . . .	37
4.3.	Testabdeckungsmaße basierend auf K-Matrizen . . . . .	38



# Glossar und Abkürzungsverzeichnis

Abkürzung	Bedeutung	Erstes Vorkommen
Aktor	Setzt elektrische Signale in mechanische Bewegung oder andere physikalische Größen um	8
CAN	Controller Area Network	2
CCP	CAN Calibration Protocol	15
COTS	Components-off-the-Shelf	23
E/E	Elektrik/Elektronik	1
ECU	Electronic Control Unit (auch: Steuergerät)	8
Integration	Die Zusammensetzung mehrerer Komponenten zu einem System	10
ISO26262	ISO-Norm für sicherheitsrelevante E/E-Systeme in Fahrzeugen (funktionale Sicherheit)	24
ISTQB	International Software Testing Qualifications Board	26
K-Matrix	Kommunikationsmatrix	19
KLH	Komponentenlastenheft	21
LIN	Local Interconnect Network	2
MC/DC	Modified Condition/Decision Coverage	23
MiL	Model-in-the-Loop	15
MOST	Media Oriented Systems Transport	2
OEM	Original Equipment Manufacturer	3
Platzhalter	Sonderimplementierung eines Moduls, das ein nicht verfügbares Modul ersetzt, falls dieses aufgerufen wird oder anderweitige Abhängigkeiten bestehen (auch: Stub)	11
RIF/ReqIF	Requirements Interchange Format	22
Sensor	Erfasst Messgrößen	8
SLH	Systemlastenheft	21
Treiber	Modul, das die Ausführung von anderen Modulen startet, steuert und überwacht (auch: Driver)	11
UML	Unified Modeling Language	24
XCP	Universal Measurement and Calibration Protocol	15



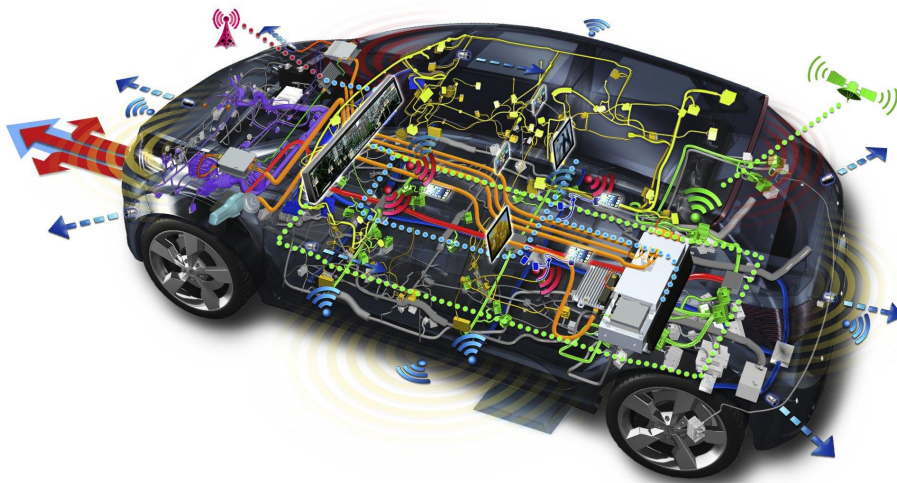
# 1. Einleitung

You can't control what you can't measure.

*(Tom DeMarco)*

Carl Friedrich Benz entwickelte 1886 den „Patent-Motorwagen Nummer 1“ und leitete damit die Geburtsstunde des Automobils ein. Damals war seine Funktionalität rudimentär ausgebildet. Das moderne Fahrzeug ist enorm weiterentwickelt und bietet elektronisch unterstützte Funktionen wie Fahrassistenzsysteme, Internetanbindung und Airbags.

Dies erfordert eine durchgehend komplexer werdende Steuerung bzw. Regelung der vernetzten Komponenten. Daher werden in heutigen Fahrzeugen immer mehr elektrische/elektronische Komponenten und Steuergeräte (Elektrik/Elektronik (E/E)-Systeme) integriert. Beispielsweise befinden sich in einer modernen Mercedes-Benz S-Klasse je nach Ausstattung ca. 100 Steuergeräte. Abbildung 1.1 zeigt die komplexe Integration von vernetzten E/E-Komponenten in heutigen Fahrzeugen.



**Abbildung 1.1.:** Vernetzte Steuergeräte im Fahrzeug [Del16]

Das folgende Beispiel soll die Bedeutung von E/E-Systemen und dem damit verbundenen Integrationstest veranschaulichen. Selbst an nur vier Grundfunktionalitäten im Fahrzeug sind

## 1. Einleitung

---

mehr als 20 Komponenten beteiligt, deren Zusammenwirken aufeinander abgestimmt und getestet werden muss.

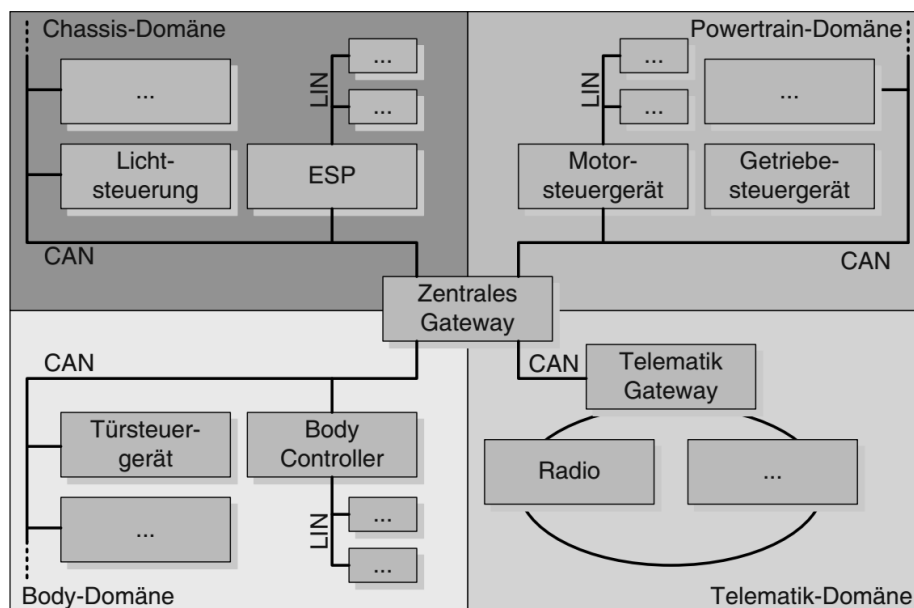
### Beispiel 1.1 (Kommunikationsstruktur)

Abbildung 1.2 zeigt die vereinfachte Darstellung einer typischen Kommunikationsstruktur als Bestandteil der E/E-Architektur von Fahrzeugen. Zu sehen sind vier Systeme, die aus E/E-Komponenten bestehen und über Controller Area Network (CAN)-, Local Interconnect Network (LIN) und Media Oriented Systems Transport (MOST)<sup>1</sup>-Busse verbunden sind. Die einzelnen Systeme wiederum sind über ein zentrales Gateway gekoppelt, das die Datenübertragung über unterschiedliche Bussysteme ermöglicht.

Unter Anderem umzusetzende Funktionen bzw. E/E-Systeme sind:

1. Lichtsteuerung
2. Türsteuerung
3. Motorsteuerung
4. Steuerung von Telematik-Funktionen

Anzahl der beteiligten Komponenten: > 20



**Abbildung 1.2.:** Beteiligte Komponenten einer typischen Kommunikationsstruktur in Fahrzeugen [ST12]

<sup>1</sup>MOST-Ring in der Telematik-Domäne der Abbildung

## 1.1. Motivation

Typischerweise entwickelt der Original Equipment Manufacturer (OEM) nicht alle Komponenten selbst. Es wird eher die Rolle des Integrators eingenommen. Zulieferer entwickeln und testen einzelne Komponenten isoliert und unter Annahmen über die vorgesehene Umgebung. OEMs integrieren diese anschließend zu einem System. Während der Integrationsphase ist es auch deren Aufgabe, Integrationstests durchzuführen, um das korrekte Zusammenwirken der Komponenten im Systemkontext zu verifizieren. Erst in dieser Phase ist ein Integrationstest unter realen Randbedingungen möglich.

Abbildung 1.3 ist zu entnehmen, welche weitreichenden Folgen unzureichend abgesicherte Integration haben kann. Takata-Airbags funktionieren als isoliertes System, nicht aber im Systemkontext. Hintergrund ist, dass Teile der Verkleidung durch zu kräftiges Auslösen des Airbags herausgeschleudert werden. Dieses Verhalten kann zu Verletzungen von Fahrzeuginsassen führen. Vorliegender Sachverhalt weist darauf hin, dass das Zusammenwirken von Airbag und Verkleidung bei der Entwicklung und insbesondere dem Test nicht genügend berücksichtigt wurde.

Es handelt sich um einen Fehler, den man spätestens während der Integration finden sollte; denn dieser wird nach der Zusammensetzung von Airbag und Fahrzeug sichtbar. Im vorhandenen Schadensfall ist der Fehler erst sehr viel später im Feld beim Kunden aufgetreten.



**Abbildung 1.3.:** Zeitungsausschnitt zum Thema Takata-Airbags [Han16]

## 1.2. Problemstellung

Der Integrationstest von E/E-Systemen konzentriert sich einerseits auf das korrekte Zusammenwirken von bereits isoliert getesteten E/E-Komponenten. Andererseits ist es das Ziel, mögliche Fehler im integrierten System aufzudecken. Ausgehend von den Anforderungen an das E/E-System sollen Fehler bei der Kommunikation zwischen E/E-Komponenten gefunden werden. Fehlfunktionen in der Interaktion sollen frühzeitig erkannt werden. Häufig werden keine anderen Testabdeckungsmaße als die Anforderungsabdeckung an das zu integrierende Sub-/System angewendet. Da es sich bei den E/E-Komponenten um Black-Boxes handelt, sind auch die beim Softwareintegrationstest einsetzbaren Testabdeckungsmaße wie Anweisungs-, Zweig- und Pfadüberdeckung nicht direkt einsetzbar.

Als Ergänzung zum anforderungsbasierten, funktionsorientierten Integrationstest sind in dieser Arbeit Testabdeckungsmaße zu definieren, die sich auf das Zusammenwirken von

E/E-Komponenten konzentrieren. Diese Testabdeckungsmaße sollen als Zusatz- und Testenkriterien hinsichtlich der Vollständigkeit des Integrationstests von E/E-Systemen dienen. Grundlage für die Entwicklung der Testabdeckungsmaße sind die zum Testobjekt verfügbaren Anforderungs-, Entwicklungs- und Testdokumente (z.B. Testkonzept).

### 1.3. Zielsetzung

Grundlage für diese Masterarbeit ist die Literaturrecherche und die damit verbundene Analyse und Bewertung der existierenden Testabdeckungsmaße für den Integrationstest von E/E-Systemen. Der Stand der Wissenschaft und Technik umfasst Literatur in der Forschung, in der industriellen Praxis und beim OEM. Ergebnis soll eine Auflistung der recherchierten Literatur und die darin enthaltenen Testabdeckungsmaße sein. Kern der Arbeit ist die Entwicklung der Testabdeckungsmaße, die beim Integrationstest von E/E-Systemen zur Bewertung der Vollständigkeit der Testfälle eingesetzt werden können. Zudem soll ein Konzept erarbeitet werden, das die methodische Eingliederung in den bestehenden Integrationstestprozess ermöglicht. Abschließend soll zur Veranschaulichung an Fallbeispielen demonstriert werden, wie die entwickelten Testabdeckungsmaße im Testprozess genutzt werden können. Dies umfasst auch eine Bewertung der Ergebnisse dieser Masterarbeit durch Expertenbefragungen. Im Folgenden sind die Ziele aufgelistet:

- Ziel 1:** Der Stand der Wissenschaft und Technik zum Thema „Testabdeckungsmaße für den Integrationstest von E/E-Systemen“ wird analysiert, zusammengefasst und bewertet.
- Ziel 2:** Testabdeckungsmaße für den Integrationstest von E/E-Systemen werden für den Einsatz beim OEM entwickelt.
- Ziel 3:** Ein Konzept zur methodischen Eingliederung der Testabdeckungsmaße in den bestehenden Testprozess wird entwickelt.
- Ziel 4:** Zur Evaluierung der entwickelten Testabdeckungsmaße wird anhand von Fallbeispielen und Expertenbefragungen demonstriert, inwieweit die Testabdeckungsmaße einen Mehrwert für den Integrationstest bringen werden.

### 1.4. Aufbau der Arbeit

Diese Masterarbeit ist wie folgt gegliedert:

Kapitel 1 (Einleitung) beschreibt die zugrunde liegende Motivation, die Besonderheiten der Problemstellung dieser Arbeit und die zu erreichenden Ziele.

Kapitel 2 (Grundlagen) beschreibt die grundlegenden Konzepte und Begrifflichkeiten des Integrationstests von E/E-Systemen. Es wird erklärt, worum es sich bei E/E-Systemen handelt



und was Testen und Integrationstest bedeutet. Weiterhin wird auf das V-Modell in der Automobilindustrie, Testabdeckungsmaße und Basen zur Ableitung von Testabdeckungsmaßen eingegangen.

Kapitel 3 (Analyse) stellt die Ergebnisse der Literaturrecherche zum Thema Testabdeckungsmaße für den Integrationstest von E/E-Systemen vor. Des Weiteren ist beschrieben, wie der Testprozess in diesem Zusammenhang beim OEM gestaltet ist und welche Testabdeckungsmaße verwendet werden. Abschließend werden relevante Basen für die Ableitung von Testabdeckungsmaßen aufgelistet.

Kapitel 4 (Testabdeckungsmaße für den Integrationstest von E/E-Systemen im Fahrzeug) erläutert, wie die allgemeine Teststrategie durch die im Rahmen dieser Arbeit entwickelten Testabdeckungsmaße für den Integrationstest von E/E-Systemen erweitert werden kann. Ferner ist beschrieben, welchen Zweck die zusätzlichen Testabdeckungsmaße, in Bezug zu den aus den Anforderungen abgeleiteten Testfälle der allgemeinen Teststrategie, erfüllen. Abschließend sind die entwickelten Testabdeckungsmaße beschrieben, die auf Architektur- und Schnittstellenbeschreibungen, Zustandsautomaten und Kommunikationsmatrizen basieren.

Kapitel 5 (Evaluierung) beschreibt den Einsatz ausgewählter Testabdeckungsmaße anhand von vereinfachten Fallbeispielen aus dem realen Umfeld der Automobilindustrie. Zur Bewertung des Potentials der entwickelten Testabdeckungsmaße für den Integrationstest von E/E-Systemen sind die Ergebnisse der Expertenbefragungen zusammengefasst, die mit Hilfe von Fragebögen durchgeführt wurden.

Kapitel 6 (Zusammenfassung, Wertung und Ausblick) fasst die Ergebnisse der Arbeit zusammen, bewertet sie und gibt einen Ausblick auf weiterführende Arbeiten.



## 2. Grundlagen

In diesem Kapitel werden die grundlegenden Konzepte und Begrifflichkeiten des Integrations-tests von E/E-Systemen beschrieben. Es wird erklärt, worum es sich bei E/E-Systemen handelt und was in diesem Kontext Testen und speziell Integrationstest bedeutet. Weiterhin wird auf das V-Modell in der Automobilindustrie, Testabdeckungsmaße und Basen zur Ableitung von Testabdeckungsmaßen eingegangen.

### 2.1. E/E-Systeme in Fahrzeugen

Abbildung 2.1 vergleicht die Entwicklung des Anteils der E/E-Systeme in Fahrzeugen in den Jahren 2005 und 2015. Deutlich zu erkennen ist, dass immer mehr E/E-Komponenten im Verbund als E/E-Systeme Funktionen realisieren, die zuvor rein mechanisch umgesetzt wurden oder Innovationen darstellen. Abbildung 1.1 in Kapitel 1 zeigte bereits die enorme Komplexität von E/E-Systemen im Fahrzeug. Das korrekte Zusammenwirken der E/E-Komponenten ist eine wichtige Voraussetzung, damit das Gesamtsystem funktionieren kann. E/E-Komponenten liegen als elektrische oder elektronische Bauteile vor und bestehen aus Hardware, Software oder deren Kombination. Letzteres wird auch als „Eingebettetes System“ bezeichnet und ist üblicherweise in der Automobilbranche vertreten. Die Schnittstellen der Systeme bestehen aus elektronischen, aber auch mechanischen Teilen. Über die Schnittstellen findet die Kommunikation der Komponenten statt, die bei E/E-Architekturen durch Bussysteme wie CAN, MOST, LIN, FlexRay bzw. neuerdings auch Ethernet verbunden sind.

