

# Certified Automotive Software Tester Lehrplan

## (CTFL<sup>®</sup>-CAST)

Version 2015 (1.1 – FINAL) vom 14.06.2015

---

**German Testing Board e.V.**

---

Deutschsprachige Ausgabe.  
Herausgegeben durch German Testing  
Board e.V.

### **Urheberrecht © 2015, German Testing Board e.V. (GTB)**

Die Autoren und German Testing Board (GTB) haben folgenden Nutzungsbedingungen zugestimmt:

Jede Einzelperson und Seminaranbieter darf den Lehrplan als Grundlage für Seminare verwenden, sofern die Inhaber der Urheberrechte als Quelle und Besitzer des Urheberrechts anerkannt und benannt werden. Des Weiteren darf der Lehrplan zu Werbezwecken erst nach der Akkreditierung durch das German Testing Board verwendet werden.

Jede Einzelperson oder Gruppe von Einzelpersonen darf den Lehrplan als Grundlage für Artikel, Bücher oder andere abgeleitete Veröffentlichungen verwenden, sofern die Autoren und das German Testing Board als Quelle und Besitzer des Urheberrechts genannt werden.

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Die Verwertung ist – soweit sie nicht ausdrücklich durch das Urheberrechtsgesetz (UrhG) gestattet ist – nur mit Zustimmung der Berechtigten zulässig. Dies gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmung, Einspeicherung und Verarbeitung in elektronischen Systemen, öffentliche Zugänglichmachung.

### **Eingetragene Marken**

CTFL® ist eine eingetragene Marke des German Testing Board (GTB) e.V.

GTB® ist eine eingetragene Marke des German Testing Board (GTB) e.V.

ISTQB® ist eine eingetragene Marke des International Software Testing Qualifications Board.

AUTOMOTIVE SPICE® ist eine eingetragene Marke des Verbandes der Automobilindustrie (VDA).

## Änderungsübersicht

Version	Datum	Bemerkung
1.0	19.01.2011	Erstausgabe V1.0 (Autor: Dr. Hendrik Dettmering, dettmering@prozesswerk.eu; entwickelt im Auftrag gasq) Die Nutzungsrechte wurden vollständig übertragen an GERMAN TESTING BOARD e.V.
1.1 (BETA)	31.01.2014	Inhaltlicher Abgleich mit dem deutschsprachigen Foundation Level Lehrplan 2011 und dem Glossar V2.2 Layout und Gestaltung an den ISTQB Foundation Level Lehrplan 2011 angepasst Lernziele überarbeitet Reihenfolge der Kapitel und Abschnitte geändert Business Outcomes (geschäftlicher Nutzen) ergänzt Literaturverzeichnis erweitert Glossar erweitert Änderungen für V1.1 wurden vorgenommen von: Horst.Pohlmann@german-testing-board.info.
1.1 (BETA 2)	31.05.2014	Änderungen nach Review durch die Mitglieder der Arbeitsgruppe
1.1 (BETA 3)	31.08.2014	Änderungen nach Review durch die Mitglieder der Arbeitsgruppe
1.1 (BETA 4)	30.09.2014	Layout Korrektur (an CTAL Lehrplan angepasst) Änderungen nach Review durch Ralf Bongard (Überarbeitungsmodus)
1.1 (BETA 5)	09.10.2014	Layout angepasst, Änderungen angenommen
1.1 (BETA 6)	17.10.2014	Im gemeinsamen Review der CAST Arbeitsgruppe in Stuttgart wurden folgende Änderungen vorgenommen: Präzisierungen der Texte 1.2.4.4, 2.2.3.2, 2.3.2, BO CAST1 Glossar-begriff Betriebsmodus diverse redaktionelle Änderungen
1.1 (BETA 7)	31.12.2014	Alle Änderungen aus BETA in die Endfassung integriert; Vorschläge des Technical Writers überprüft und angenommen.
1.1 (BETA 8a)	24.05.2015	Änderungen aus dem CTFL-CAST Meeting vom 15.03.2015 und gemeldete Befunde in MANTIS bearbeitet. Keine inhaltliche Änderungen, sondern nur Verbesserungen im Wording.
1.1 (BETA 8B)	31.05.2015	Verbesserungen im Wording
1.1 (Final)	14.06.2015	Freigabe gem. CTFL-CAST Meeting vom 15.03.2015 (München)

## Inhaltsverzeichnis

Änderungsübersicht.....	3
Dank .....	6
Einführung in den CTFL®-CAST Syllabus .....	7
Zweck des Dokuments .....	7
ISTQB® Certified Tester, Foundation Level, Automotive Extension .....	7
Lernziele/Kognitive Stufen des Wissens .....	7
Begriffe .....	7
Die Prüfung.....	7
Akkreditierung.....	8
Detaillierungsgrad .....	8
Lehrplanaufbau.....	8
Foundation Level Extension (CTFL®-CAST) .....	9
Aufbau .....	9
1 Baustein 1: Relevante Normen und Standards für das Testen von E/E-Systemen – [400 Minuten]11	
1.1 Einführung – [5 Minuten].....	11
1.2 Functional Safety Essentials (K2) – [140 Minuten].....	11
1.2.1 Einführung Functional Safety - ISO 26262 (15 min) .....	12
1.2.2 Zielsetzung der ISO 26262 (30 min) .....	12
1.2.3 Betriebliche Relevanz (15 Minuten) .....	12
1.2.4 AUTOMOTIVE Safety Integrity Level (80 min) .....	13
1.3 AUTOMOTIVE SPICE Essentials (K2) – [210 Minuten].....	15
1.3.1 Einführung Automotive SPICE und Zielsetzung (15 Minuten) .....	16
1.3.2 Reifegraddimension (30 Minuten) .....	16
1.3.3 Prozessdimension (30 Minuten) .....	16
1.3.4 Testmanagement in AUTOMOTIVE SPICE (75 Minuten) .....	17
1.3.5 Test- und Verifikationsaktivitäten in den Engineering- und Supporting-Prozessen nach AUTOMOTIVE SPICE (60 Minuten) .....	17
1.4 Vergleich AUTOMOTIVE SPICE und ISO 26262 (K2) – [25 Minuten].....	18
1.5 Übungen/Verständnisfragen/Fallstudien (K3) - [20 Minuten] .....	18
2 Baustein 2: Testen im E/E-Entwicklungsprozess - [380 Minuten].....	19
2.1 Softwareentwicklung und -test im Automotive-Umfeld (K2) - [150 Minuten] .....	19
2.1.1 Grundlagen des Softwaretestens im Automobilbau (30 Minuten) .....	19
2.1.2 Automotive PEP (30 Minuten).....	20
2.1.3 Musterstände (30 Minuten) .....	20
2.1.4 Einbindung der Softwareentwicklung in den automobilen Produktentstehungsprozess (30 Minuten) .....	20

2.1.5	Integrationsstufenkonzept (30 Minuten) .....	20
2.2	Testen in virtueller Umgebung (K3) – [120 Minuten].....	21
2.2.1	Einführung und Zielsetzung des Testens in virtueller Umgebung (45 Minuten).....	21
2.2.2	Hardware in the Loop (45 Minuten) .....	21
2.2.3	Software in the Loop (30 Minuten).....	22
2.2.4	Abgrenzung zum Open-Loop-Test [informativ].....	23
2.3	Besonderheiten in der automobilen Softwareentwicklung (K2) - [60 Minuten].....	23
2.3.1	AUTOSAR (30 Minuten) .....	23
2.3.2	Betriebsmodi (10 Minuten).....	23
2.3.3	Variantenvielfalt (20 Minuten) .....	23
2.4	Übungen/Verständnisfragen/Fallstudien (K3) - [30 Minuten] .....	24
3	Literaturverzeichnis .....	25
4	Anhang A – Glossar und Abkürzungsverzeichnis.....	27
4.1	Glossar CTFL-EXTENSION “Certified Automotive Software Tester (CTFL®-CAST)” .....	27
4.2	Abkürzungen.....	32
5	Anhang B – Hintergrundinformation zum Lehrplan.....	33
5.1	Zur Geschichte dieses Dokuments.....	33
5.2	Voraussetzungen für die Ergänzung (Extension) zur Basisstufe .....	33
6	Anhang C – Lernziele/Kognitive Ebenen des Wissens.....	34
6.1	Taxonomiestufe 1: Kennen (K1).....	34
6.2	Taxonomiestufe 2: Verstehen (K2).....	34
6.3	Taxonomiestufe 3: Anwenden (K3) .....	35
6.4	Taxonomiestufe 4: Analysieren (K4) .....	35
7	Anhang D – Verwendete Regeln bei der Erstellung des Lehrplans.....	36
7.1	Allgemeine Regeln.....	36
7.2	Aktualität .....	36
7.3	Lernziele .....	36
7.4	Gesamtstruktur .....	36
7.5	Referenzen .....	37
7.6	Informationsquellen .....	37
8	Anhang E – Hinweise für Ausbildungsanbieter .....	38
9	Anhang F – Release Notes .....	39
10	Index.....	40

## Dank

Das German Testing Board (GTB) dankt ebenso dem Reviewteam der deutschsprachigen Fassung 2014, V1.1: Horst Pohlmann (Leitung), Ralf Bongard, Klaudia Dussa-Zieger, Kai Lepler, Uwe Hehn, Jork Warnecke, Alexander Schulz, Thorsten Geiselhart, Olaf Janßen.

## Allgemeiner Hinweis:

Als [informativ] gekennzeichnete Abschnitte des vorliegenden Lehrplans sind mit der vorliegenden Version NICHT prüfungsrelevant.

Beispiel:

2.2.4 Abgrenzung zum Open-Loop-Test [informativ]

## Einführung in den CTFL®-CAST Syllabus<sup>1</sup>

### Zweck des Dokuments

Dieser Lehrplan definiert eine Ergänzung zur Basisstufe (Extension to Foundation Level) des Softwaretest-Ausbildungsprogramms des International Software Testing Qualifications Board (im Folgenden kurz ISTQB® genannt). Anhand des vorliegenden Lehrplans erstellen Ausbildungsanbieter ihre Kursunterlagen und legen eine angemessene Unterrichtsmethodik für die Akkreditierung fest. Die Lernenden bereiten sich anhand des Lehrplans auf die Prüfung vor.