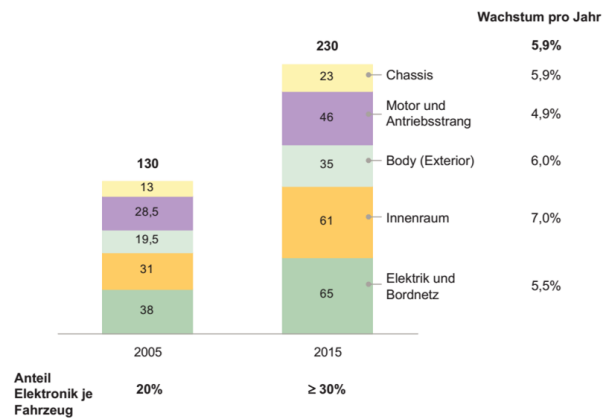
#### **Definition 2.1.1 (E/E-Komponente)**

*Eine E/E-Komponente kann ein Steuergerät, Sensor oder Aktor sein. Sie besteht aus Hardware, Software oder einer Kombination von Hardware und Software.*

#### **Definition 2.1.2 (E/E-System)**

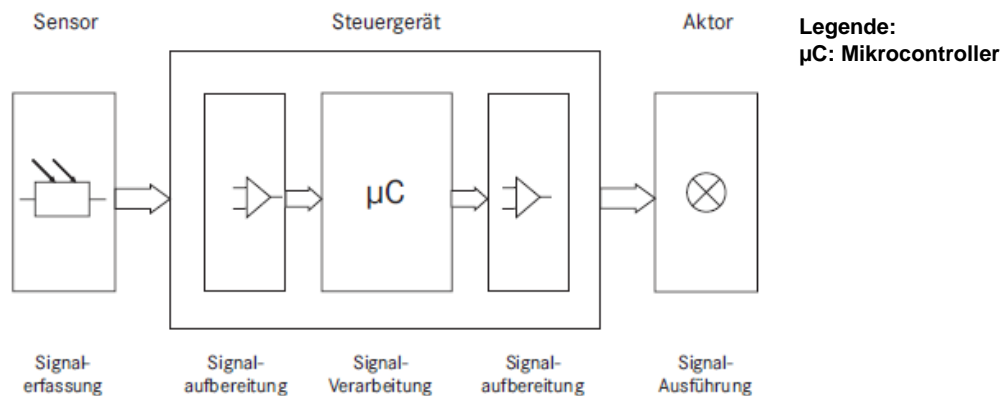
*Bei einem E/E-System handelt es sich um ein System, das aus elektrischen und/oder elektronischen Komponenten besteht. Mit inbegriffen sind programmierbare elektronische Elemente. [ISO11b]*

## 2. Grundlagen



**Abbildung 2.1.:** Trendvergleich des Anteils der E/E-Systeme im Fahrzeug in den Jahren 2005 und 2015 [Nör12]

Abbildung 2.2 zeigt den Aufbau einer Electronic Control Unit (auch: Steuergerät) (ECU) und die daran beteiligten, vorgeschalteten Sensoren (z.B. Radar-Sensor zur Messung des Abstandes) und nachgeschalteten Aktoren (z.B. zur Umwandlung elektrischer Signale in mechanische Bewegung). ECUs sind eingebettete Systeme, die mit ihrem Umfeld kommunizieren. Sensoren erfassen Daten und leiten diese an die ECUs weiter. Aktoren reagieren auf Kontrollnachrichten ausgehend von ECUs und führen angeforderte Aktionen aus.



**Abbildung 2.2.:** E/E-Komponenten und deren Funktion [Nör12]

## 2.2. Was ist Testen?

### Definition 2.2.1 (Testen)

*Testen ist das Ausführen von Funktionen eines Programms in einer definierten Umgebung und Vergleichen der erzielten mit den erwarteten Ergebnissen, um festzustellen, inwieweit sich das Programm so verhält, wie es seine Spezifikation verlangt. [VDI93]*

Sinn und Zweck des Testens ist die Steigerung der Qualität und das damit verbundene Vertrauen in das zu testende Produkt. Es handelt sich dabei um stichprobenartiges Vorgehen. In diesem Zusammenhang stellte Edsger W. Dijkstra schon 1972 fest, dass durch Testen das Vorhandensein von Fehlern gezeigt, nie aber deren Abwesenheit bewiesen werden kann.

Grundsätzlich wird bei Testverfahren zwischen statischen und dynamischen Techniken unterschieden. Letztere werden weiterhin in funktionale und strukturelle Tests untergliedert. Zusätzlich werden auch die Begriffe Black-Box- und White-Box-Test verwendet, die aber nicht als Synonyme für funktionale und strukturelle Testverfahren zu verstehen sind.

### Definition 2.2.2 (Black-Box Test)

*Testverfahren, die Testobjekte ohne Einsicht in die interne Struktur behandeln, werden als Black-Box Test bezeichnet (z.B. funktionale Äquivalenzklassenbildung und Zufallstest).*

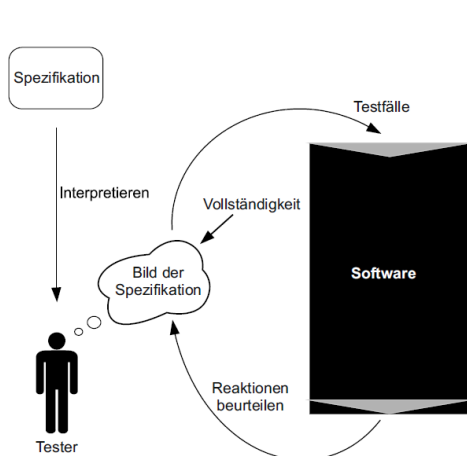
### Definition 2.2.3 (White-Box Test)

*Testverfahren, die Informationen über die Struktur der Software nutzen, werden als White-Box Test bezeichnet (z.B. kontrollfluss- und datenflussorientierte Tests).*

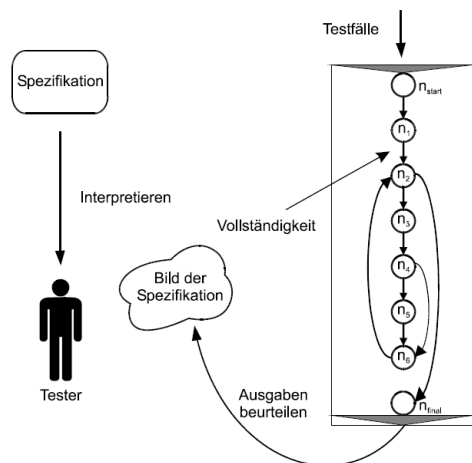
Statische Verfahren werden ohne Ausführung der zu prüfenden Software durchgeführt (z.B. Reviews und Inspektionen). Dies erlaubt jedoch keine vollständigen Aussagen über Korrektheit oder Zuverlässigkeit des zu testenden Produkts unter realen Randbedingungen. Dynamische Techniken werden mit Ausführung der zu prüfenden Software in Verbindung mit konkreten Eingabe- und erwarteten Ausgabewerten durchgeführt. Dadurch wird Testen in der realen Betriebsumgebung ermöglicht. Die Korrektheit des zu testenden Produkts kann jedoch nicht bewiesen werden. Durch Testen kann maximal gezeigt werden, dass Fehler in bestimmten Fällen bzw. Stichproben vorhanden sind.

Abbildung 2.3 zeigt den schematischen Ablauf funktionaler, Abbildung 2.4 den Ablauf struktureller Tests. Bei funktionalen Tests liegt das Testobjekt i.d.R. als Black-Box vor. Als Referenz für Vollständigkeit und Korrektheit dient die Spezifikation der Anforderungen (siehe Kapitel 2.6). Der Tester führt Testfälle aus, die anhand der Spezifikation erstellt wurden. Die Ergebnisse werden ausgehend von dem laut Spezifikation erwarteten Verhalten ausgewertet. Strukturelle Tests beziehen sich üblicherweise auf Strukturelemente der internen Abläufe (Quellcode). Das Testobjekt liegt demzufolge als White-Box vor. Aussagen hinsichtlich der Vollständigkeit werden in Bezug zur Abdeckung der Strukturelemente (z.B. Anweisungs-, Zweig-, Pfadabdeckung) getroffen. Die Ermittlung der Korrektheit ist, wie auch bei funktionalen Tests, abhängig von der Spezifikation. (Vgl. [Lig09]).

## 2. Grundlagen



**Abbildung 2.3.:** Ablauf funktionaler Test (Black-Box-Sicht) [Lig09]



**Abbildung 2.4.:** Ablauf struktureller Test (White-Box-Sicht) [Lig09]

Des Weiteren existieren verschiedene Testfallermittlungsverfahren. Im Folgenden wird eine für diese Arbeit relevante Auswahl beschrieben (vgl. [BSB08]).

**Funktionale Äquivalenzklassenbildung:** Ziel ist das Finden von Stellvertretern einer Gesamtmenge. Dadurch wird eine große Anzahl potentieller Testfälle durch Auswahl aus Äquivalenzklassen minimiert. Eingaben, die gleiches funktionales Verhalten hervorrufen werden in gleiche Klassen eingeteilt.

**Grenzwertanalyse:** Grenzwerte decken häufig Fehler auf. Basierend auf einer Äquivalenzklassenbildung können die Grenzwerte der einzelnen Klassen zur Testfallermittlung genutzt werden.

## 2.3. Integrationstest

Das IEEE Standard Glossary of Software Engineering Terminology definiert Integration wie folgt:

### Definition 2.3.1 (Integration)

*Integration bezeichnet das Verfahren der Kombination von Software-Komponenten, Hardware-Komponenten oder beidem zu einem Gesamtsystem [IEEE90].*

Bei der Integration geht es somit um die Zusammensetzung von Komponenten zu Sub-/Systemen. Zum Verfahren der Integration gibt es verschiedene Strategien. Neben dem „Big Bang“-Verfahren (alle Komponenten auf einmal zusammensetzen) existieren auch inkrementelle Integrationsverfahren (Komponenten werden schrittweise zusammengesetzt), die je nach

Voraussetzung gewählt werden. Zum Zeitpunkt der Integration liegen nicht immer alle Komponenten als reales Bauteil vor, daher ist z.T. eine Simulation mancher Komponenten und Funktionalitäten erforderlich. Je nach Integrationsverfahren kommen dabei Platzhalter und Treiber zum Einsatz. Generell sind inkrementelle Integrationsstrategien zu bevorzugen, da sie die Fehler bzw. Fehlerursachen lokalisieren. Eine Fehlerbehebung wird dadurch besser unterstützt als durch nicht-inkrementelle Integrationsansätze.

Erwarten bereits integrierte Komponenten eine bestimmte Funktionalität von noch nicht integrierten Komponenten, so kommen Platzhalter zum Einsatz. Diese werden so implementiert, dass für die Ausführung notwendige Daten an die aufrufende, integrierte Komponente gesendet werden. Platzhalter werden in realistische und spezifische Platzhalter kategorisiert. Erstere simulieren weitgehend die komplette Funktionalität der realen Komponente und können daher in vielen Testfällen genutzt werden. Ein Nachteil ist, dass diese Umsetzung aufwändig und anfällig für Fehler ist. Letztere stellen nur rudimentäre Funktionalität bereit, sind dafür weniger fehleranfällig, aber auch in wenigen Testfällen einsetzbar. Im Gegensatz dazu dienen Treiber zum Anstoßen der Funktionalität der einzelnen Komponenten. Sie können automatisiert als Testfall oder manuell vom Tester ausgeführt werden. (Vgl. [WES+13]).

Tabelle 2.1 listet die gängigsten inkrementellen Integrationsstrategien auf. In der ersten Spalte der Tabelle sind die Verfahren zu sehen, in der zweiten Spalte werden diese kurz beschrieben. Spalte drei ordnet dem jeweiligen Verfahren einen Typ zu. Eine nicht-inkrementelle Big-Bang-Integration kann bei der Durchführung eines Smoke-Tests<sup>1</sup> Sinn machen, um zu prüfen, ob ein System prinzipiell läuft. Dieses Testverfahren ist jedoch nicht für eine weitergehende Analyse des Systems geeignet.

---

<sup>1</sup>Smoke-Test: Test der Hauptfunktionalitäten ohne Berücksichtigung von Details

## 2. Grundlagen

---

Verfahren	Beschreibung	Typ
Top-Down	Beginnend mit den hierarchisch am weitesten „oben“ befindlichen Bausteinen wird das System zusammengesetzt. Noch nicht integrierte Bausteine werden durch Platzhalter simuliert.	Strukturabhängig
Bottom-Up	Beginnend mit Basismodulen ohne Abhängigkeiten, wird das System zusammengesetzt. Nicht integrierte Komponenten werden durch Testtreiber ersetzt.	Strukturabhängig
Sandwich	Kombiniert die beiden Ansätze Bottom-Up und Top-Down	Strukturabhängig
Terminabhängig	Komponenten werden je nach Verfügbarkeit integriert.	Funktionsorientiert
Risikobasiert	Risikobehaftete Komponenten werden mit Vorrang integriert.	Funktionsorientiert
Testgetrieben	Komponenten werden ausgehend von den spezifizierten Testfällen integriert.	Funktionsorientiert
Use-Case-abhängig	Komponenten werden ausgehend von den spezifizierten Anwendungsfällen integriert.	Funktionsorientiert

**Tabelle 2.1.:** Bekannte inkrementelle Integrationsstrategien (vgl. [WES+13])

### **Definition 2.3.2 (Integrationstest I)**

*Der Integrationstest testet und beurteilt die Interaktion zwischen integrierten Komponenten. Ziel ist die Sicherstellung der korrekten Funktion innerhalb des Gesamtsystems. (Vgl.[IEEE90], [ISO10]).*

Da es sich beim Testen immer um ein stichprobenartiges Vorgehen handelt, kann dadurch keine korrekte Funktion bewiesen werden. Es handelt sich daher eher um einen Wunsch als ein Ziel. Ausgehend von den Anforderungen können aber gezielt Fehler gesucht und wie in der folgenden Definition beschrieben, das Vertrauen in die Korrektheit erhöht werden.

### **Definition 2.3.3 (Integrationstest II)**

*Der Integrationstest prüft die Abhängigkeiten zwischen Komponenten. Ziel ist, mögliche Fehler im Zusammenspiel der Komponenten aufzudecken und das Vertrauen in dessen korrekte Funktionsweise zu erhöhen [SL10].*

Für diese Arbeit wird eine Kombination von beiden Definitionen verwendet. In Summe geht es darum, Vertrauen in die Korrektheit des Zusammenwirkens von Komponenten zu gewinnen und darin enthaltene Fehler aufzudecken.

Integrationstests können auf unterschiedlichen Integrationsstufen durchgeführt werden. Tabelle 2.2 listet einige Möglichkeiten auf. Die erste Spalte benennt die jeweilige Integrationsstufe.

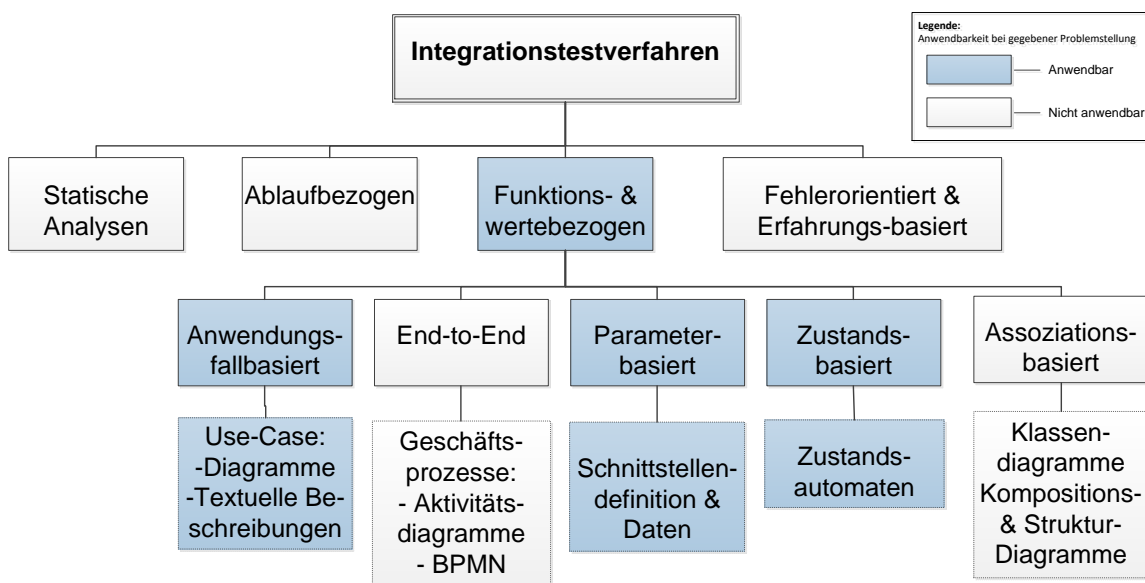


Die zweite Spalte beschreibt, welche Testobjekte auf der jeweiligen Stufe zur Ausführung gebracht werden.

Integrationsstufe	Testobjekte
Member	Das Zusammenwirken der Methoden und Variablen.
Klassen bzw. Modul	Das Zusammenwirken der Klassen und Module, bzw. deren Objekte innerhalb einer Komponente.
Komponenten- bzw. Subsystem	Das Zusammenwirken mehrerer Komponenten bzw. Subsysteme im Hinblick auf die Schnittstellen und die Kommunikation (Nachrichten/Ereignisse) zwischen Komponenten.

**Tabelle 2.2.:** Übersicht über typische Integrationsstufen (vgl. [WES+13])

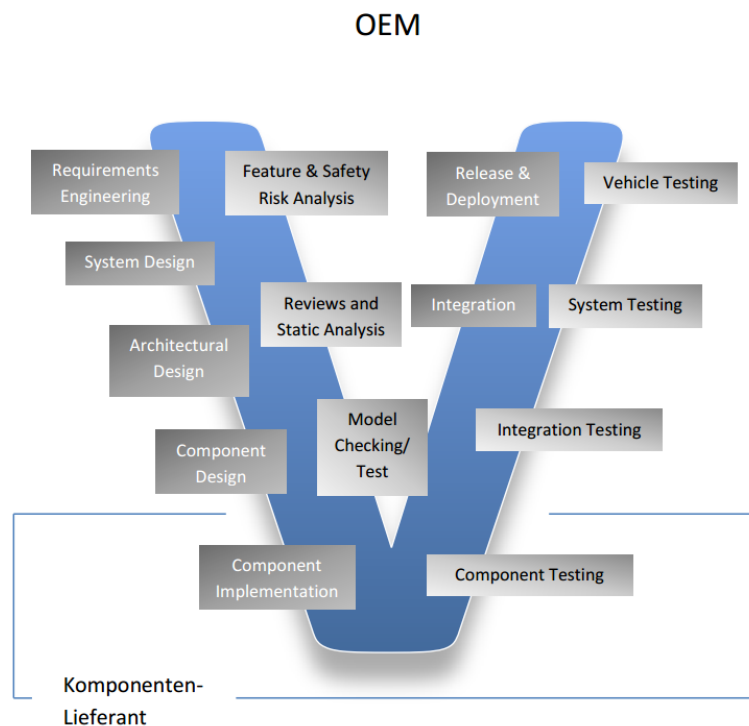
Abbildung 2.5 zeigt verschiedene Integrationstestverfahren und verdeutlicht die im Kontext dieser Arbeit anwendbaren Verfahren (blaue). Statische Analysen und ablaufbezogene Verfahren setzen die Möglichkeit voraus, den Quelltext zu betrachten. Fehlerorientiertes und erfahrungsbasiertes Vorgehen fokussiert sich auf bereits vorliegende Werte und zuvor durchgeführte Tests, werden in dieser Arbeit jedoch nicht weiter betrachtet. Funktions- und wertebezogene Verfahren hingegen eignen sich für Black-Box-Verfahren (z.B. Anwendungsfälle, Signalparameter und Zustandsautomaten).



**Abbildung 2.5.:** Klassifikation von Integrationstestverfahren (vgl. [WES+13])

### 2.3.1. Der Integrationstest im V-Modell

Das in der Softwareentwicklung bekannte V-Modell nach Boehm<sup>2</sup> hat sich auch in der Automobilindustrie durchgesetzt. Das V-Modell ist ein Vorgehensmodell, bei dem der Softwareentwicklungsprozess in Phasen strukturiert ist. Bei der Entwicklung von E/E-Systemen im Fahrzeug existiert zu jedem Entwicklungsschritt ein zugehöriger Prüfschritt. In Abbildung 2.6 ist das für die Entwicklung von E/E-Systemen üblicherweise verwendete V-Modell dargestellt. Entwicklungsschritte in der Konstruktionsphase sind durch graue, Prüfschritte durch hellblaue Kästen gekennzeichnet. Der OEM übernimmt den oberen Teil der Entwicklungs- und Testschritte im V-Modell. Die Komponentenlieferanten konzentrieren sich auf die Implementierung und den Test der einzelnen Komponenten. Der Integrationstest ist von großer Bedeutung, da erst hier das Zusammenwirken der gelieferten Komponenten im integrierten System unter realen Randbedingungen getestet werden kann.



**Abbildung 2.6.:** E/E-V-Modell inklusive Prüftätigkeiten

Der erste Entwicklungsschritt im Modell beinhaltet die Analyse und Spezifikation der Anforderungen, die die Funktionen des Systems definieren. Dem gegenüber steht der Prüfschritt der Machbarkeitsanalyse bzw. Risikoanalyse. Nächster Entwicklungsschritt umfasst den Entwurf des Systemdesigns, validiert durch statische Analysen und Reviews. Architektur- und Komponentenentwurf werden durch Model Checking (Verifikation des Modells gegen die

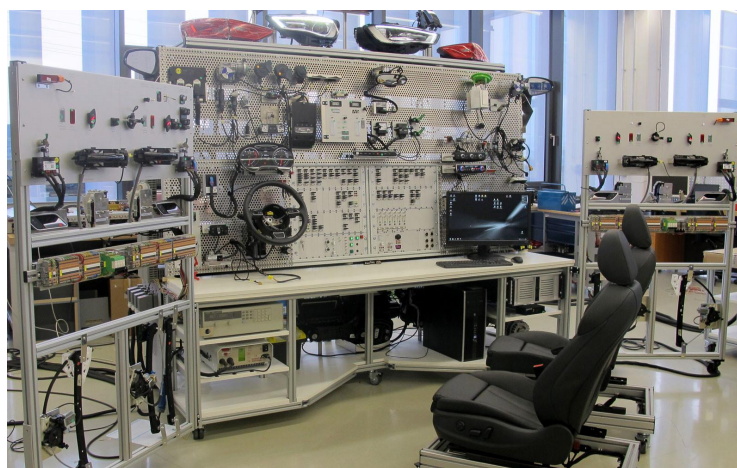
<sup>2</sup>Für eine Beschreibung des V-Modells siehe z.B. [BB83]

zugrundeliegende Spezifikation) und ersten Tests in einer virtuellen Integration geprüft. Die Implementierung der Komponenten findet beim Lieferanten statt und wird durch diesen getestet. Nach Auslieferung der Komponenten werden diese beim OEM integriert und durch Integrationstests abgesichert. Dies ist der Fokus dieser Masterarbeit. Der letzte Integrationsschritt beinhaltet die Komposition zu einem kompletten System. Der Einsatz des E/E-Systems im Fahrzeug wird abschließend durch Erprobungsfahrten, d.h. durch Tests im Fahrzeug getestet.

### 2.3.2. E/E-Integrationstest in Fahrzeugen

Neben Model-in-the-Loop (MiL) und Mockups ist eine typische Testplattform in Bezug zum Integrationstest von E/E-Systemen der Hardware-in-the-Loop (HiL)-Prüfstand. Dieser wird genutzt, um Sub-/Systeme bzw. Steuergeräteverbunde, aber auch einzelne Komponenten zu testen. Die Schnittstellen dieser Systeme liegen optisch, elektrisch oder mechanisch vor. HiL ermöglicht die Kombination von automatisierten Tests und Mensch-Maschine-Interaktionen. Fehlende Komponenten werden mittels Platzhaltern und Treibern simuliert. Je nach Aufwand der Verbauung können so auch Eigenschaften von Sensoren und Aktoren simuliert werden. Reale Kabelsätze, wie sie im Endprodukt vorkommen, werden durch dem HiL angepasste Testkabelsätze ersetzt. Zur Fehlersuche sind Schaltungen zur elektrischen Fehlersimulation vorgesehen. Bussysteme wie CAN können zur Nachrichtenauswertung genutzt werden, Protokolle wie das CAN Calibration Protocol (CCP) und das Universal Measurement and Calibration Protocol (XCP) dienen zur Diagnose und Manipulation der Daten.

Abbildung 2.7 zeigt einen beispielhaften Aufbau eines HiL-Prüfstands. Zu sehen sind verbaute E/E-Komponenten, spezialisierte Kabelsätze und ein Simulationsrechner der zum automatisierten Test dient. Der manuelle Eingriff kann in diesem Fall über ein integriertes Lenkrad erfolgen.



**Abbildung 2.7.:** Beispiel: HiL-Prüfstand [imc15]

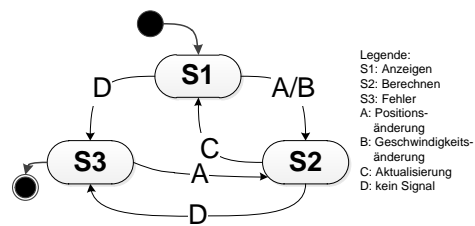


Das folgende fiktive Beispiel soll die Anwendung von Testabdeckungsmaßen veranschaulichen.

**Beispiel: Zustandsübergangsabdeckung (Transition Coverage) eines Autonavigationssystems**

Gegeben sei die Zustandsübergangstabelle 2.3 eines vereinfachten Autonavigationssystems mit den drei Zuständen  $S_1$ : Anzeigen,  $S_2$ : Berechnen und  $S_3$ : Fehler sowie den vier Zustandsübergängen A: Positionsänderung, B: Geschwindigkeitsänderung, C: Aktualisierung, D: kein Signal. Abbildung 2.9 zeigt den dazugehörigen Zustandsautomaten.

	A	B	C	D
S1	S2	S2	-	S3
S2	-	-	S1	S3
S3	S2	-	-	-



**Tabelle 2.3.:** Zustandsübergangstabelle des Autonavigationssystems [aio12]

**Abbildung 2.9.:** Zustandsautomat des Autonavigationssystems

Die sogenannte Zustandsübergangsabdeckung ist wie folgt definiert:

**Definition 2.4.1 (Zustandsübergangsabdeckung)**

Die Zustandsübergangsabdeckung bezeichnet den Grad der Ausführung von Zustandsübergängen im Verhältnis zu allen vorhandenen Zustandsübergängen (siehe Gleichung 2.1).

Die Abdeckung des Zustandsautomaten aus Abbildung 2.9 anhand des Testabdeckungsmaßes „Zustandsübergangsabdeckung“ wird durch die Abdeckung der Zustandsübergänge erreicht. Die prozentuale Abdeckung wird wie in Gleichung 2.1 berechnet.

$$\text{Überdeckungsgrad}_{\text{transition coverage}} = \frac{\# \text{ durchlaufene Zustandsübergänge}}{\# \text{ mögliche Zustandsübergänge}} \quad (2.1)$$

⇒ Mit einem Testfall, in dem jeder mögliche Zustandsübergang A, B, C, D mindestens einmal abgedeckt ist, wird eine vollständige Erfüllung des Testabdeckungsmaßes „Zustandsübergangsabdeckung“ erreicht.

## 2.5. Basen zur Ableitung von Testabdeckungsmaßen

### 2.5.1. Zustandsautomaten

Zustandsautomaten dienen als Spezifikation von Zuständen und Zustandsübergängen. Sie sind ein Beschreibungsmittel gedächtnisbehalteter Systeme und bestehen aus Zuständen und erlaubten Zustandsübergängen. Es existiert immer ein Start- und ein Endzustand. Zustandsübergänge können mit Bedingungen versehen sein, die je nach Auswertung, einen Zustandsübergang vom aktuellen Zustand in einen Folgezustand hervorrufen.

Abbildung 2.10 zeigt einen Zustandsautomaten, der die Funktionalität eines Telefons beschreibt. Beispielsweise kann nach dem Wählvorgang ein Zustandsübergang vom Zustand „Wählen“ in den Zustand „Verbunden“ erfolgen, falls der Zustandsübergang durch eine gültige Rufnummer bzw. einen Verbindungsaufbau durchlaufen wird.

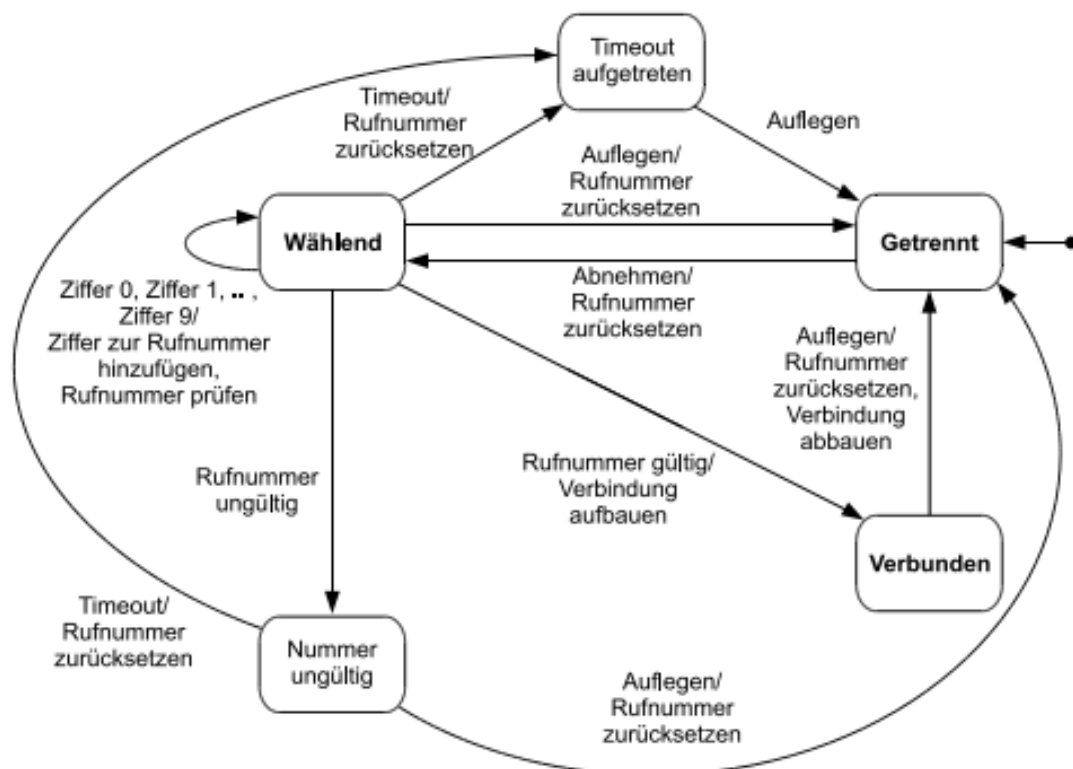


Abbildung 2.10.: Zustandsautomat eines Telefons [Lig09]

### 2.5.2. Kommunikationsmatrix

Eine Kommunikationsmatrix (K-Matrix) liefert eine tabellarische Übersicht über nachrichtenorientierte Kommunikationsbeziehungen, die in einem Sub-/System zwischen Komponenten auftreten können. In der Praxis kann eine solche K-Matrix auch grafisch vorliegen. Dargestellt sind die Kommunikationsmöglichkeiten als Sender-Empfänger-Beziehungen.

Abbildung 2.11 zeigt einen vereinfachten Auszug einer K-Matrix mit den drei Netzknoten ABS-Steuergerät, Motorsteuergerät, Getriebesteuergerät, den Nachrichten ABS\_1, ABS\_2, MS\_1, MS\_2, GS\_1 und den Signalen „Raddrehzahl vorne links“, „Raddrehzahl vorne rechts“, „Raddrehzahl hinten links“, „Raddrehzahl hinten rechts“, „Fahrpedalwert“, „Motordrehzahl“, „Motortemperatur“ und „Motorsollwert“. Über die Abkürzungen „S“ und „E“ ist definiert, welcher Netzknoten eine bestimmte Nachricht empfängt oder sendet. Ebenso ist das zugehörige Kommunikationsdiagramm im Fahrzeug abgebildet (siehe Bild oben).

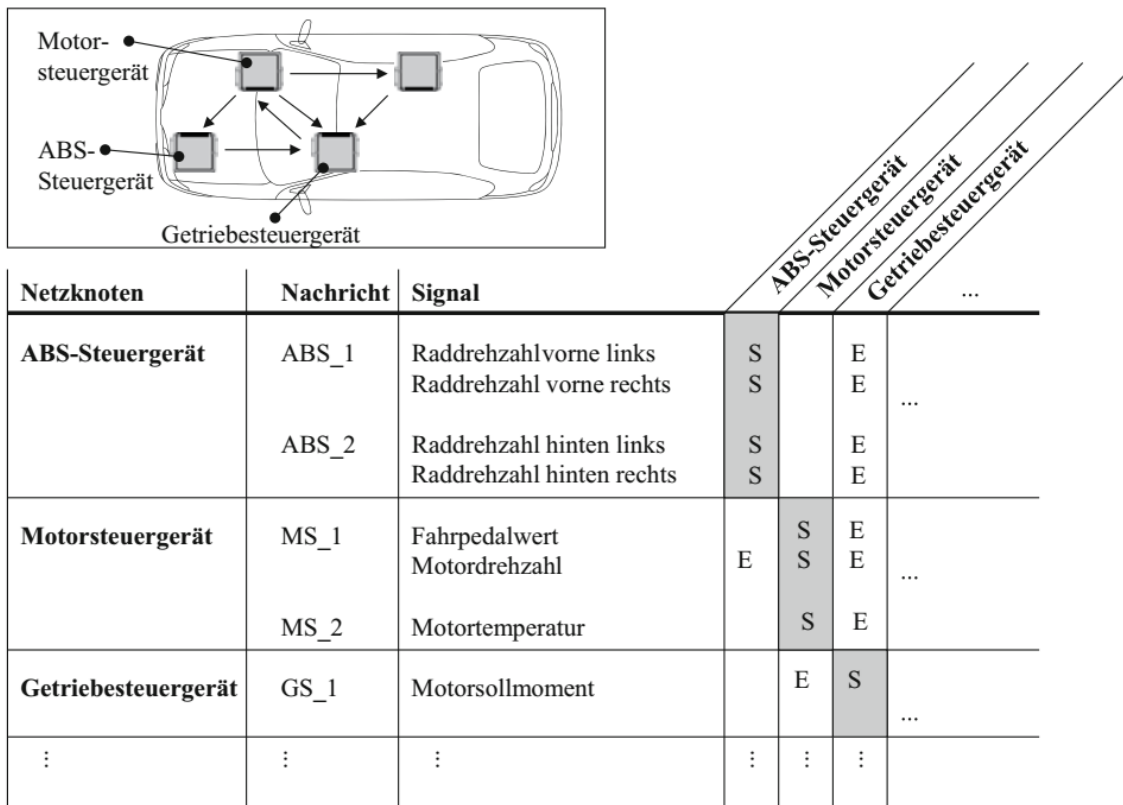


Abbildung 2.11.: Aufbau einer K-Matrix [SZ13]

Im Realfall ist die K-Matrix weitaus komplexer und enthält Informationen über den Sendezyklus der Nachrichten, Vorbedingungen, Signalattribute und weitere Parameter (vgl. [SZ13]). Zu beachten ist, dass die K-Matrix Kommunikationsbeziehungen ganzer Systeme beschreibt.

### 2.5.3. Architekturmodelle

Architekturmodelle von E/E-Systemen in Fahrzeugen beschreiben Komponenten und die damit verbundene Vernetzung, die zur Realisierung eines spezifizierten Systems benötigt werden. Dies wird auf verschiedenen Ebenen modelliert. Abbildung 2.12 zeigt die vier wesentlichen Ebenen bestehend aus der Beschreibung des Funktionsumfangs in Form von Anforderungen, der dazugehörigen Funktions- und Softwarearchitektur, Vernetzungsarchitektur und Komponententopologie.

Auf der Ebene des Funktionsumfangs werden die zu realisierenden Funktionen und deren Merkmale bzw. Anforderungen festgehalten. Dies umfasst sowohl kundenerlebbare (z.B. automatisches Fahrlicht), als auch nicht-kundenerlebbare Anforderungen (z.B. Kommunikationszeiten auf dem CAN-Bus). Ebenso werden von weiteren Funktionen abhängige Funktionen beschrieben und miteinander in Beziehung gesetzt.

Die Ebene der Funktions-/Softwarearchitektur stellt die Modellierung der zuvor definierten Funktionen dar. Es handelt sich um eine logische Architektur, also eine Abstraktion der zu realisierenden Funktionalität. Sie dient als Basis für die Entwicklung der Hardware und Software und spezifiziert daher Sensoren, Aktoren, Schnittstellen und Konnektoren auf funktionaler Ebene ohne umsetzungsrelevante Informationen vorzugeben. Mögliche Festlegungen beinhalten Typ der Kommunikation (z.B. Master-Slave) und Art der Daten, die zwischen E/E-Komponenten ausgetauscht werden können.