Weitere Informationen über Geschichte und Hintergrund des vorliegenden Lehrplans sind im Anhang A dieses Lehrplans aufgeführt.

### ISTQB® Certified Tester, Foundation Level, Automotive Extension

Die vorliegende Ergänzung zur Basisstufe des Certified Tester-Ausbildungsprogramms soll alle in das Thema Softwaretesten involvierten Personen in der Domäne „Automotive“ ansprechen. Das schließt Personen in Rollen wie Tester, Testanalysten, Testingenieure, Testberater, Testmanager, Abnahmetester und Softwareentwickler mit ein. Die Basisstufe richtet sich ebenso an Personen in der Rolle Projektleiter, Qualitätsmanager, Softwareentwicklungsmanager, Systemanalytiker (Business-Analysten), IT-Leiter oder Managementberater, welche sich ein Basiswissen und Grundlagenverständnis über das Thema Softwaretesten in der Domäne „Automotive“ erwerben möchten.

### Lernziele/Kognitive Stufen des Wissens

Jeder Abschnitt dieses Lehrplans ist einer kognitiven Stufe zugeordnet:

- K1: sich erinnern
- K2: verstehen
- K3: anwenden
- K4: analysieren

Weitere Details und Beispiele von Lernzielen finden Sie in Anhang B.

### Begriffe

Alle Begriffe, die im Absatz direkt unter der Überschrift unter „Begriffe“ genannt werden, sollen wiedergegeben werden können (K1), auch wenn das in den Lernzielen nicht explizit angegeben ist. Es gelten die Definitionen des ISTQB-Glossars bzw. der nationalen Übersetzung in der jeweils freigegebenen Fassung (inkl. der zusätzlichen Begriffe aus dem vorliegenden Lehrplan).

### Die Prüfung

Auf diesem Lehrplan basiert eine zusätzliche Prüfung für das domänenspezifische Zertifikat der CTFL-Extension „Certified Automotive Software Tester (CTFL®-CAST)“. Eine Prüfungsfrage kann Stoff aus mehreren Kapiteln des Lehrplans abfragen. In der Regel ist jede Prüfungsfrage einem Lernziel zugeordnet mit der Ausnahme der Fragen, die einem Schlüsselbegriff zugeordnet sind. Das Format der Prüfung ist Multiple Choice. Prüfungen können unmittelbar im Anschluss an einen akkreditierten

---

<sup>1</sup> Der Text wurde im Wesentlichen übernommen aus dem ISTQB® CTFL Lehrplan 2011, siehe [www.german-testing-board.info](http://www.german-testing-board.info).

Ausbildungslehrgang oder Kurs, aber auch unabhängig davon (z. B. in einem Prüfzentrum oder einer öffentlich zugänglichen Prüfung) abgelegt werden. Die Teilnahme an einem Kurs stellt keine Voraussetzung für das Ablegen der Prüfung dar. Die vom German Testing Board zugelassenen Prüfungsanbieter sind die für die Extension „Certified Automotive Software Tester (CTFL®-CAST)“ aufgelisteten Prüfungsanbieter.

## Akkreditierung<sup>2</sup>

Das German Testing Board akkreditiert Ausbildungsanbieter, deren Ausbildungsunterlagen entsprechend diesem Lehrplan aufgebaut sind. Die Akkreditierungsrichtlinien sind bei diesem nationalen Board erhältlich. Ein akkreditierter Kurs ist als zu diesem Lehrplan konform anerkannt und darf als Bestandteil eine Zusatzprüfung enthalten.

Weitere Hinweise für Ausbildungsanbieter sind im Anhang E enthalten.

## Detaillierungsgrad

Der Detaillierungsgrad dieses Lehrplans erlaubt konsistentes Lehren und Prüfen. Um dieses Ziel zu erreichen, enthält dieser Lehrplan Folgendes:

- allgemeine Lernziele, welche die Intention der (erweiterten) Basisstufe beschreiben
- Inhalte, die zu lehren sind, mit einer Beschreibung und, wo notwendig, Referenzen zu weiterführender Literatur
- Lernziele für jeden Wissensbereich, welche das beobachtbare kognitive Ergebnis der Schulung und die zu erzielende Einstellung des Teilnehmers beschreiben
- eine Liste von Begriffen, welche der Teilnehmer wiedergeben und verstehen soll
- eine Beschreibung der wichtigen zu lehrenden Konzepte, einschließlich Quellen wie anerkannte Fachliteratur, Normen und Standards

Der Lehrplan ist keine vollständige Beschreibung des Wissensgebiets „Testen für softwarebestimmte Systeme in automobilen Elektronikentwicklungsprojekten“. Er reflektiert lediglich den nötigen Umfang und Detaillierungsgrad, welcher für die Lernziele relevant ist.