#### **Definition 2.5.1 (Schnittstellen)**

*Schnittstellen bezeichnen die Ein- und Ausgänge einer E/E-Komponente über die eine Kommunikation mit anderen Komponenten bzw. Systemteilen möglich ist. Sie definieren die Art des Ports, anliegende Daten und deren Datentypen.*

Die Ebene der Vernetzungsarchitektur listet alle beteiligten E/E-Komponenten auf, die zur Realisierung einer Funktion benötigt werden. I.d.R. wird diese Ebene in die vier Unterkategorien „Kommunikationsstruktur“, „Leistungsversorgung“, „Komponentenarchitektur“ und „Leitungssatz“ aufgeteilt. Die Kommunikationsstruktur modelliert die Vernetzung der Komponenten und die verwendeten Bussysteme. Die Komponentenarchitektur definiert die interne Struktur von E/E-Komponenten und deren benötigte Energieversorgung, die wiederum in einer eigenen Unterebene beschrieben wird. Auch der Leitungssatz wird separat beschrieben. Hier werden die Pin- und Leitungszuordnungen und die damit verbundenen Kabeltypen festgelegt. Die Komponententopologie spezifiziert die vorgesehenen Bauräume und Verbindungen im Fahrzeug.



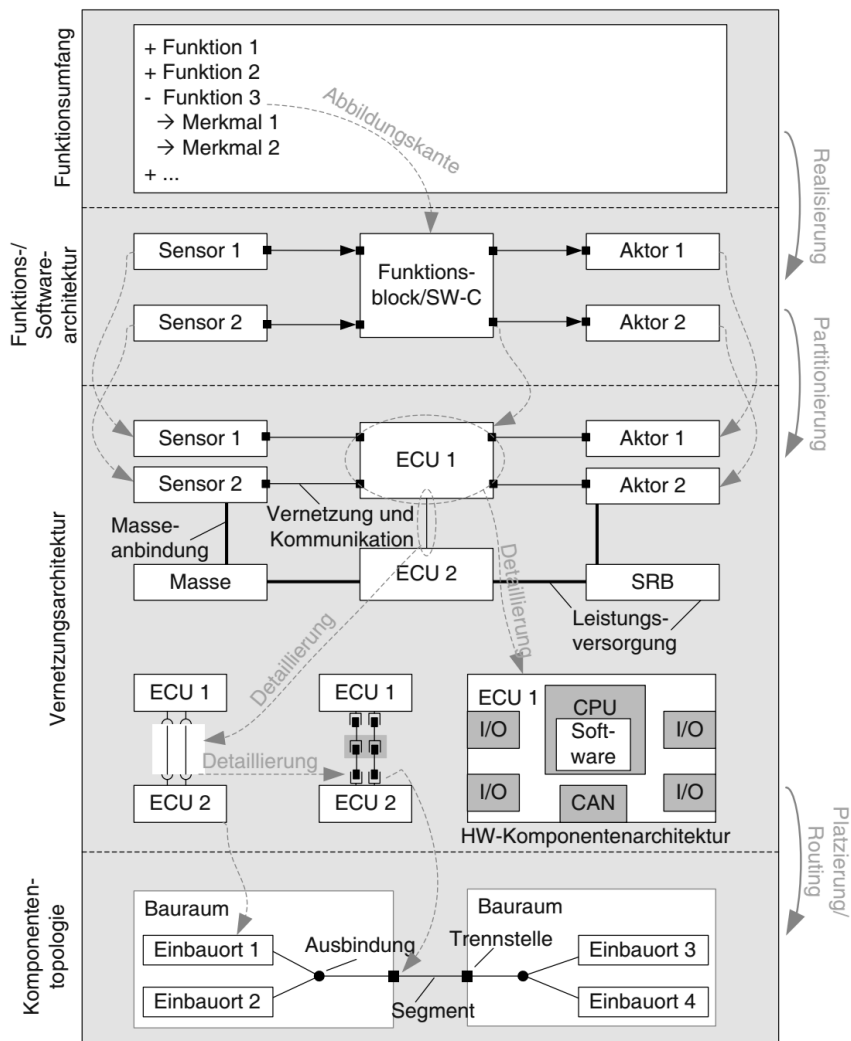


Abbildung 2.12.: Modellierung einer E/E-Architektur auf vier Ebenen [ST12]

## 2.6. Anforderungsspezifikation

Anforderungen an E/E-Systeme werden in einer Spezifikation strukturiert festgehalten. Zu einem System existiert i.d.R. eine Systemspezifikation, das sogenannte Systemlastenheft (SLH) und das Komponentenlastenheft (KLH), das wiederum integrierte Komponenten beschreibt (meist mehrere KLHs). Ein Lastenheft dient als Vertragsbasis zwischen OEM und Lieferanten, sowie als Entwicklungsbasis des Produkts. Des Weiteren werden die Anforderungen an das System bzw. die Komponente als Referenz zu Testzwecken genutzt. Minimalkriterium ist üblicherweise die Anforderungsabdeckung, d.h. zu jeder Anforderung existiert mindestens ein

## 2. Grundlagen

---

Testfall. Zur einheitlichen Dokumentation der Anforderungen in Spezifikationsdokumenten sind genormte Vorlagen vorhanden. Die folgenden Vorlagen finden Verwendung:

- IEEE Standard 1362-1998 Guide for Information Technology System Definition Concept of Operations Document (siehe [IEE98a])
- IEEE Standard 830-1998 Recommended Practice for Software Requirements Specifications (siehe [IEE98b])
- Volere Template (siehe [RR16])

In SLHs werden Hard- und Softwarekomponenten, Schnittstellenbeschreibungen und Funktionsverhalten im Gesamtsystem beschrieben. Hieraus werden Anforderungen an die einzelnen Komponenten abgeleitet, die wiederum in KLHs beschrieben werden. Wichtig ist hierbei die Nachverfolgbarkeit und Konsistenz von Anforderungen. Um diese Eigenschaften zu gewährleisten, werden oftmals Werkzeuge eingesetzt mit denen sich Lastenhefte verwalten lassen.

Es existiert eine Vielzahl an Werkzeugen zur Verwaltung von Anforderungsdokumenten. Im Automobilbereich ist u.a. „IBM Rational DOORS“ (siehe [IBM16]) im Einsatz. Es handelt sich um ein unterstützendes Werkzeug zum Verfassen, Archivieren und Verfolgen von Anforderungen. Anforderungen können dabei in einem Modul (z.B. Lastenheft/Pflichtenheft) zusammengefasst und mit Hilfe von Überschriften unterteilt werden (z.B. Einleitung, Glossar, Funktionale Anforderungen). Dabei werden die erstellten Anforderungen mit Systemattributen versehen (z.B. Zeitstempel, Ersteller und eindeutiger Identifier). Zusätzlich erlaubt DOORS das Erstellen von benutzerdefinierten Attributen (z.B. Priorität). Um Verfolgbarkeit zu gewährleisten, können Verlinkungen innerhalb von Modulen und über Modulgrenzen hinweg vorgenommen werden. Anforderungen können als Export für Dritte als Word-, Excel- oder CSV-Export zur Verfügung gestellt werden. Die Migration und der Austausch von Projekten erfolgt über standardisierte Formate (z.B. Requirements Interchange Format (RIF/ReqIF) (siehe [Gro16])).

## 3. Analyse

Dieses Kapitel stellt die Ergebnisse der Literaturrecherche zum Thema Testabdeckungsmaße für den Integrationstest von E/E-Systemen vor. Des Weiteren ist beschrieben, wie der Testprozess in diesem Zusammenhang beim OEM gestaltet ist und welche Testabdeckungsmaße verwendet werden. Abschließend werden relevante Basen für die Ableitung von Testabdeckungsmaßen für den Integrationstest aufgelistet.

### 3.1. Stand der Wissenschaft und Technik

Beim Testen von Software haben sich Testabdeckungsmaße zum Bewerten der Güte von Testfällen etabliert. Liggesmeyer [Lig09] listet die gängigsten Testabdeckungsmaße auf. Im Bereich der kontrollflussorientierten Testverfahren zählen hierbei Anweisungs-, Zweig-, Bedingungs- und Pfadabdeckung zu den bekanntesten Vertretern. In der Industrie hat sich in Bezug zur Bedingungsabdeckung auch das durch Kelly J. et al. [KDJL01] beschriebene Testabdeckungsmaß „Modified Condition/Decision Coverage (MC/DC)“ als höchstes Maß durchgesetzt. Diese Testabdeckungsmaße beziehen sich jedoch auf Software und deren strukturellen Aufbau. Sie sind daher nicht direkt auf den black-box-orientierten Integrationstest von E/E-Systemen anwendbar. Es müssen Testabdeckungsmaße speziell für diesen Anwendungsbereich entwickelt werden bzw. die bestehenden Maße auf eine höhere Ebene abstrahiert werden. Dabei gilt es den Fokus auf die Schnittstellen und das Zusammenwirken von Komponenten, folglich die Kommunikation zu legen.

Während der Literaturrecherche konnten keine konkreten wissenschaftlichen Arbeiten über Testabdeckungsmaße speziell für den Integrationstest von E/E-Systemen ermittelt werden. Es existieren jedoch Arbeiten bezüglich des Integrationstests von komponentenbasierter Software. Weiterhin existieren Arbeiten zum Integrationstest von Components-off-the-Shelf (COTS)-Systemen. Hierbei handelt es sich um „Soft- oder Hardware, die in Serienfertigung hergestellt und unverändert benutzt werden“ [ITW16].

Die im folgenden beschriebenen Arbeiten weisen Testabdeckungsmaße auf, die teilweise auch auf den Integrationstest von E/E-Systemen angewandt werden können (vgl. Tabelle 3.1). Voraussetzung ist immer die Verfügbarkeit einer geeigneten Spezifikation bzw. der zugrunde liegenden Basis (z.B. Zustandsautomaten, K-Matrix, Schnittstellenbeschreibungen).

Winter et al. [WES+13] erklären in ihrem umfassenden Werk Grundlagen, Methoden und Prozesse für den Integrationstest. Sie nennen Testabdeckungsmaße für den Integrationstest auf

### 3. Analyse

---

Basis von Zustandsautomaten, die sich auf die Zustände und Zustandsübergänge konzentrieren. Mit Zustandsübergängen mögliche verbundene Bedingungen werden jedoch nicht betrachtet. Die genannten Testabdeckungsmaße für den zustandsbasierten Test sind „Prozentsatz der im Test eingenommenen Zustände und Prozentsatz der im Test ausgeführten Zustandsübergänge, Prozentsatz der im Test angeregten, nicht spezifizierten Zustandsübergänge“. Weitere Testabdeckungsmaße, die für den black-box-orientierten Integrationstest von E/E-Systemen angewendet werden können, werden nicht genannt.

Nörenberg [Nör12] erläutert in seiner Dissertation zum Thema „Regressionstest von E/E-Systemen“ Testabdeckungsmaße, die auch für den Integrationstest von E/E-Systemen genutzt werden können. Relevante Testabdeckungsmaße sind: Anforderungsabdeckung, Zustandsabdeckung, Schnittstellenabdeckung, Anwendungsfallabdeckung. Überdies werden SW-Strukturabdeckung, Abdeckung der Sicherheitsmechanismen bzw. Fehlererkennungs-/Fehlerbehandlungsmechanismen, Variantenabdeckung genannt.

Power und McQuillan [PM05] liefern eine Übersicht über wissenschaftliche Arbeiten, die Testabdeckungsmaße für Softwaretests auf Basis von Unified Modeling Language (UML)-Beschreibungen aufweisen. Mögliche Testabdeckungsmaße für den Integrationstest basieren auf Kommunikationsdiagrammen, Zustandsautomaten und Anwendungsfällen. Weitere Informationen zu Anwendungsfällen gibt Winter [Win99]. Er definiert folgende Maße: Use Case Step Coverage, Use Case Branch Coverage, Use Case Scenario Coverage, Use Case Boundary Body Coverage, Use Case Path Coverage.

Wu, Pan und Chen [WPC01] stellen ein Modell vor, dass durch einen Component Interaction Graph (CIG) dargestellt wird. Anhand der Interaktionen und Abhängigkeiten, die aus dem CIG hervorgehen, werden Testkriterien abgeleitet. Wu, Chen und Offutt [WCO03] präsentieren einen Ansatz, der mittels Interaktionsdiagrammen Interaktionsgraphen erstellt. Diese Graphen stellen den Kontrollfluss zwischen Komponenten dar. Kollaborationsdiagramme dienen zur Ableitung von Abhängigkeiten zwischen den Schnittstellen und Funktionen, verfeinert durch Zustandsdiagramme. Die Autoren unterscheiden hierbei zwischen Kontext- und Inhaltsabhängigkeiten.

Baresel et al. [BCSW03] vergleichen Codeabdeckung mit Modellabdeckung und nennen Maße wie Decision und Condition Coverage. Dies setzt die Verwendung von Simulink-Modellen voraus.

Robinson-Mallett et al. [RHPL08] definieren Abdeckungsmaße für parallele Automaten. Es handelt sich um Sender-, Receiver-, sr-pairs- und sr-paths-Coverage.

Die ISO26262 beschäftigt sich mit der funktionalen Sicherheit (vgl. [ISO11a]). Auch werden Anforderungen an Testabdeckungsmaße für den Integrationstest gelistet. Es handelt sich um die Abdeckung der Schnittstellen und Zustände. Als Testfallermittlungsverfahren wird in Bezug zum Test von Schnittstellen die Äquivalenzklassen- und Grenzwertbildung empfohlen. Dieses Verfahren betrachtet jedoch nur einzelne Ein- und Ausgabewerte, nicht aber Beziehungen bzw. Abhängigkeiten und Wechselwirkungen zwischen E/E-Komponenten.

Abdeckungsmaße für Anwendungsfälle werden eher im Telematikbereich eingesetzt. Systeme, die in dieser Arbeit betrachtet werden sind jedoch meist regelungstechnische Systeme. Abdeckungsmaße auf Basis von Simulink-Modellen behandeln Modelle einzelner Komponenten und somit die interne Struktur der Modelle. Anwendbar sind diese beim MiL-Test bei der virtuellen Integration. Die Maße bezüglich zu sendender und empfangener Nachrichten können bei der Verwendung von K-Matrizen eingesetzt werden.

Eine Auflistung der genannten Testabdeckungsmaße ist in Tabelle 3.1 zu finden.

Quelle	Testabdeckungsmaße
Winter et al. [WES+13]	Zustandsabdeckung, Zustandsübergangsabdeckung und Abdeckung nicht spezifizierter Zustandsübergänge
Nörenberg [Nör12]	Anforderungs-, Zustands-, Schnittstellen- und Anwendungsfallabdeckung
Power und McQuillan [PM05]	Message Sequence Path Coverage, Condition Coverage (Cond), Full Predicate Coverage (FP), Each Message on Link (EML), All Message Paths (AMP), Collection Coverage (Coll), All Transitions Coverage, All Message Sequence Paths Coverage, All Content-Dependence Relationships Coverage, All Transitions Coverage, Control and Data Flow Coverage, All Transitions Coverage, Full Predicate Coverage, Transition-Pair Coverage, Complete Sequence Coverage, All Transitions Coverage, Context-Dependence Relationship Coverage
Winter [Win99]	Use Case Step Coverage, Use Case Branch Coverage, Use Case Scenario Coverage, Use Case Boundary Body Coverage, Use Case Path Coverage
Wu, Pan und Chen [WPC01]	All-Content-Dependences, All-Context-Dependences/Some-Content-Dependences, All-Context-Dependences, All-Events, All-Interfaces
Wu, Chen und Offutt [WCO03]	Each transition in each collaboration diagram, each valid sequence in each collaboration diagram, each transition in each statechart diagram, each content-dependence relationship derived from each collaboration diagram, each effective content-dependence relationship derived from each statechart
Baresel et al. [BCSW03]	Decision Coverage, Condition Coverage
Robinson-Mallett et al. [RHPL08]	Sender-, Receiver-, sr-pairs und sr-paths-Coverage

**Tabelle 3.1.:** Testabdeckungsmaße aus der Literatur

## 3.2. Stand der Technik beim Original Equipment Manufacturer

Abbildung 3.1 zeigt den grundsätzlichen Ablauf von der Analyse bis hin zur Integration von Sub/-Systemen beim OEM. Das System wird analysiert und Anforderungen an das System und dessen Komponenten definiert. Verschiedene Zulieferer werden mit der Entwicklung der Komponenten beauftragt. Vertragsgrundlage sind dabei die Anforderungen, die im SLH bzw. KLH zu finden sind. Die Zulieferer entwickeln und testen die beauftragten Komponenten isoliert und liefern diese in Form von Black-Boxes an den OEM aus. Auch eine Eigenentwicklung von Komponenten kann Bestandteil der Entwicklung sein. Anschließend werden alle zum System gehörigen Komponenten beim OEM integriert und mittels HiL-Prüfstand getestet. Nicht real vorhandene Komponenten werden mit Hilfe von Simulink-Modellen simuliert. Mit Werkzeugen zur Ausführung von Testskripten wird eine automatisierte Ausführung der Testfälle erzielt. Die Testfälle werden aus der Anforderungsspezifikation manuell abgeleitet. Als Testabdeckungsmaß dient üblicherweise die Anforderungsabdeckung (siehe Abschnitt 2.6).

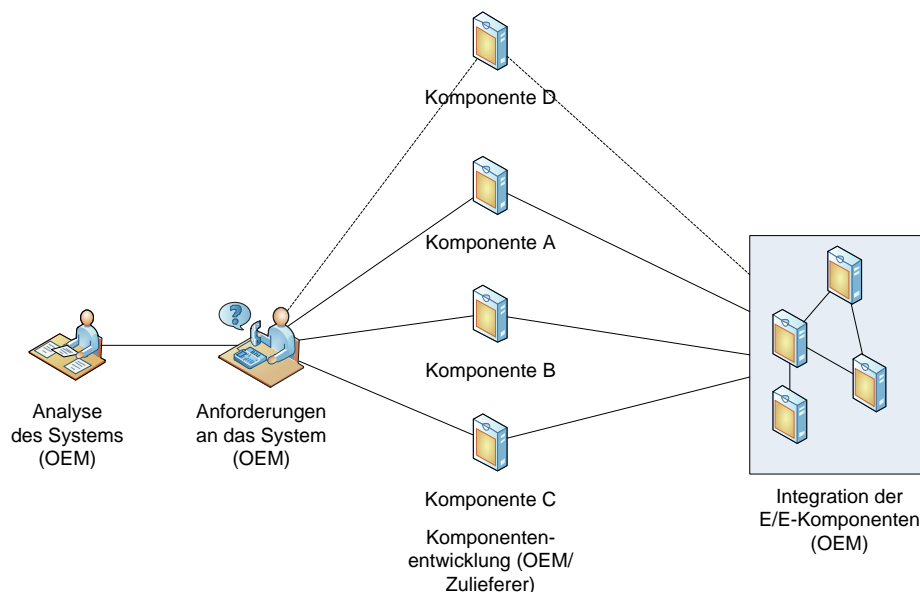


Abbildung 3.1.: Von der Analyse zur Integration

### 3.2.1. Testprozess beim Original Equipment Manufacturer

Viele OEMs testen nach einem Prinzip, das abstrakt betrachtet dem für den Softwaretest von der International Software Testing Qualifications Board (ISTQB) definierten allgemeinen Testprozess ähnelt. Abbildung 3.2 zeigt die verschiedenen Phasen, die während dieses Prozesses

durchlaufen werden (vgl. [Inn]).

In der Planungsphase wird die Teststrategie festgelegt. Ferner wird der zu erreichende Überdeckungsgrad der Testabdeckungsmaße spezifiziert.

Im Zuge der Analyse wird die Testbasis untersucht. Anhand der Ergebnisse der Analyse wird eine geeignete Testumgebung ausgewählt. Anschließend erfolgt die Festlegung der Testziele und Testenkriterien, sowie die Konzeption der Testfälle.

Aus den konzeptionellen Testfällen werden konkrete Testdaten ermittelt und die damit verbundenen Voraussetzungen für die Realisierung bzw. Durchführung der Tests hergestellt. Die Testfälle werden anschließend durchgeführt und die resultierenden Ergebnisse dokumentiert.

Nach der Durchführung der Testfälle werden die Ergebnisse (Ist-Werte) mit den definierten Referenzwerten (Soll-Werten) verglichen.

Für die Archivierung wird eine Übersicht der ausgeführten Testfälle mit den dazugehörigen Testergebnissen erstellt. Durch eine Überwachung (Monitoring) während des gesamten Testprozesses wird die Kontrolle der Testabdeckung ermöglicht.

In Bezug zu den Testenkriterien werden die ausgewerteten Testergebnisse überprüft. Gegebenenfalls werden weitere Testfälle spezifiziert. Zur Gewährleistung der Wiederholbarkeit und Reproduzierbarkeit werden die durchgeführten Testfälle und deren Ergebnisse und Auswertungen gespeichert.

Das Zusammenwirken der E/E-Komponenten wird gemäß des zuvor beschriebenen Testprozesses am HiL-Prüfstand durch Integrationstests getestet. Abbildung 3.3 zeigt den Einsatz des Testprozesses. Anhand der Anforderungsspezifikation (SLH/KLH) wird pro Anforderung mindestens ein Testfall ermittelt. Die generierten Testfälle werden durch spezifische Testdaten auf Basis des Testobjekts ersetzt. Bei diesem handelt es sich immer um ein integriertes Sub-/System. Soll-Werte werden mit Hilfe der Testdaten bestimmt und dienen zur Überprüfung der Korrektheit der Ist-Werte. Das Testobjekt wird ausgeführt und die Ist-Werte mit den zuvor ermittelten Soll-Werten verglichen. Falls Fehler auftreten, müssen diese behoben und es muss erneut getestet werden. Liegen keine zu bearbeitenden Fehler vor und wurden alle Testenkriterien erfüllt, kann der Integrationstest der E/E-Systeme beendet werden.

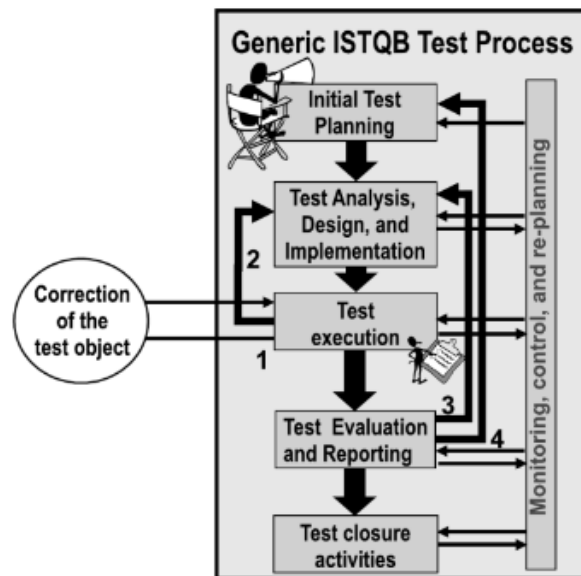


Abbildung 3.2.: Testprozess nach ISTQB [Has14]

### 3. Analyse

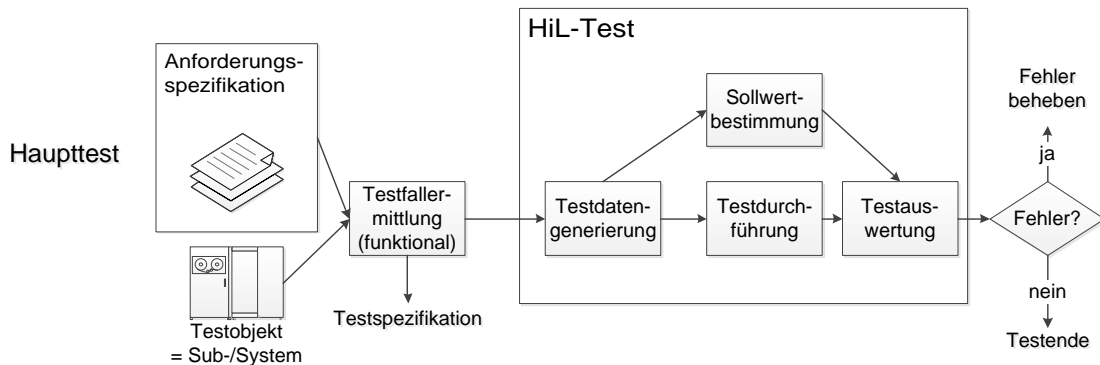


Abbildung 3.3.: Allgemeiner Testprozess

Falls Testbasen vorliegen, die das Zusammenwirken der E/E-Komponenten beschreiben, können diese verwendet werden, um strukturelle Testabdeckungsmaße einzusetzen. Der beschriebene Testprozess kann dann dahingehend erweitert werden, dass zusätzlich zum Haupttest eine ergänzende Auswertung hinsichtlich der strukturellen Testabdeckungsmaße durchgeführt wird (analog zu Abbildung 3.3). Nach einer möglichen Fehlerbehebung auftretender Fehler im Haupttest, können bei Bedarf weitere Testfälle generiert werden, die einen geforderten Überdeckungsgrad bzw. ein zuvor festgelegtes Vollständigkeitsziel erreichen.

#### 3.2.2. Basen für die Ableitung von Testabdeckungsmaßen

Die Analyse der Informationen zu Sub-/Systemen umfasst die Untersuchung von Anforderungsspezifikationen (SLH/KLH), Entwicklungs- und Testdokumenten (z.B. Testkonzept und Testspezifikation). Daraus ergibt sich eine Auswahl an möglichen Basen, die als Grundlage für die Ableitung der Testabdeckungsmaße dienen können. Grundsätzlich existiert beim OEM zu jedem Projekt ein SLH und mehrere KLHs. Die darin beschriebenen Anforderungen sind bereits Basen der Anforderungsabdeckung. Es finden sich darin jedoch auch Zustandsautomaten, die im Falle einer Beschreibung von komponentenübergreifenden Funktionen für die Anwendung von Testabdeckungsmaßen geeignet sind. Für die Integrationstests mittels MiL bieten sich Simulink-Modelle an. Wünschenswert wären hierbei Modelle, die das Zusammenwirken mehrerer Komponenten beschreiben. Beim OEM werden Simulink-Modelle jedoch oft nur für interne Eigenentwicklungen von einzelnen Komponenten genutzt. Diese beschreiben die internen Abläufe einer Komponente. Während der Entwicklung wird Quellcode ausgehend von diesen Simulink-Modellen generiert. Im Bereich der Telematik sind oft Kommunikationsmodelle bzw. Sequenzdiagramme zu finden. Für die Kommunikation und den Test am HiL-Prüfstand können Netzwerkbeschreibungen wie die K-Matrix als Basis für Testabdeckungsmaße dienen. Des Weiteren werden beim OEM Schnittstellenbeschreibungen in Architekturmodellen (z.B. in PREEvision) verwendet.



Relevante Testbasen für den Integrationstest von E/E-Systemen sind daher:

- Anforderungsspezifikationen
- Zustandsautomaten (nur komponentenübergreifend)
- Simulink-Modelle
- Kommunikationsmodelle
- Netzwerkbeschreibungen, insbesondere die K-Matrix
- Architektur- und Schnittstellenbeschreibungen



# 4. Testabdeckungsmaße für den Integrationstest von E/E-Systemen im Fahrzeug

Dieses Kapitel erläutert, wie die in Abschnitt 3.2.1 beschriebene allgemeine Teststrategie durch die im Rahmen dieser Arbeit entwickelten Testabdeckungsmaße für den Integrationstest von E/E-Systemen erweitert werden kann. Ferner ist beschrieben, welchen Zweck die zusätzlichen Testabdeckungsmaße, in Bezug zu den aus den Anforderungen abgeleiteten Testfälle der allgemeinen Teststrategie, erfüllen. Abschließend sind die entwickelten Testabdeckungsmaße beschrieben, die auf Architektur- und Schnittstellenbeschreibungen, Zustandsautomaten und der K-Matrix basieren.

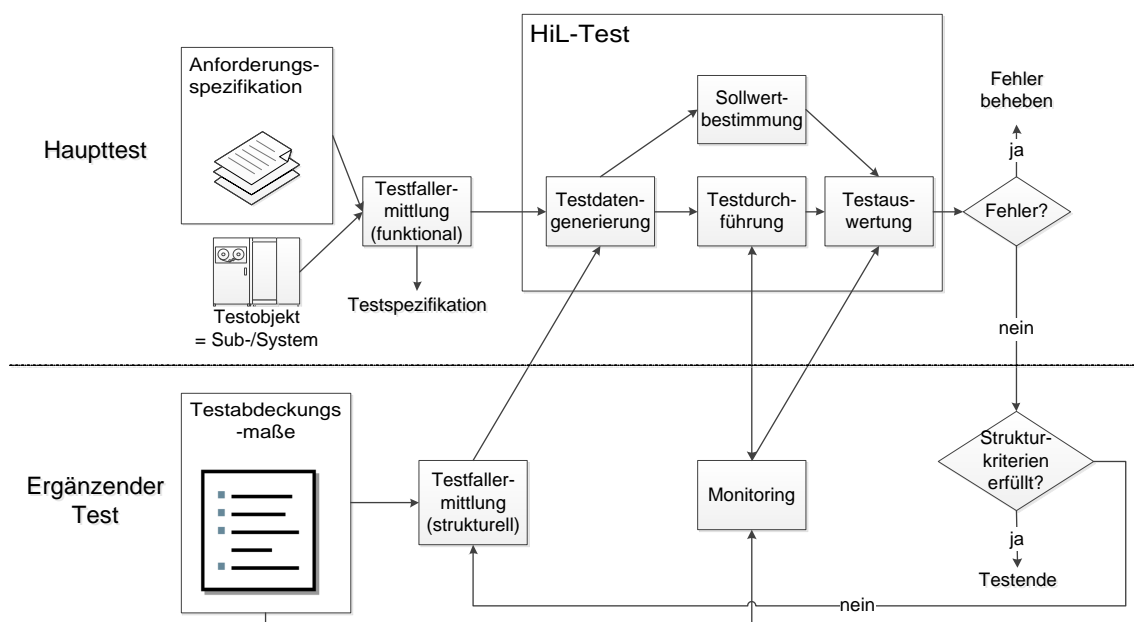
## 4.1. Erweiterte Teststrategie

Abschnitt 3.2.1 beschreibt die bei OEMs typischerweise eingesetzte allgemeine Teststrategie. Als Minimalkriterium bzw. Testendekriterium wird auf die Verifizierung der Anforderungen abgezielt (Anforderungsabdeckung). Folglich liegt der Fokus auf der Abdeckung der Anforderungsspezifikation, die gleichzeitig auch Vertragsgrundlage mit den Zulieferern ist. Auf der Ebene der E/E-Komponenten werden meist keine weiteren Testabdeckungsmaße zur Bewertung der Testgüte herangezogen. Dahingehend ergänzt die in diesem Abschnitt beschriebene erweiterte Teststrategie den ursprünglichen Testvorgang. Die im Rahmen dieser Arbeit entwickelten Testabdeckungsmaße für den Integrationstest von E/E-Systemen werden ergänzend eingesetzt. Dadurch wird die Möglichkeit geschaffen, weitere Testfälle abzuleiten, falls durch die Tests eine unzureichende Testabdeckung im Vergleich zur gewünschten zu erreichenden Testabdeckung erzielt wurde (z.B. auf Basis von Architektur- und Schnittstellenbeschreibungen, Zustandsautomaten und K-Matrizen).

Der Einsatz der erweiterten Teststrategie setzt voraus, dass Beschreibungen des Zusammenwirkens von E/E-Komponenten des integrierten E/E-Systems existieren (siehe dazu Abschnitt 3.2.2). Bezüglich des Integrationstest am HiL-Prüfstand kann davon ausgegangen werden, dass i.d.R. Architektur-/Schnittstellenbeschreibungen und K-Matrizen vorliegen. Während der Analyse im Zusammenhang mit diesem Thema wurde festgestellt, dass komponentenübergreifende Zustandsautomaten nicht in jedem Fall existieren und deshalb z.T. erstellt werden müssen.

#### 4. Testabdeckungsmaße für den Integrationstest von E/E-Systemen im Fahrzeug

Abbildung 4.1 zeigt die bereits in Abschnitt 3.2.1 dargestellte allgemeine Teststrategie und deren Erweiterung durch den Einsatz zusätzlicher, im Rahmen dieser Arbeit entwickelter Testabdeckungsmaße. Dargestellt ist die Erweiterung der allgemeinen Teststrategie um den ergänzenden Test, der zusätzliche Testabdeckungsmaße für den Integrationstest von E/E-Systemen zur Bewertung der Vollständigkeit der Testfälle auswertet. Höchste Priorität hat die Ausführung des funktionalen Haupttests, der die aus den Anforderungen abgeleiteten Testfälle zur Ausführung bringt. Nachdem dieser Test abgeschlossen ist, werden die in dieser Arbeit entwickelten Testabdeckungsmaße zur Bewertung der Güte der Testfälle herangezogen. Die erweiterte Teststrategie darf nicht als Ersatz für den eigentlichen funktionalen Haupttest angesehen werden.



**Abbildung 4.1.:** Erweiterung der allgemeinen Teststrategie

Anhand der vorliegenden Basis, die das Zusammenwirken der E/E-Komponenten beschreibt, wird zuvor definiert welcher Prozentsatz der real erreichten Testabdeckung genügt, um das Beenden der Tests einzuleiten. Sind diese während des Tests erfüllt worden, kann der Test beendet werden. Ansonsten werden anhand der entwickelten Testabdeckungsmaße weitere Testfälle ermittelt, die die Vollständigkeit der Testabdeckung erhöhen sollen. Diese fließen dann in die Durchführung des HiL-Tests ein. Erneut werden Testdaten generiert, der hinzugefügte Test durchgeführt und anschließend Soll- und Ist-Werte zur Testauswertung verglichen. Für die Praxis ist eine werkzeugunterstützte Überwachung bzw. Messung (Monitoring) der Testabdeckung notwendig. Deren Konzeption und Implementierung ist jedoch nicht Bestandteil dieser Arbeit. Etwaige beim Test auftretende Fehler müssen anschließend behoben werden und deren Abwesenheit durch eine erneute Testdurchführung geprüft werden. Liegen keine Fehler

## 4.2. Testabdeckungsmaße basierend auf Architektur- und Schnittstellenbeschreibungen

mehr vor, erfolgt die erneute Überprüfung der erreichten Testabdeckung. Je nach Resultat wiederholt sich dieser Vorgang, bis das gewünschte Ergebnis erzielt wird.

Wichtig ist, dass der abdeckungsorientierte ergänzende Test auf Basis der im Rahmen dieser Arbeit entwickelten Testabdeckungsmaße für den Integrationstest von E/E-Systemen nicht als Testziel dient. Es handelt sich eher um ein formales zusätzliches Kriterium und somit um eine Ergänzung des funktionsorientierten Tests bezüglich Vollständigkeit bzw. Testendekriterien. Der ergänzende Test darf kein Ersatz für den Haupttest auf Basis der Anforderungen sein. Strukturelle Testabdeckungskriterien sind als zusätzliche Testvollständigkeitsmaße zu sehen.

## **4.2. Testabdeckungsmaße basierend auf Architektur- und Schnittstellenbeschreibungen**

### **Voraussetzung:**

Der Verlauf der Nachrichten und Signale muss in den Beschreibungen zu erkennen sein. In diesem Zusammenhang muss es möglich sein, deren Inhalt bzw. Parameter zu überwachen. Die Analyse der Kommunikation kann beispielsweise über den Nachrichten-Trace<sup>1</sup> erfolgen. Weiterhin ist es notwendig, dass die physikalischen Schnittstellen messbar sind.

### **Hinweise:**

Die Erläuterung von Architektur- und Schnittstellenbeschreibungen ist in Abschnitt 2.5.3 zu finden. Der beispielhafte Einsatz der Testabdeckungsmaße ist in Abschnitt 5.1 beschrieben. Eine Auflistung der Testabdeckungsmaße inklusive Beschreibung ist in Tabelle 4.1 dargestellt.