## Lehrplanaufbau

Der Lehrplan besteht aus zwei Hauptkapiteln. Jeder Haupttitel eines Kapitels zeigt die anspruchsvollste Lernzielkategorie / höchste kognitive Stufe, welche mit dem jeweiligen Kapitel abgedeckt werden soll und legt die Unterrichtszeit fest, welche in einem akkreditierten Kurs mindestens für dieses Kapitel aufgewendet werden muss.

### Beispiel:

### Testen in virtueller Umgebung (K2) - [115 Minuten]

Das Beispiel zeigt, dass im Kapitel „Testen in virtueller Umgebung (K2)“ K1<sup>3</sup> und K2 (aber nicht K3) erwartet werden und 115 Minuten für das Lehren des Materials in diesem Kapitel vorgesehen sind.

Die Lernziele legen fest, was Sie nach Beenden des jeweiligen Abschnitt/Kapitels/Moduls gelernt haben sollten. Jedes Kapitel enthält eine Anzahl von Unterkapiteln. Jedes Unterkapitel kann wiederum Lernziele und einen Zeitrahmen vorgeben. Wird bei einem Unterkapitel keine Zeit angegeben, so ist diese im Oberkapitel bereits enthalten.

<sup>2</sup> Der Text wurde i.w. übernommen aus dem ISTQB® CTFL Lehrplan 2011.

<sup>3</sup> Ein Lernziel einer höheren Taxonomie-Stufe impliziert die Lernziele der tieferen Stufen.



## Foundation Level Extension (CTFL®-CAST)

### Einführung Certified Automotive Software Tester [20 Minuten]

#### Geschäftlicher Nutzen

In diesem Abschnitt wird der geschäftliche Nutzen (Business Outcomes nach ISTQB®) aufgelistet, den man von Kandidaten mit einer zusätzlichen Zertifizierung als Certified Automotive Software Tester (CTFL®-CAST) erwarten kann.

Ein Certified Automotive Software Tester (CTFL®-CAST) kann im Rahmen des Testens von E/E-Systemen

CAST01 ... in einem Testteam effektiv mitarbeiten („collaborate“),

CAST02 ... die aus dem Foundation Level bekannten Testverfahren an die automotivespezifischen Projektbedingungen anpassen („adapt“),

CAST03 ... bei der Auswahl von angemessenen Testverfahren im Automotive-Kontext die grundlegenden Anforderungen der relevanten Normen und Standards (Automotive SPICE®, ISO 26262, etc.) berücksichtigen („select“),

CAST04 ... das Testteam bei der risikoorientierten Planung der Testaktivitäten im Automotive-Kontext unterstützen und dabei bekannte Elemente der Strukturierung (z. B. Integrationsstufen) anwenden („support & apply“),

CAST05 ... die virtuellen Testmethoden (z. B. HiL, SiL, MiL, etc.) in automotivespezifischen Testumgebungen anwenden („apply“).

## Aufbau

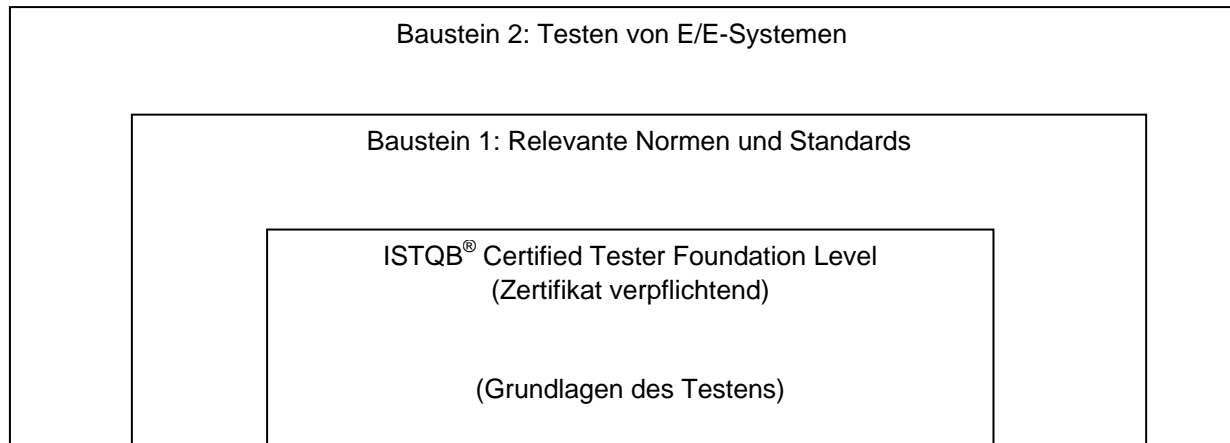
Der zertifizierte Automotive Software Tester hat einen Überblick über wichtige Testverfahren und Normen, die im Rahmen des Testens von E/E-Systemen zum Einsatz kommen. Der vorliegende Lehrplan umfasst folgende Bausteine:

- relevante Normen für das Testen von E/E-Systemen (ISO 26262, Automotive SPICE, ...)
- Testen von softwarebestimmten E/E-Systemen, bestehend aus
  - Grundlagen: PEP, Musterstände, Integrationsstufenkonzept, etc.
  - Testen in virtueller Umgebung: HiL, SiL, etc.
  - Besonderheiten: AUTOSAR, Betriebsmodi, Variantenvielfalt, etc.