### **Testabdeckungsmaße:**

Zur Abdeckung der Architekturmodelle sind die darin beschriebenen Komponenten und deren Kommunikation untereinander im Fokus. Zur Überprüfung der Funktionsfähigkeit der Komponenten, muss jede Komponente während des Tests mindestens einmal ausgeführt worden sein. Das zugehörige Testabdeckungsmaß wird fortan als Komponentenabdeckung (Component Coverage) bezeichnet. An dieser Stelle ist zu bemerken, dass die korrekte Funktionsweise der Komponenten bereits durch den Zulieferer sichergestellt werden muss. Es soll beim Integrationstest der E/E-Systeme nicht die E/E-Komponente an sich überprüft werden. Durch die Ausführung jeder Komponente im Verbund, können jedoch Rückschlüsse gezogen werden, ob diese korrekt entwickelt sowie integriert wurden und ob das spezifizierte Zusammenwirken wie gefordert gegeben ist.

Zur weiteren Vertiefung des Kommunikationsverhaltens müssen verbundene Komponentenpaare mindestens einmal miteinander kommuniziert haben, wodurch ebenfalls eine Komponentenabdeckung erreicht wird. Als Testabdeckungsmaß ist in diesem Fall die Kompo-

---

<sup>1</sup>Nachrichten-Trace: zeitlich geordnete Auflistung aller Nachrichten auf Kommunikationsbussen wie z.B. CAN-Bus

#### 4. Testabdeckungsmaße für den Integrationstest von E/E-Systemen im Fahrzeug

nentenpaarabdeckung (Component-Pair Coverage) zu nutzen. Abbildung 4.2 zeigt den stark vereinfachten Aufbau eines Komponentennetzwerks. In den Teilbildern a)-d) ist die beispielhafte Abdeckung der Komponentenpaare (rot) und implizit die vollständige Abdeckung der beteiligten Komponenten dargestellt. Jede E/E-Komponente verfügt mindestens über je eine Schnittstelle am Eingang und Ausgang. Kommunizierende Komponenten enthalten Schnittstellen mit gemeinsamen Eigenschaften und Protokollen, die zum Datenaustausch genutzt werden.

Die Schnittstellenabdeckung (Interface Coverage) erfordert, dass jede Schnittstelle einer Komponente während des Tests mindestens einmal angesprochen bzw. aktiviert worden ist. Schnittstellen können auf verschiedenen Wegen aktiviert werden. Diese verschiedenen Möglichkeiten werden Ereignisse genannt. Die Abdeckung dieser Ereignisse wird über die Ereignisabdeckung (Event Coverage) gemessen, d.h. jedes Ereignis, das zur Aktivierung einer Schnittstelle einer Komponente führt, muss während des Tests mindestens einmal aufgetreten sein. Hierzu ein Beispiel: die Schnittstelle zur Außenlichtansteuerung kann durch Ereignisse wie Blinkeranforderung, Fernlichtanforderung, oder Standlichtanforderung angesprochen werden. Jedes dieser Ereignisse aktiviert die Schnittstelle durch einen alternativen Auslöser. Abbildung 4.3 zeigt die E/E-Komponenten ECU 1, ECU 2 und einen Aktor, der je nach Kontrollsignal der ECUs reagiert. Die abzudeckenden Schnittstellen sind rot markiert. Die obere Schnittstelle am Eingang des Aktors, kann entweder durch ECU 1 oder durch ECU 2 aktiviert werden (entspricht den Ereignissen).

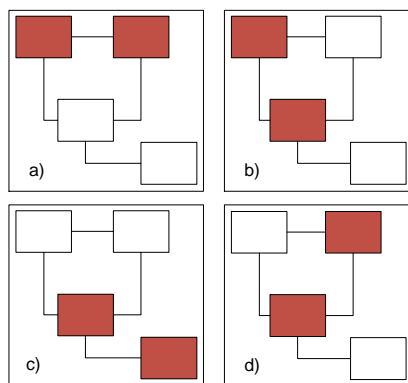


Abbildung 4.2.: Komponentenpaare

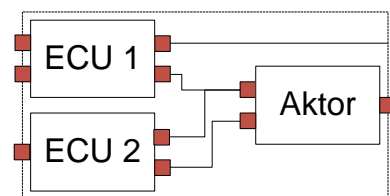


Abbildung 4.3.: Schnittstellen

Weiterhin dienen die Signale, die zwischen den Komponenten ausgetauscht werden als Messobjekt zur Messung der Signalabdeckung (Signal Coverage). Bei der Signalabdeckung muss jedes Signal mindestens einmal während des Tests gesendet worden sein. Signale enthalten Parameter, die mit unterschiedlichen Werten/Wertebereichen belegt werden können. Die Signalparameterabdeckung (All-Signal-Parameter Coverage) erfordert, dass jeder Parameter eines Signals jeden spezifizierten Wert/Wertebereich mindestens einmal während des Tests angenommen hat. Auf der physikalischen Ebene ist die Pin/Port-Abdeckung (Pin/Port Coverage) einzusetzen, d.h. jeder physikalische Pin/Port muss mindestens einmal während des Tests angesprochen werden. Zu beachten ist, dass beim Test am HiL-Prüfstand auch zusätzliche Pins

bzw. Ports zur Diagnose existieren können. Diese werden nicht zur Messung der Abdeckung herangezogen.

Tabelle 4.1 liefert einen Überblick über die entwickelten Testabdeckungsmaße. Die Spalte „Testabdeckungsmaße“ beinhaltet die Bezeichnung der jeweiligen Testabdeckungsmaße, die Spalte „Kurzbeschreibung“ erläutert, welche Elemente abzudecken sind.

Testabdeckungsmaße	Kurzbeschreibung
Komponentenabdeckung (Component Coverage)	Jede E/E-Komponente muss mindestens einmal während des Tests ausgeführt worden sein.
Komponentenpaarabdeckung (Component Pair Coverage)	Jedes Paar von E/E-Komponenten, das eine Kommunikationsbeziehung pflegt, muss mindestens einmal während des Tests ausgeführt worden sein.
Schnittstellenabdeckung (Interface Coverage)	Jede Schnittstelle einer Komponente muss mindestens einmal während des Tests aktiviert worden sein.
Ereignisabdeckung (Event Coverage)	Jedes Ereignis, das zur Aktivierung einer Schnittstelle einer Komponente führt, muss mindestens einmal während des Tests aufgetreten sein.
Signalabdeckung (Signal Coverage)	Jedes Signal, das von E/E-Komponenten gesendet/empfangen werden kann, muss mindestens einmal während des Tests gesendet/empfangen werden.
Signalparameterabdeckung (All-Signal-Parameter Coverage)	Jeder Parameter eines Signals muss mindestens einmal während des Tests dessen spezifizierte Werte/Wertebereiche angenommen haben.
Pin-/Port-Abdeckung (Pin-/Port Coverage)	Jeder physikalische Port/Pin muss mindestens einmal während des Tests angesprochen werden.

**Tabelle 4.1.:** Testabdeckungsmaße basierend auf Architektur- und Schnittstellenbeschreibungen

## 4.3. Testabdeckungsmaße basierend auf Zustandsautomaten

### Voraussetzung:

Der Zustandsautomat, der als Basis dient, darf sich nicht allein auf die internen Vorgänge der einzelnen Komponenten beziehen. Für das hier betrachtete Thema Integrationstest ist es notwendig, dass dieser komponentenübergreifende Funktionen und Zustände modelliert. D.h. es sind Zustände und Zustandsübergänge beschrieben, die das Zusammenwirken mehrerer E/E-Komponenten implizieren. Ferner muss es eine Möglichkeit geben, die Zustände am HiL-Prüfstand von „außen“ zu identifizieren.

## 4. Testabdeckungsmaße für den Integrationstest von E/E-Systemen im Fahrzeug

### Hinweise

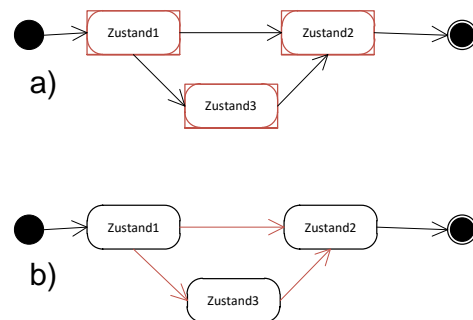
Die Erläuterung von Zustandsautomaten ist in Abschnitt 2.5.1 zu finden. Der beispielhafte Einsatz der Testabdeckungsmaße ist in Abschnitt 5.1 beschrieben. Eine Auflistung der Testabdeckungsmaße inklusive Beschreibung ist in Tabelle 4.2 dargestellt.

### Testabdeckungsmaße:

Zum einen sollen die Zustände abgedeckt werden und somit überprüft werden, ob auch jede damit verbundene Funktionalität bereitgestellt werden kann. Das bedeutet, jeder Zustand, an dessen Realisierung mehrere Komponenten beteiligt sind, muss während des Tests mindestens einmal erreicht werden. Zugehöriges Testabdeckungsmaß ist die Zustandsabdeckung (All-States Coverage). Ebenso muss jeder Zustandsübergang, der durch das Zusammenwirken mehrerer E/E-Komponenten ausgelöst wird, durchlaufen worden sein. Zur Messung dient die Zustandsübergangsabdeckung (All-Transition Coverage). Abbildung 4.4 a) zeigt rot markiert eine vollständige Abdeckung der drei vorhandenen Zustände. Dazu muss der Test so ausgerichtet werden, dass ein bestimmter Pfad im Zustandsautomat durchlaufen wird. D.h. die Testfälle müssen so gestaltet werden, dass sich der Zustandsautomat mindestens einmal in jedem definierten Zustand befindet.

Teil b) der Abbildung zeigt eine vollständige Abdeckung der Zustandsübergänge, ebenfalls in rot markiert. Die Testfälle müssen so gestaltet werden, dass das Durchlaufen des Zustandsautomaten jeden Zustandsübergang mindestens einmal ausführt. Durch eine vollständige Abdeckung der Zustandsübergänge sind automatisch schon alle Zustände abgedeckt.

Weiterhin soll eine festgelegte Pfadabdeckung (All-Paths Coverage) erreicht werden. Dies ist in der Praxis i.d.R. nicht anwendbar, denn es können Schleifen existieren, die zu einer nicht endlichen Pfadanzahl führen. Möglich ist aber die Festlegung einer relevanten Anzahl von Schleifendurchläufen. Durch eine Pfadabdeckung sind ebenfalls teilweise Zustände und Zustandsübergänge abgedeckt. Zu bemerken ist, dass eine vollständige Abdeckung nicht immer erreicht werden kann. Es kann vorkommen, dass sich verschiedene Kombinationen von Zuständen logisch ausschließen. Zusätzlich zur Pfadabdeckung lassen sich Sequenzen ausführen, die einen praktisch relevanten Anwendungsfall darstellen. Das dazugehörige Testabdeckungsmaß wird als Sequenzabdeckung (Complete-Sequence Coverage) bezeichnet. Auch dies ist in der Praxis i.d.R. nicht anwendbar, auf Grund von einer Vielzahl möglicher Sequenzen. Abhilfe kann durch die Definition einer Untermenge sinnvoller Sequenzen geschaffen werden.



**Abbildung 4.4.:** Abdeckung der Zustände und Zustandsübergänge in einem Zustandsautomaten



#### 4.4. Testabdeckungsmaße basierend auf Kommunikationsmatrizen

Tabelle 4.2 liefert einen Überblick über die entwickelten Testabdeckungsmaße. Die Spalte „Testabdeckungsmaße“ beinhaltet die Bezeichnung der jeweiligen Testabdeckungsmaße, die Spalte „Kurzbeschreibung“ erläutert, welche Elemente abzudecken sind.

<b>Testabdeckungsmaße</b>	<b>Kurzbeschreibung</b>
Zustandsabdeckung (All-States Coverage)	Jeder Zustand, der durch mehrere E/E-Komponenten realisiert wird, muss während des Tests mindestens einmal erreicht worden sein.
Zustandsübergangsabdeckung (All-Transition Coverage)	Jeder Zustandsübergang, der durch das Zusammenwirken mehrerer E/E-Komponenten ausgelöst wird, muss mindestens einmal während des Tests durchlaufen werden.
Pfadabdeckung (All-Paths Coverage)	Jeder Pfad muss während des Tests mindestens einmal durchlaufen worden sein.
Sequenzabdeckung (Complete-Sequence Coverage)	Jede Sequenz, die einen praktisch relevanten Anwendungsfall durchläuft, muss während des Tests mindestens einmal durchlaufen worden sein.

**Tabelle 4.2.:** Testabdeckungsmaße basierend auf Zustandsautomaten

## 4.4. Testabdeckungsmaße basierend auf Kommunikationsmatrizen

### Voraussetzung:

Anhand der K-Matrix müssen alle relevanten Nachrichten und Signale betrachtet werden, die zwischen den Komponenten des zu testenden Sub-/Systems ausgetauscht werden. Es muss sich immer um die Kommunikation zwischen Komponenten handeln. Des Weiteren muss es eine Möglichkeit geben, die Nachrichten und Signale in einem Nachrichten-Trace zu analysieren.

### Hinweise

Die Erläuterung einer K-Matrix ist in Abschnitt 2.5.2 zu finden. Der beispielhafte Einsatz der Testabdeckungsmaße ist in Abschnitt 5.1 beschrieben. Eine Auflistung der Testabdeckungsmaße inklusive Beschreibung ist in Tabelle 4.3 dargestellt.

### Testabdeckungsmaße

Für jede Komponente werden alle zu sendenden und empfangenen Nachrichten mindestens einmal abgedeckt, d.h. versendet und empfangen. Das bedeutet, dass die Testfälle so gestaltet werden, dass jede Nachricht, die für ein Sub-/System relevant ist, mindestens einmal von der spezifizierten Komponente versendet/empfangen wird. Die verwendeten Testabdeckungsmaße werden als Senderabdeckung (Sender Coverage) und Empfängerabdeckung (Receiver Coverage) bezeichnet. Alle Signale müssen mindestens einmal gesendet werden. Analog zur

#### 4. Testabdeckungsmaße für den Integrationstest von E/E-Systemen im Fahrzeug

---

Definition der Testabdeckungsmaße in Abschnitt 4.2 wird von der Signalabdeckung (Signal Coverage) gesprochen. Diesbezüglich müssen die Testfälle so gestaltet werden, dass jedes Signal, das für ein Sub-/System relevant ist, mindestens einmal versendet wird. Dies erfüllt ebenfalls die Senderabdeckung. In Bezug zu zyklischen Nachrichten sind die bisher genannten Testabdeckungsmaße nicht aussagekräftig. Gut geeignet sind diese jedoch bei sporadisch auftretenden Nachrichten. Des Weiteren werden alle in den Nachrichten enthaltenen Signale mindestens einmal mit allen Möglichkeiten der Parametrisierung abgedeckt. Zur Messung dient die Signalparameterabdeckung (All-Signal-Parameter Coverage). Die Testfälle müssen hierzu so gestaltet werden, dass jeder Parameter eines Signals, das für ein Sub-/System relevant ist, mindestens einmal jeden möglichen Wert angenommen hat. Dieses Testabdeckungsmaß eignet sich auch zur Messung von zyklischen Nachrichten.

Tabelle 4.3 liefert einen Überblick über die entwickelten Testabdeckungsmaße. Die Spalte „Testabdeckungsmaße“ beinhaltet die Bezeichnung der jeweiligen Testabdeckungsmaße, die Spalte „Kurzbeschreibung“ erläutert, welche Elemente abzudecken sind.

---

<b>Testabdeckungsmaße</b>	<b>Kurzbeschreibung</b>
Senderabdeckung (Sender Coverage)	Jede von einer Komponente zu sendende Nachricht muss während des Tests mindestens einmal auf dem verwendeten Bus gesendet worden sein.
Empfängerabdeckung (Receiver Coverage)	Jede von einer Komponente zu empfangene Nachricht muss während des Tests mindestens einmal empfangen (bzw. vom Bus gelesen) worden sein.
Signalabdeckung (Signal Coverage)	Jedes Signal, das von E/E-Komponenten gesendet/empfangen werden kann, muss mindestens einmal während des Tests gesendet/empfangen werden.
Signalparameterabdeckung (All-Signal-Parameter Coverage)	Jeder Parameter eines Signals muss mindestens einmal während des Tests dessen spezifizierte Werte/Wertebereiche angenommen haben.

---

**Tabelle 4.3.:** Testabdeckungsmaße basierend auf K-Matrizen

### 4.5. Abschließende Bemerkungen

Grundlage für die Definition der Maße sind die Ergebnisse der Literaturrecherche und Diskussionsrunden mit Experten im Bereich Testen. Der Integrationstest am HiL-Prüfstand setzt Systemanforderungen (SLH), Schnittstellenbeschreibungen und K-Matrizen als Testbasen voraus. Bei komplexen Systemen werden i.d.R. auch Zustandsautomaten erstellt, um das Systemverhalten zu beschreiben. Daher sind die zuvor beschriebenen Kategorien von Testabdeckungsmaßen für den Integrationstest von E/E-Systemen relevant.

Bei allen Testabdeckungsmaßen, die in Verbindung zu Eingabewerten bzw. Wertebereichen stehen, kann weiter verfeinert werden. Hierzu sei auf die Verwendung der Äquivalenzklassenbildung und Grenzwertbildung verwiesen.

Die in dieser Arbeit definierten Testabdeckungsmaße sind kompatibel zu den Abdeckungsmaßen der ISO26262. Sie stellen eine Verfeinerung dar und können somit auch für den Integrationstest von funktional abzusichernden Systemen nützlich sein.



# 5. Evaluierung

Dieses Kapitel erläutert den Einsatz ausgewählter Testabdeckungsmaße anhand von vereinfachten Fallbeispielen aus dem realen Umfeld der Automobilindustrie. Zur Bewertung der entwickelten Testabdeckungsmaße für den Integrationstest von E/E-Systemen sind die Ergebnisse der Befragungen von Experten zusammengefasst, die mit Hilfe von Fragebögen durchgeführt wurden.

## 5.1. Fallbeispiele: Anwendung der Testabdeckungsmaße

### Beispiel 5.1 (Architektur- und Schnittstellenbeschreibungen)

*Testobjekt dieses Fallbeispiels ist mit PREEvision dargestelltes Architekturmodell (siehe Abbildung 5.1).*

#### **Berechnung der Komponentenabdeckung:**

Die Hardware-Netzwerk-Architektur umfasst einen Sensor, drei ECUs und einen Aktor. Daher dienen fünf E/E-Komponenten als Testbasis für den Integrationstest.