So stellt der Lehrplan zur CTFL®-Extension „Certified Automotive Software Tester (CTFL®-CAST)“ einen ergänzenden Ausbildungszweig<sup>4</sup> komplementär zu den bestehenden Foundation Level des International Software Testing Qualifications Board (ISTQB®) dar.

---

<sup>4</sup> Das Zertifikat zum ISTQB® Foundation Level wird vorausgesetzt.



Der Baustein „Grundlagen des Testens“ in der CTFL-Extension „Certified Automotive Software Tester (CTFL®-CAST)“ ist ein optionaler Baustein für diejenigen, die bereits über das Certified-Tester-Foundation-Level-Zertifikat verfügen und lediglich eine „Auffrischung“ benötigen, weil sie bereits vor vielen Jahren die Foundation Level-Prüfung abgelegt haben. Diese Anteile sollten aus dem Foundation Level bekannt sein und sind für den Erweiterungsbaustein „Automotive“ nur implizit d. h. im Anwendungskontext Automotive prüfungsrelevant.

# 1 Baustein 1: Relevante Normen und Standards für das Testen von E/E-Systemen – [400 Minuten]

## 1.1 Einführung – [5 Minuten]

Die Entwicklung von softwarebestimmten Systemen in Automotive-Elektronik-Entwicklungsprojekten wird getrieben durch Normen und Standards wie AUTOMOTIVE SPICE®, ISO 26262 und AUTOSAR. Diese Standards stellen Anforderungen an einen systematischen Test auf allen Stufen: vom Komponententest, über den Integrationstest, zum Softwaretest und dem Systemtest.

## 1.2 Functional Safety Essentials (K2) – [140 Minuten]

### Begriffe

AUTOMOTIVE Safety Integrity Level (ASIL), Sicherheitsziel (nach ISO 26262), Sicherheitsfunktion (nach IEC 61508)

### 1.2.1 Einführung Functional Safety - ISO 26262 (15 min)

LO-1.2.1 Sie können begründen, warum die ISO 26262 für das Testen von E/E-Systemen relevant ist, und wissen, welche Aspekte durch sie adressiert werden. (K2)

### 1.2.2 Zielsetzung der ISO 26262 (30 min)

LO-1.2.2 Sie können die wichtigsten Adressaten der ISO 26262 benennen und die Motivation für die Norm beschreiben. (K2)

### 1.2.3 Betriebliche Relevanz (15 Minuten)

LO-1.2.3 Sie kennen die betrieblichen Ziele der ISO 26262 und können die wichtigsten Aspekte der Norm beschreiben. (K2)

### 1.2.4. AUTOMOTIVE Safety Integrity Level (80 min)

LO-1.2.4.1 Sie kennen die Grundlagen des ASIL und können die wichtigsten Bestandteile und Parameter beschreiben. (K2)

LO-1.2.4.2 Sie können beschreiben, wie bei der Berechnung des ASIL vorzugehen ist und wie der Bewertungsprozess gestaltet sein muss. (K2)

LO-1.2.4.3 Sie haben einen Überblick über die praktischen Ansätze bei der Risikobewertung und können eine Identifizierung und eine Bewertung von Risiken unterstützen. (K2)

LO-1.2.4.4 Sie können die Bereiche benennen, in denen eine ASIL-Berechnung verwendet werden kann und welche Vorteile dadurch erreicht werden. (K1)

### 1.2.1 Einführung Functional Safety - ISO 26262 (15 min)

Die ISO 26262 konkretisiert als Norm zur funktionalen Sicherheit von Straßenfahrzeugen die IEC 61508 in der Automobilindustrie.

Die ISO 26262 adressiert in der aktuell veröffentlichten Ausgabe 2011 Personenkraftwagen bis 3,5 t zulässiges Gesamtgewicht.

Dieser Lernabschnitt ordnet die Norm zeitlich und rechtlich ein, gibt einen Überblick über ihre Struktur und Inhalte und betrachtet ausgewählte Aspekte, die sich beim Softwaretesten ergeben.

Die ISO 26262 („Road vehicles – Functional safety“)

- ist eine ISO-Norm für sicherheitsrelevante elektrische/elektronische Systeme in Kraftfahrzeugen,
- ist ein auch für den Softwaretest von sicherheitsrelevanten Anteilen relevantes Prozess-Rahmenwerk und Vorgehensmodell zusammen mit geforderten Aktivitäten und Arbeitsprodukten und (definiert via Methodentabellen die anzuwendenden Testverfahren, die je nach ASIL als „optional, empfohlen oder dringend empfohlen“ eingestuft werden und)
- ermöglicht mit der Einhaltung der Norm die Umsetzung der Maßnahmen für die Funktionale Sicherheit eines elektrischen/elektronischen Systems im Kraftfahrzeug.