Die prozentuale Abdeckung der Komponenten wird wie folgt berechnet:

$$\text{Überdeckungsgrad}_{\text{component coverage}} = \frac{\# \text{ ausgeführte E/E-Komponenten}}{5} \quad (5.1)$$

Es sei angenommen, dass kein Testfall existiert, der ECU 1 aktiviert. Dies entspricht einer Abdeckung von 80 Prozent.

#### **Berechnung der Komponentenpaarabdeckung:**

Es existieren die folgenden fünf Komponentenpaare:

1. Sensor und ECU 1
2. Sensor und ECU 2
3. ECU 1 und ECU 3
4. ECU 2 und ECU 3
5. ECU 3 und Aktor

## 5. Evaluierung

---

Die prozentuale Abdeckung der Komponentenpaare wird wie folgt berechnet:

$$\text{Überdeckungsgrad}_{\text{component pair coverage}} = \frac{\# \text{ kommunizierende E/E-Komponentenpaare}}{5} \quad (5.2)$$

Es sei angenommen, dass kein Testfall existiert, der eine Kommunikation zwischen dem Sensor und der ECU 1 hervorruft. Dies entspricht einer Abdeckung von 80 Prozent.

### **Berechnung der Schnittstellenabdeckung:**

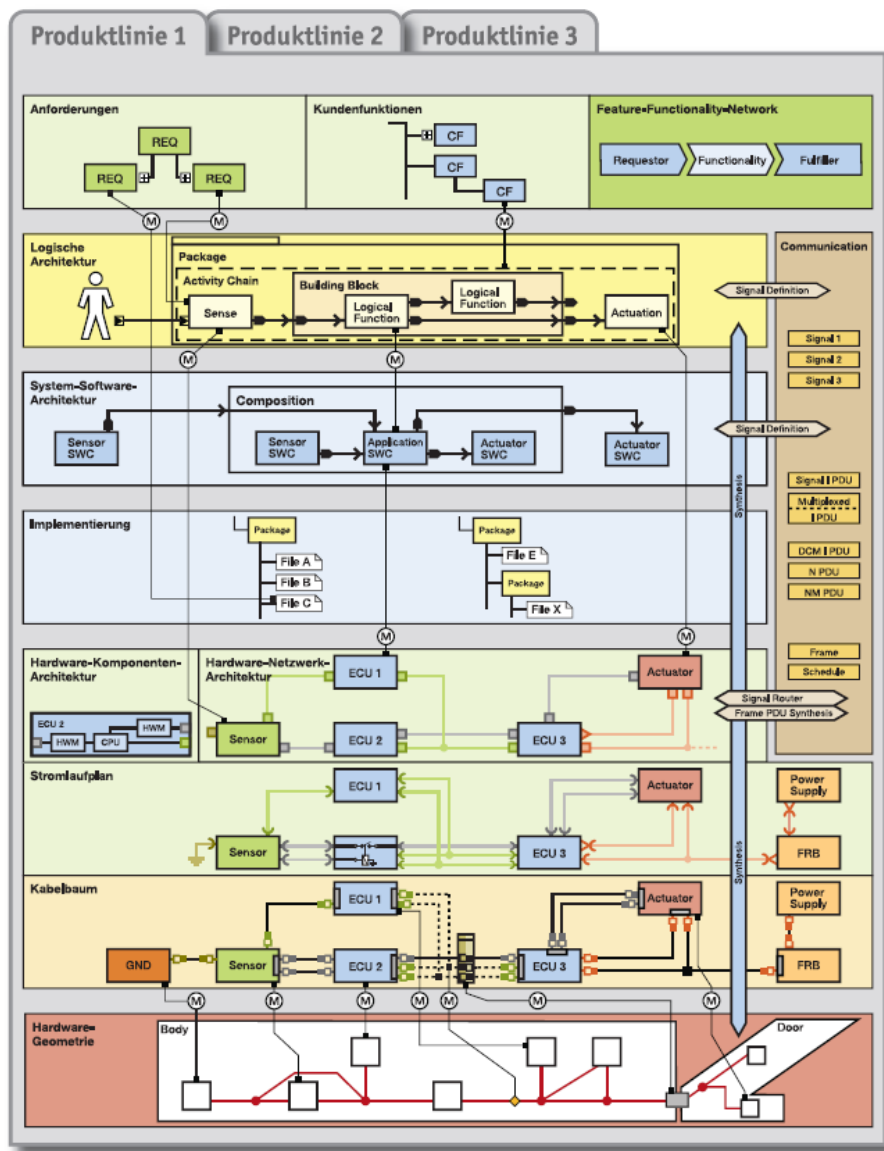
Der Sensor und die ECU 2 besitzen eine Eingangs- und zwei Ausgangsschnittstellen. Die ECU 1 ist mit je einer Eingangs- und Ausgangsschnittstelle versehen. Die ECU 3 verfügt über zwei Eingangs- und drei Ausgangsschnittstellen. Der Aktor besitzt drei Eingangsschnittstellen. In Summe müssen während des Tests 16 Schnittstellen aktiviert werden, um eine vollständige Abdeckung zu erreichen.

Die prozentuale Abdeckung der Schnittstellen wird wie folgt berechnet:

$$\text{Überdeckungsgrad}_{\text{interface coverage}} = \frac{\# \text{ aktivierte Schnittstellen}}{16} \quad (5.3)$$

Es sei angenommen, dass die zuvor definierten Annahmen gelten. Bei 13 aktivierten Schnittstellen ergibt sich eine Abdeckung von 81 Prozent.

Signale und deren Parameter sind in diesem Beispiel nicht erkenntlich. Dazu muss das Modell in PREvision analysiert werden. Die Berechnung deren Abdeckung erfolgt analog zu den vorangehenden Berechnungen.



**Abbildung 5.1.:** Architekturmodell zum Fallbeispiel „Einsatz der Testabdeckungsmaße auf Basis von Architektur- und Schnittstellenbeschreibungen“ [Vecka]

**Beispiel 5.2 (Einsatz der Testabdeckungskriterien auf Basis von Zustandsautomaten)**  
 Testobjekt dieses Fallbeispiels ist das System „Außenlicht“, dessen Funktionsweise durch einen komponentenübergreifenden Zustandsautomaten beschrieben ist. Zur Betrachtung dient der Zustandsübergang vom Zustand „Standlicht“ in den Zustand „Fahrtlicht“ (siehe Abbildung 5.2).

Die folgende Auflistung zeigt einen Auszug aus dem SLH zur Spezifikation des Außenlichts von Fahrzeugen. Dargestellt sind die Anforderungen für die Zustandsübergänge der Funktion „Fahrtlicht“:

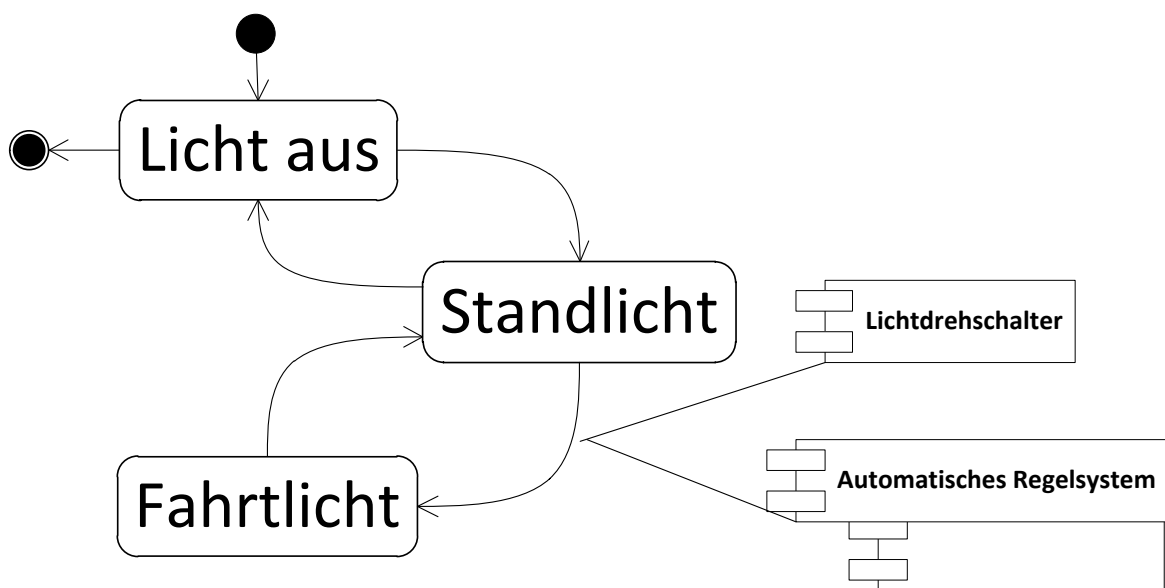
## 5. Evaluierung

Der Zustandsübergang vom Zustand „Standlicht“ in den Zustand „Fahrtlicht“ kann durch zwei verschiedene Möglichkeiten erfolgen:

**Anforderung 1** Stellung des Lichtdrehhalters ist nicht „Manuelles Fahrlicht“.

**Anforderung 2** Zündschlüssel in Stellung 1 und Motor ausgeschaltet und Standlicht an.

Der zugehörige Zustandsautomat ist in Abbildung 5.2 vereinfacht dargestellt. Es existieren drei Zustände „Licht aus“, „Standlicht“, „Fahrtlicht“ sowie vier Zustandsübergänge. Die Zustandsübergänge werden durch das Zusammenwirken mehrerer Komponenten (z.B. Lichtdrehhalter und Automatisches Regelsystem) realisiert.



**Abbildung 5.2.:** Zustandsautomat zum Fallbeispiel „Einsatz der Testabdeckungskriterien auf Basis von Zustandsautomaten“

### Berechnung der Zustandsabdeckung:

Die prozentuale Abdeckung der Zustände wird wie folgt berechnet:

$$\text{Überdeckungsgrad}_{state\ coverage} = \frac{\# \text{ angenommene Zustände}}{3} \quad (5.4)$$

Es sei angenommen, dass kein Testfall existiert, der den Zustand „Fahrtlicht“ einnimmt. Es ergibt sich eine Abdeckung von 67 Prozent.

### Berechnung der Zustandsabdeckung:

Die prozentuale Abdeckung der Zustandsübergänge wird wie folgt berechnet:

$$\text{Überdeckungsgrad}_{transition\ coverage} = \frac{\# \text{ durchlaufene Zustandsübergänge}}{4} \quad (5.5)$$



Es sei angenommen, dass kein Testfall existiert, der den Zustandsübergang zwischen den Zuständen „Standlicht“ und „Fahrtlicht“ aktiviert. Es ergibt sich eine Abdeckung von 75 Prozent.

### **Berechnung der Pfadabdeckung:**

Die prozentuale Abdeckung der Pfade wird wie folgt berechnet:

$$\text{Überdeckungsgrad}_{\text{path coverage}} = \frac{\# \text{ durchlaufene Pfade}}{6} \quad (5.6)$$

Es sei angenommen, dass die Anzahl der zu testenden Pfade auf sechs mögliche Pfade beschränkt ist. Weiterhin sei angenommen, dass während des Tests nur der Pfad „Licht aus“ → „Standlicht“ → „Fahrtlicht“ → „Licht aus“ durchlaufen wurde. Es ergibt sich eine Abdeckung von 17 Prozent.

### **Berechnung der Sequenzabdeckung:**

Die prozentuale Abdeckung der Szenarien wird wie folgt berechnet:

$$\text{Überdeckungsgrad}_{\text{scenario coverage}} = \frac{\# \text{ durchlaufene Szenarien}}{2} \quad (5.7)$$

Es sei angenommen, dass die folgenden zwei möglichen Szenarien existieren:

1. Einschalten des Lichts nach Stillstand des Fahrzeugs.
2. Ausschalten des Lichts während der Fortbewegung des Fahrzeugs.

Zudem sei angenommen, dass nur das erste Szenario während des Tests ausgeführt wurde. Es ergibt sich eine Abdeckung von 50 Prozent.

### **Beispiel 5.3 (Einsatz der Testabdeckungsmaße auf Basis von K-Matrizen)**

*Testobjekt dieses Fallbeispiels ist das System „Außenlicht“, dessen Nachrichten, Signale und beteiligte Komponenten in einer K-Matrix gelistet sind.*

Abbildung 5.3 zeigt einen Auszug einer K-Matrix, welche die Kommunikationsmöglichkeiten eines Blinkersystems beschreibt.

## 5. Evaluierung

	Bedeutung	I- C A N	F- C A N	F- FL EX	LI N- Bu s	Sender	Lae nge	Mi n- We rt	Ma x- We rt	Enumeration/Codierung
P_FRAHR_b	Ansteuerung Blinkerleuchte vorne rechts	X					1 Bit	0	1	0 - Leuchte aus 1 - Leuchte an
P_FRAVL_b	Ansteuerung Blinkerleuchte hinten links	X					1 Bit	0	1	0 - Leuchte aus 1 - Leuchte an
P_FRAVR_b	Ansteuerung Blinkerleuchte hinten rechts	X					1 Bit	0	1	0 - Leuchte aus 1 - Leuchte an
P_Abbiegelichf. inks_pc	Status Abbiegelicht (Abbiegen nach Links)	X					7 Bit	0	127	0 .. 100 - Ansteuerung in Prozent 101 .. 127 Lampe aus

**Abbildung 5.3.:** K-Matrix zum Fallbeispiel „Einsatz der Testabdeckungsmaße auf Basis von K-Matrizen“ [Fra15]

### Berechnung der Senderabdeckung:

Die prozentuale Abdeckung der zu sendenden Nachrichten wird wie folgt berechnet:

$$\text{Überdeckungsgrad}_{\text{sender coverage}} = \frac{\# \text{ gesendete Nachrichten}}{4} \quad (5.8)$$

Es sei angenommen, dass kein Testfall in der Testspezifikation definiert ist, der die Ansteuerung der Blinkerleuchte hinten links aktiviert (diese Annahme gilt auch für die folgenden Berechnungen). Es ergibt sich eine Abdeckung von 75 Prozent.

### Berechnung der Empfängerabdeckung:

Die prozentuale Abdeckung der zu empfangenen Nachrichten wird wie folgt berechnet:

$$\text{Überdeckungsgrad}_{\text{receiver coverage}} = \frac{\# \text{ empfangene Nachrichten}}{4} \quad (5.9)$$

Es ergibt sich eine Abdeckung von 75 Prozent.

### Berechnung der Signalabdeckung:

Die prozentuale Abdeckung der Signale wird wie folgt berechnet:

$$\text{Überdeckungsgrad}_{\text{signal coverage}} = \frac{\# \text{ gesendete Signale}}{4} \quad (5.10)$$

Es ergibt sich eine Abdeckung von 75 Prozent.

### Berechnung der Signalparameterabdeckung:

Die prozentuale Abdeckung der Signalparameter wird wie folgt berechnet:

$$\text{Überdeckungsgrad}_{\text{parameter coverage}} = \frac{\# \text{ geänderte Parameter}}{4} \quad (5.11)$$

Es sei angenommen, dass die Blinkerleuchte vorne rechts während des Tests nie eingeschaltet wird. Es ergibt sich beispielsweise für das Signal „P\_FRAHR\_b“ eine Parameterabdeckung von 50 Prozent.

An dieser Stelle sei darauf hingewiesen, dass das Signal „P\_AbbiegelichtLinks\_pc“ als Parameter einen Wertebereich enthält. Hier kann eine Abdeckung der Äquivalenzklassen und Grenzwerte sinnvoll sein.

## 5.2. Bewertung durch Experten

Im Anhang liegt ein Fragebogen vor, der genutzt wurde, um verschiedene Experten zu den entwickelten Testabdeckungsmaßen für den Integrationstest von E/E-Systemen zu befragen. Der Fragebogen diente als Leitfaden für die Diskussionsrunden. Es wurden drei Gruppen befragt:

1. Interne Test-Experten, die im Bereich des Integrationstest am HiL tätig sind
2. Interne Experten im Bereich Testmethoden
3. Externe Test-Experten aus unternehmensrelevanten Testhäusern

Die Auswertung der Einschätzung durch Experten hat ergeben, dass die Anwendung der entwickelten Testabdeckungsmaße durchaus sinnvoll sein kann. Allein die Anforderungsabdeckung lässt oftmals keine Aussage über die strukturelle Abdeckung der Systeme zu. Interessant wäre auch eine Betrachtung von systemübergreifenden Beschreibungen. Zur automatisierten Messung muss es jedoch eine werkzeugunterstützte Lösung geben. Da die Testabdeckungsmaße eine Verfeinerung der Testabdeckungsmaße in der ISO26262 darstellen, ist ein Einsatz beim Integrationstest von sicherheitsrelevanten E/E-Systemen im Fahrzeugbereich sinnvoll.

In der Praxis können Testabdeckungsmaße basierend auf K-Matrizen dahingehend eingesetzt werden, dass der Nachrichtenverlauf auf dem Bus ausgewertet wird. Dazu muss es eine Möglichkeit geben relevante Nachrichten und die darin enthaltenen Signale zu filtern. An dieser Stelle muss ein Werkzeug eingesetzt werden, das diese Arbeit übernimmt (z.B. Skriptgesteuerte Auswertung von Nachrichten-Traces in XML-Form). Die Aktivierung bestimmter Signale kann jedoch komplexe Nachrichtenverläufe voraussetzen. Durch die Messung der zu sendenden Nachrichten können nicht zum Test genutzte sporadische Nachrichten aufgedeckt werden. Bei zyklischen Nachrichten bietet sich die Messung der Signale (und Parameter) an. Nach Einschätzung der Experten werden oft nur ca. fünf Prozent der Signale der kompletten K-Matrix während des Integrationstests gesendet. Demzufolge wäre es interessant zu sehen welcher Anteil der Signale tatsächlich ausgetauscht wird.