### 1.2.2 Zielsetzung der ISO 26262 (30 min)

Ziel der ISO 26262 ist es, Anforderungen bezüglich Sicherheit vorzugeben. Dabei soll diese:

- weder den jeweiligen Lösungsraum allzu sehr einschränken,
- weder Innovation noch Wettbewerbsdifferenzierung behindern
- Wettbewerbsverzerrungen vermeiden

In der Automobilindustrie setzen sich Modulstrategien immer stärker durch. Hierdurch kommen sehr ähnliche Systeme in verschiedenen Fahrzeugklassen zum Einsatz. Beispielsweise weichen Fensterheber für Nutzfahrzeuge nur geringfügig von denen in PKW verwendeten Fensterhebern ab.

Da die ISO 26262 Nutzfahrzeuge jedoch nicht explizit adressiert (ebenso Busse, Motorräder etc.), ist hier die IEC 61508 weiterhin die formaljuristisch gültige Norm. Fahrzeugklassenübergreifende Systeme sind also potentiell nach mehreren Sicherheitsstandards zu entwickeln.

### 1.2.3 Betriebliche Relevanz (15 Minuten)

Zum Zeitpunkt der Veröffentlichung spiegelt eine Norm den Stand von Wissenschaft und Technik wider. Da die Neuigkeiten und Innovationen schneller voranschreiten als die entsprechende Norm weiterentwickelt wird, genügt es aber nicht, lediglich die Norm umzusetzen. Normerfüllung ist zum Nachweis der Einhaltung des Stands von Wissenschaft und Technik notwendig, aber nicht hinreichend.

Werden die Anforderungen einer Norm jedoch nicht erfüllt, und es kommt in einem Fall der Produkthaftung zum Vorwurf, das Produkt entspräche nicht dem Stand von Wissenschaft und Technik, so kann eine Beweislastumkehr angestrebt werden. Dies kann sich beliebig kompliziert gestalten. Daher sollte ein betriebliches Interesse zur Einhaltung der Norm vorhanden sein, um unberechenbare Produkthaftungsrisiken zu reduzieren. Alle Produkte müssen ab Zeitpunkt der Veröffentlichung der Norm bereits nach den darin geforderten Entwicklungsprozessen entwickelt werden und die geforderten Produkteigenschaften besitzen (vgl. auch [Saul2011]).

## 1.2.4 AUTOMOTIVE Safety Integrity Level (80 min)

### 1.2.4.1 Aufbau Automotive SIL (15 min)

Der „AUTOMOTIVE Safety Integrity Level (ASIL)“ bewertet die Sicherheitsrelevanz einer Funktion. Hierbei werden drei Parameter berücksichtigt (vgl. [Saul2011]):

- **Eintrittswahrscheinlichkeit eines** (Exposure (E)): die Häufigkeit der Situationen, in denen die Fehlerwirkung auftritt
- **Beherrschbarkeit des Fehlers** (Controllability (C)): die Beherrschbarkeit der Fehlerwirkung, wenn sie zum Tragen kommt.
- **Schwere eines Fehlers** (Severity (S)): das Schadensausmaß, wenn die Fehlerwirkung nicht beherrscht werden kann

### 1.2.4.2 Vorgehen bei der ASIL-Berechnung (15 min)

Basierend auf den drei Parametern Eintrittswahrscheinlichkeit (Exposure (E)), Beherrschbarkeit des Fehlers (Controllability (C)) und Schwere des Fehlers (Severity (S)) wird der ASIL auf einer Skala A bis D bestimmt, wobei A die niedrigste und D die höchste Einstufung darstellt<sup>5</sup>. Die Anforderungen der ISO 26262 sind schließlich in Abhängigkeit des ermittelten ASIL umzusetzen.

Die ISO 26262 hält eine Methode zur Bestimmung des Automotive Safety Integrity Level (ASIL) vor; jedoch lassen die drei Parameter der Bewertungsgrundlage Eintrittswahrscheinlichkeit (Exposure E), Beherrschbarkeit des Fehlers (Controllability C) und Schwere des Fehlers (Severity S) einen großen Interpretationsspielraum zu (vgl. [Saul2011]).

Bei der Beherrschbarkeit des Fehlers (Controllability C) und der Schwere des Fehlers (Severity S) spielt zudem die konkrete Fahrzeugkonfiguration eine Rolle. Diese wird durch die Methodik der ISO 26262 nur implizit berücksichtigt. Herausforderung bei der ASIL-Einstufung ist also, eine Methodik zu finden, nach der sich ein möglichst konsistentes „ASIL-Gefüge“ ergibt (vgl. [Saul2011]).

### 1.2.4.3 Risikobewertungsmethoden (35 min)

Testen dient der Qualitätssicherung. Um effizient testen zu können, ist die Kenntnis über das Risiko einer Software unabdingbar. Die ISO 26262 beinhaltet eine Methode der Risikobewertung basierend auf Informationen zur Schadeneintrittswahrscheinlichkeit. Die Schwierigkeit hierbei ist, dass die Informationsgewinnung zu einem frühen Zeitpunkt in der Produktentstehung erfolgen muss. Es wird zwischen folgenden methodischen Bereichen der Risikobewertung unterschieden:

- Ermittlung von Schadensausmaßen
- Ermittlung von Schadenshäufigkeiten
- Abschätzung des Ausgleichsrisikos

Die folgenden bekannten Ansätze können Grundlage für die praktische Unterstützung bei der Risikobewertung sein:

- HAZOP
- FTA und Auswirkungsanalyse

### 1.2.4.4 Einsatz von ASIL (15 min)

Die Bestimmung des ASIL erfolgt in einer frühen Projektphase. Die damit verbundenen Aktivitäten bedeuten eine zeitintensive Arbeit in den frühen Phasen der Produktentwicklung und Testplanung durch den Auftraggeber.

---

<sup>5</sup> Unterhalb von ASIL A bewertete Anteile werden durch das Qualitätsmanagement-Systems eines Unternehmens behandelt.

Durch eine ausführlich durchgeführte Risikoanalyse können die Tests im Sinne des risikobasierten Testens priorisiert werden, wodurch Zeit und Ressourcen geschont werden können, ohne wesentliche Einschnitte in der Produktqualität hinnehmen zu müssen. Durch die Kenntnis des ASIL sind möglicherweise Vorteile in den folgenden Bereichen, z. B. durch Anpassung des Aufwands, erreichbar:

- Planungs- und Zeitaspekte
- Bestimmung der Teststufen (Vehicle Test bis Software Unit Test)
- je nach ASIL-Stufe zu verwendende Methoden und Metriken
- Vergleichbarkeit und Standardisierung zwischen Projekten, Firmen, Lieferanten

Die Vorgaben der ISO 26262 ermöglichen eine klare Festlegung von Maßnahmen. Es findet also keine Diskussion oder langwierige Überlegung zu evtl. notwendigen Maßnahmen statt. Der Aufwand ist einfacher abzuschätzen und u.U. wird ein hoher Testaufwand für einfache ASIL's vermieden.

## 1.3 AUTOMOTIVE SPICE Essentials (K2) – [210 Minuten]

### Begriffe

Reifegrad, Testbarkeit, Testbarkeitsanalyse, Reifegraddimension, Prozessdimension, Assessment

#### 1.3.1 Einführung Automotive SPICE und Zielsetzung (15 Minuten)

LO-1.3.1.1 Sie kennen die Historie von AUTOMOTIVE SPICE und können die Ziele, sowie die Dimensionen von AUTOMOTIVE SPICE benennen. (K1)

#### 1.3.2 Reifegraddimension (30 Minuten)

LO-1.3.2.1 Sie können die sechs Prozessreifestufen im Reifegradmodell von AUTOMOTIVE SPICE benennen. (K1)

#### 1.3.3 Prozessdimension (30 Minuten)

LO-1.3.3.1 Sie können die Prozesse in den Prozessbereichen „organisatorische Prozesse“, „primäre Prozesse“ und „unterstützende Prozesse“ im Reifegradmodell von AUTOMOTIVE SPICE erläutern. (K2)

#### 1.3.4 Testmanagement in AUTOMOTIVE SPICE (75 Minuten)

LO-1.3.4.1 Sie kennen den Zweck der Testbarkeitsanalyse. (K1)

LO-1.3.4.2 Sie können den Unterschied zwischen statischen Analyseverfahren und dynamischen Testentwurfsverfahren erklären. Beispielsweise anhand der Entwicklung einer Teststrategie mit den Elementen Verifikationsstrategie und Verifikationskriterien mit Bezug zu AUTOMOTIVE SPICE. (K2)

LO-1.3.4.3 Sie können den Unterschied zwischen Blackbox- und Whitebox-Testverfahren in Bezug zu AUTOMOTIVE SPICE erklären. (K2)

LO-1.3.4.4 Sie können den Testdokumentationsbedarf (Testarbeitsprodukte) nach AUTOMOTIVE SPICE erklären und können mit den Arbeitsprodukten der Testprozesse gemäß IEEE 829 arbeiten. (K2)

LO-1.3.4.5 Sie können die Bestandteile einer Testspezifikation gem. IEEE 829 benennen und ihre Anwendung in einem Automotive E/E-Entwicklungsprojekt erklären. (K2)

#### 1.3.5 Test- und Verifikationsaktivitäten in den Engineering- und Supporting-Prozessen nach AUTOMOTIVE SPICE (60 Minuten)

LO-1.3.5.1 Sie kennen die AUTOMOTIVE SPICE Engineering- und Supporting-Prozesse und können ihre Relevanz bezüglich des Testens erklären. (K2)

### 1.3.1 Einführung Automotive SPICE und Zielsetzung (15 Minuten)

AUTOMOTIVE SPICE wurde ab 2001 durch die SPICE User Group und die AUTOSIG (engl. AUTOMOTIVE Special Interest Group) entwickelt. Dazu gehören die deutschen Automobilhersteller Audi, BMW, Daimler, Porsche und Volkswagen, die sich zu diesem Zweck in der Herstellerinitiative „Software“ zusammengeschlossen haben. 2005 wurde der branchenspezifische Standard AUTOMOTIVE SPICE durch die AUTOSIG Automotive veröffentlicht. Abgeleitet wurde die Norm aus der ISO 15504 für Softwareprozess-Assessments. Dieses verbindliche Verfahren wird zur objektiven Prozessbewertung mit daraus resultierender anschließender Prozessverbesserung auf Projekt- und Organisationsebene herangezogen.