Testabdeckungsmaße auf Basis von Architektur- und Schnittstellenbeschreibungen können bereits beim MiL-Test eingesetzt werden. Auch beim Integrationstest am HiL-Prüfstand ist deren Einsatz nützlich, um beispielsweise nicht kommunizierende Komponenten zu bestimmen,

## 5. Evaluierung

---

bei denen eine Kommunikation jedoch erforderlich ist. Ein Werkzeug, welches diese Abdeckung misst, sollte Architekturmodelle verarbeiten können, um die darin enthaltenen Komponenten und Kommunikationsbeziehungen den real verfügbaren Komponenten auf HiL-Ebene zuordnen zu können.

Falls E/E-Systeme durch Zustandsautomaten beschrieben sind, können die diesbezüglichen Testabdeckungsmaße eingesetzt werden, wenn die aktuell erreichten Zustände am HiL-Prüfstand ersichtlich sind. Ein Werkzeug zur Messung der Abdeckung muss erkennen können, welche E/E-Komponenten am HiL-Prüfstand an bestimmten Zuständen und Zustandsübergängen beteiligt sind. In manchen Fällen existieren zudem Komponenten, die den beschränkten Zugriff auf interne Informationen (z.B. zur Diagnose) ermöglichen. Folglich kann es möglich sein, dass dadurch Zustände erkennbar sind. Bei der Testfallermittlung mit Hilfe von Zustandsautomaten kann es jedoch auch zu Komplexitätsproblemen kommen. Daher sollten hier nur relevante Szenarien abgedeckt werden.

## 6. Zusammenfassung, Wertung und Ausblick

Dieses Kapitel fasst die Ergebnisse der Arbeit zusammen, bewertet sie und gibt einen Ausblick auf weiterführende Arbeiten.

Die folgende Auflistung zeigt die Ergebnisse der Literaturrecherche:

- Es existieren Testabdeckungsmaße für den Softwaretest (z.B. Anweisungs-, Zweig- und Pfadabdeckung).
- Es existieren Testabdeckungsmaße für den Software-Integrationstest (z.B. Schnittstellenabdeckung und Call-Coverage).
- Es existieren Testabdeckungsmaße für verschiedene Modelle (z.B. Zustandsautomaten, Architekturmodelle).
- Es wurden keine veröffentlichten Testabdeckungsmaße speziell für den black-box-orientierten Integrationstest von E/E-Systemen im Fahrzeugbereich gefunden.

Auf Grund der Ergebnisse der Literaturrecherche wurden aus dem Softwaretest bekannte Testabdeckungsmaße auf den Integrationstest von E/E-Systemen übertragen. Als Basis für die Ableitung der Testabdeckungsmaße dient immer eine Beschreibung des E/E-Systems bzw. der Kommunikation zwischen den E/E-Komponenten.

Es wurde ein Konzept zur Erweiterung der Teststrategie entworfen. Die erweiterte Teststrategie auf Basis der entwickelten Testabdeckungsmaße dient:

- ⇒ nicht als Testziel, sondern als formales zusätzliches Kriterium,
- ⇒ nicht als Ersatz für den Haupttest auf Basis der Anforderungen,
- ⇒ als Ergänzung zum funktionsorientierten Test (basierend auf der Anforderungsspezifikation) bezüglich Vollständigkeit bzw. als Testendekriterium.

Im Rahmen dieser Arbeit wurden Testabdeckungsmaße für den Integrationstest von E/E-Systemen entwickelt, die auf folgenden Beschreibungsmitteln basieren:

- Architektur- und Schnittstellenbeschreibungen,
- Zustandsautomaten,

- K-Matrizen.

Die Evaluierung der entwickelten Testabdeckungsmaße erfolgte anhand von Fallbeispielen und Diskussionsrunden mit internen Test-Experten, die im Bereich des Integrationstest am HiL tätig sind, internen Experten im Bereich Testmethoden und externen Test-Experten aus unternehmensrelevanten Testhäusern. Ergebnis der Befragungen ist, dass die im Rahmen dieser Arbeit entwickelten Testabdeckungsmaße sinnvoll sind. Voraussetzung für den Einsatz der Testabdeckungsmaße in der Praxis ist die Möglichkeit zur automatisierten Messung während des Integrationstests.

### **Ausblick**

Falls die Zustandsübergänge des Zustandsautomaten mit Bedingungen versehen sind, können die bekannten Abdeckungsmaße des Bedingungsüberdeckungstests von Software genutzt werden. Die Beschreibung der einzelnen Ausführungen der Bedingungsabdeckung ist in Liggesmeyer [Lig09] zu finden. Signale in K-Matrizen sollten gefiltert werden. Zum Zwecke der Nachverfolgung der korrekten Weiterverarbeitung können auch Kommunikationssequenzen zur Abdeckungsmessung herangezogen werden (z.B. anhand von Send-Receive-Pairs). Dies erfordert jedoch oftmals eine sehr komplexe Testfallgestaltung, da bestimmte Reaktionen erst durch eine Kette von Aktionen hervorgerufen werden.

Mögliche weiterführende Arbeiten sind demzufolge:

- Testabdeckungsmaße für den bedingungsorientierten Test von Transitionen in Zustandsautomaten
- Messbare Auswertung der Testabdeckungsmaße über eine größere Zeitspanne
- Kommunikation mit HiL-Herstellern über mögliche Automatisierung der Messungen
- Entwicklung von werkzeunterstützten Lösungen (z.B. Filterung von Signalen)



# Anhang

## A. Fragebogen

### UMFRAGE ZUR EVALUIERUNG DER TESTABDECKUNGSMAÙE FÜR DEN INTEGRATIONSTEST VON E/E-SYSTEMEN

Name: Max Mustermann
Gruppe: Intern (HiL/Funktionsmodell)/Extern(Testhaus)

Bei jeder Frage kreisen Sie diejenige Nummer rechts ein,  
die Ihrer Meinung nach am besten der Bedeutung dieses Themas entspricht.  
Verwenden Sie die oben vorgegebene Skala, um Ihre Meinung damit abzugleichen.

Frage	Bewertungsskala				
	Sehr gut	gut	Keine Meinung	schlecht	sehr schlecht
Wie schätzen Sie die Anwendbarkeit der TestabdeckungsmaÙe auf Basis von komponenten-übergreifenden Zustandsautomaten ein?	1	2	3	4	5
Wie schätzen Sie die Anwendbarkeit der TestabdeckungsmaÙe auf Basis der K-Matrix ein?	1	2	3	4	5
Wie schätzen Sie die Anwendbarkeit der TestabdeckungsmaÙe auf Basis der Architekturbeschreibungen ein?	1	2	3	4	5
Wie schätzen Sie die Aussagekraft der TestabdeckungsmaÙe auf Basis von komponenten-übergreifenden Zustandsautomaten ein?	1	2	3	4	5
Wie schätzen Sie die Aussagekraft der TestabdeckungsmaÙe auf Basis der K-Matrix ein?	1	2	3	4	5
Wie schätzen Sie die Aussagekraft der TestabdeckungsmaÙe auf Basis der Architekturbeschreibungen ein?	1	2	3	4	5
Wie schätzen Sie die Umsetzbarkeit der Automatisierbarkeit der TestabdeckungsmaÙe auf Basis von komponenten-übergreifenden Zustandsautomaten ein?	1	2	3	4	5
Wie schätzen Sie die Umsetzbarkeit der Automatisierbarkeit der TestabdeckungsmaÙe auf Basis der K-Matrix ein?	1	2	3	4	5
Wie schätzen Sie die Umsetzbarkeit der Automatisierbarkeit der TestabdeckungsmaÙe auf Basis der Architekturbeschreibungen ein?	1	2	3	4	5



# Literaturverzeichnis

- [aio12] aiotestking. *iSQI Questions and Answers*. <http://www.aiotestking.com/isqi/how-many-test-cases-are-needed-to-achieve-0-switch-coverage/>. 2012 (zitiert auf S. 17).
- [BB83] B. W. Boehm, R. Beach. „GUIDELINES FOR VERIFYING AND VALIDATING SOFTWARE REQUIREMENTS AND DESIGN SPECIFICATIONS I. OBJECTIVES“. In: 1 (1983) (zitiert auf S. 14).
- [BCSW03] A. Baresel, M. Conrad, S. Sadeghipour, J. Wegener. „The interplay between model coverage and code coverage“. In: *Proc. EuroCAST* (2003), S. 1–14. URL: [http://www.test-agentur.de/documents/Baresel%20et%20al%20EuroSTAR%202003%20\(Paper\).pdf](http://www.test-agentur.de/documents/Baresel%20et%20al%20EuroSTAR%202003%20(Paper).pdf) (zitiert auf S. 24, 25).
- [BSB08] C. Bommer, M. Spindler, V. Barr. *Softwarewartung - Grundlagen, Management und Wartungstechniken*. dpunkt.verlag, 2008. ISBN: 978-3-89864-482-2 (zitiert auf S. 10).
- [cyr13] cyrussystems. *Hardware-In-the-Loop (HIL)*. <http://www.cyrussystems.de/index.php/produkte/hardware-in-the-loop-hil>. 2013 (zitiert auf S. 16).
- [Del16] Delphi. *Umfeldererkennung und Erfordernisse für die E/E Architektur*. 2016. URL: <https://allesueberautotechnik.de/2016/03/12/umfeldererkennung-und-erfordernisse-fuer-die-ee-architektur/> (besucht am 09. 08. 2016) (zitiert auf S. 1).
- [Fra15] D. Frank Houdek e.al. „Durchgängige Fallstudie zur Evaluierung der Anwendbarkeit der SPES Methodik“. In: *SPESxt Software Plattform Embedded Systems 2020* (2015) (zitiert auf S. 46).
- [Gro16] O. M. Group. „Requirements Interchange Format ( ReqIF )“. In: July (2016) (zitiert auf S. 22).
- [Han16] Handelsblatt. *Die größten Rückrufe der Geschichte*. 2016. URL: <http://www.handelsblatt.com/unternehmen/industrie/takata-airbag-skandal-beschert-zulieferer-weitere-verluste/13533122.html> (besucht am 08. 08. 2016) (zitiert auf S. 3).
- [Has14] A. M. Hass. *Guide to Advanced Software Testing*. 2nd. Norwood, MA, USA: Artech House, Inc., 2014. ISBN: 1608078043, 9781608078042 (zitiert auf S. 27).
- [IBM16] IBM. *IBM - Rational DOORS*. 2016. URL: <http://www-03.ibm.com/software/products/de/ratidoor> (besucht am 14. 10. 2016) (zitiert auf S. 22).

- [IEE98a] IEEE. „IEEE Guide for Information Technology - System Definition - Concept of Operations (ConOps) Document“. In: *IEEE Std 1362-1998* (Dez. 1998), S. 1–24. DOI: [10.1109/IEEESTD.1998.89424](https://doi.org/10.1109/IEEESTD.1998.89424) (zitiert auf S. 22).
- [IEE98b] IEEE. „IEEE Recommended Practice for Software Requirements Specifications“. In: *IEEE Std 830-1998* (Okt. 1998), S. 1–40. DOI: [10.1109/IEEESTD.1998.88286](https://doi.org/10.1109/IEEESTD.1998.88286) (zitiert auf S. 22).
- [IEEE90] „IEEE Standard Glossary of Software Engineering Terminology“. In: *IEEE Std 610.12-1990* (Dez. 1990), S. 1–84. DOI: [10.1109/IEEESTD.1990.101064](https://doi.org/10.1109/IEEESTD.1990.101064) (zitiert auf S. 10, 12).
- [imc15] imc-berlin. *Aufbau Laborfahrzeug*. [http://www.imc-berlin.de/fileadmin/Public/Products/Measurement\\_Hardware/imc\\_CANSAS/imc\\_CANSAS-IHR/BFFT\\_Aufbau\\_Laborfahrzeug.jpg](http://www.imc-berlin.de/fileadmin/Public/Products/Measurement_Hardware/imc_CANSAS/imc_CANSAS-IHR/BFFT_Aufbau_Laborfahrzeug.jpg), 17.07.2015. 2015 (zitiert auf S. 15).
- [Inn] T. T.-. und InnovationsConsult GmbH. *ISTQB-Testprozess*. URL: <http://www.ates-netzwerk.de/istqb-testprozess.html> (zitiert auf S. 27).
- [ISO10] „Systems and software engineering – Vocabulary“. In: *ISO/IEC/IEEE 24765:2010(E)* (Dez. 2010), S. 1–418. DOI: [10.1109/IEEESTD.2010.5733835](https://doi.org/10.1109/IEEESTD.2010.5733835) (zitiert auf S. 12).
- [ISO11a] „Road vehicles – Functional safety“. In: *ISO 26262* (2011) (zitiert auf S. 24).
- [ISO11b] „Road vehicles - Functional safety – Vocabulary“. In: *ISO 26262-1:2011(E)* (Nov. 2011), S. 1–32 (zitiert auf S. 7).
- [ITW16] ITWissen. *COTS :: commercial off-the-shelf :: ITWissen.info*. 2016. URL: <http://www.itwissen.info/definition/lexikon/commercial-off-the-shelf-COTS.html> (zitiert auf S. 23).
- [KDJL01] H. Kelly J., V. Dan S., C. John J., R. Leanna K. *A Practical Tutorial on Modified Condition/Decision Coverage*. Techn. Ber. 2001 (zitiert auf S. 23).
- [Lig09] P. Liggesmeyer. *Software-Qualität: Testen, Analysieren und Verifizieren von Software*. 2009. ISBN: 3827420563. DOI: [10.1007/978-3-540-76323-9](https://doi.org/10.1007/978-3-540-76323-9) (zitiert auf S. 9, 10, 18, 23, 50).
- [Nör12] R. Nörenberg. *Effizienter Regressionstest von E/E-Systemen nach ISO 26262* -. Karlsruhe: KIT Scientific Publishing, 2012. ISBN: 978-3-866-44842-1 (zitiert auf S. 8, 24, 25).
- [PM05] J. F. Power, J. A. McQuillan. „A Survey of UML-Based Coverage Criteria for Software Testing“. In: *National University of Ireland, Department of Computer Science, Technical Report Series* (2005) (zitiert auf S. 24, 25).
- [RHPL08] C. Robinson-Mallett, R. M. Hierons, J. Poore, P. Liggesmeyer. „Using communication coverage criteria and partial model generation to assist software integration testing“. In: *Software Quality Journal* 16.2 (2008), S. 185–211 (zitiert auf S. 24, 25).
- [RR16] J. Robertson, S. Robertson. *Requirements Specification Template*. 2016. URL: <http://www.volere.co.uk/template.htm> (besucht am 14. 10. 2016) (zitiert auf S. 22).

- [Sax08] E. Sax. *Automatisiertes Testen Eingebetteter Systeme in der Automobilindustrie*. München, Wien: Carl Hanser Verlag, 2008. ISBN: 978-3-446-41635-2 (zitiert auf S. 16).
- [SL10] A. Spillner, T. Linz. *Basiswissen Softwaretest*. Heidelberg: d.punkt Verlag, 2010 (zitiert auf S. 12).
- [Spi08] A. Spillner. *Praxiswissen Softwaretest - Testmanagement: Aus- und Weiterbildung zum Certified Tester - Advanced Level nach ISTQB-Standard*. dpunkt-Verlag, 2008. ISBN: 9783898645577. URL: <https://books.google.de/books?id=GUK8PQAACAAJ> (zitiert auf S. 16).
- [ST12] T. Streichert, M. Traub. *Elektrik/Elektronik-Architekturen im Kraftfahrzeug*. VDI-Buch. Berlin: Springer Vieweg, 2012. ISBN: 978-3-642-25477-2. DOI: [10.1007/978-3-642-25478-9](https://doi.org/10.1007/978-3-642-25478-9) (zitiert auf S. 2, 21).
- [SZ13] J. Schäuuffele, T. Zurawka. *Automotive Software Engineering - Grundlagen, Prozesse, Methoden und Werkzeuge effizient einsetzen*. 5. Aufl. Berlin Heidelberg New York: Springer-Verlag, 2013. ISBN: 978-3-834-82470-7 (zitiert auf S. 19).
- [VDI93] VDI-Gemeinschaftsausschuß Industrielle Systemtechnik (VDI-GIS). *Software-Zuverlässigkeit*. Bd. 1542. 9. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993, S. 33–36. ISBN: 978-3-540-62305-2. DOI: [10.1007/978-3-642-95800-7](https://doi.org/10.1007/978-3-642-95800-7). URL: <http://link.springer.com/10.1007/978-3-642-95800-7> (zitiert auf S. 9).
- [Vecka] Vector. *PREEvision - Modellbasierte Elektrik-/Elektronik-Entwicklung vom Architektorentwurf bis zur Serienreife*. k.a. (Zitiert auf S. 43).
- [WCO03] Y. Wu, M.-h. Chen, J. Offutt. „UML-based Integration Testing for Component-based Software“. In: *Engineering* 2580 (2003), S. 251–260. ISSN: 03029743. DOI: [10.1007/3-540-36465-X\\_24](https://doi.org/10.1007/3-540-36465-X_24). URL: <http://www.springerlink.com/index/3EAK2Y7PWXP27CKW.pdf> (zitiert auf S. 24, 25).
- [WES+13] M. Winter, M. Ekssir-Monfared, H. M. Sneed, R. Seidl, L. Borner. *Der Integrationstest: Von Entwurf und Architektur zur Komponenten- und Systemintegration*. München: Carl Hanser Verlag, 2013. ISBN: 978-3-446-42951-2 (zitiert auf S. 11–13, 23, 25).
- [Win99] D.-I. M. Winter. *Qualitätssicherung für objektorientierte Software : Anforderungsermittlung und Test gegen die Anforderungsspezifikation*. September. 1999. ISBN: 3898250318 (zitiert auf S. 24, 25).
- [WPC01] Y. Wu, D. Pan, M.-H. Chen. „Techniques for testing component-based software“. In: *Proceedings Seventh IEEE International Conference on Engineering of Complex Computer Systems* (2001), S. 222–232. ISSN: 1050-4729. DOI: [10.1109/ICECCS.2001.930181](https://doi.org/10.1109/ICECCS.2001.930181) (zitiert auf S. 24, 25).

Alle URLs wurden zuletzt am 09. 10. 2016 geprüft.



## **Erklärung**

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

---

Ort, Datum, Unterschrift