AUTOMOTIVE SPICE beinhaltet:

- Process Reference Model (PRM) und
- Process Assessment Model (PAM)

AUTOMOTIVE SPICE besitzt grundsätzlich zwei Dimensionen:

- Prozess
- Reifegrad

### 1.3.2 Reifegraddimension (30 Minuten)

Die Reifegrade entsprechen den sechs Prozessreifegraden, wie sie in der ISO 15504 definiert sind:

- Stufe 0 (unvollständiger Prozess): nicht ausgeführt bzw. erfüllt seinen Zweck nicht
- Stufe 1 (durchgeführter Prozess): erfüllt seinen Prozesszweck
- Stufe 2 (gesteuerter Prozess): geplant, überwacht und angepasst; Arbeitsprodukte werden angemessen erstellt, gelenkt und gepflegt
- Stufe 3 (etablierter Prozess) wird unter Einsatz eines definierten Vorgehens umgesetzt
- Stufe 4 (vorhersagbarer Prozess): läuft innerhalb definierter Grenzen ab, um seine Prozessergebnisse zu erreichen
- Stufe 5 (optimierender Prozess): wird kontinuierlich verbessert, damit gegenwärtige und zukünftige Unternehmensziele erreicht werden

Zur Durchführung von Assessments steht ein ISO 15504-2 kompatibles Assessmentmodell zur Verfügung. Auf der Basis von AUTOMOTIVE SPICE können Assessments für Lieferanten von Steuergeräten durchgeführt werden.

### 1.3.3 Prozessdimension (30 Minuten)

Die Prozesse der Norm basieren auf der ISO 12207, die um automobilspezifische Ergänzungen erweitert bzw. angepasst wurden.

Im Rahmen eines AUTOMOTIVE SPICE Assessments wird die Reife jedes einzelnen Prozesses bewertet. Diese enthält für alle Prozesse Indikatoren zur Beurteilung, inwieweit die Prozesse durchgeführt werden. Die in AUTOMOTIVE SPICE verwendeten Prozesse werden in drei Gruppen aufgeteilt:

- primäre Prozesse im Lebenszyklus (Primary Life Cycle Processes),
- organisatorische Prozesse im Lebenszyklus (Organisational Life Cycle Processes) und
- unterstützende Prozesse im Lebenszyklus (Supporting Life Cycle Processes).

Die Prozesskategorie der primären Prozesse im Lebenszyklus enthält eine weitere Aufteilung in:

- (ACQ) Akquisition-Prozessgruppe
- (SPL) Supply-Prozessgruppe
- (ENG) Engineering-Prozessgruppe



Die Prozesskategorie der organisatorischen Prozesse im Lebenszyklus enthält eine weitere Aufteilung in:

- (MAN) Management-Prozessgruppe
- (PIM) Prozessverbesserungs-Prozessgruppe
- (REU) Re-Use-Prozessgruppe

Die Prozesskategorie der unterstützenden Prozesse im Lebenszyklus enthält keine weitere Aufteilung.

Die in AUTOMOTIVE SPICE relevanten Prozesse sind eine Teilmenge der ISO 15504 Prozesse inkl. einiger zusätzlicher Prozesse in der Prozessgruppe Akquisition. Als Teilmenge von AUTOMOTIVE SPICE wurde der HIS-Scope definiert, der eine Minimalanforderung der in der Herstellerinitiative Software (HIS) zusammengeschlossenen deutschen Automobilhersteller darstellt.

### 1.3.4 Testmanagement in AUTOMOTIVE SPICE (75 Minuten)

#### 1.3.4.1 Testbarkeitsanalyse (15 Minuten)

Die Testbarkeitsanalyse legt die Basis für den anforderungsbasierten Test der Software in einem softwarebestimmten Elektronikentwicklungsprojekt. Zweck der Testbarkeitsanalyse ist die Sicherstellung der Testbarkeit des jeweils aktuellen Softwarestands als Voraussetzung für die Spezifikation der anforderungsbasierten Testfälle.

#### 1.3.4.2 Anforderungen an die Testprozesse in AUTOMOTIVE SPICE (30 Minuten)

AUTOMOTIVE SPICE stellt im Rahmen der Entwicklung einer Teststrategie sowohl Anforderungen an die Umsetzung von statischen Tests (wie Reviews und statischer Analysen) als auch von dynamischen Tests (wie Blackbox- und Whitebox-Testentwurfverfahren).

Anforderungen an die durchzuführenden Testverfahren werden u. a. in Form von Verifikationskriterien definiert; d. h. Verifikationskriterien spezifizieren, was erfüllt sein muss, damit ein Testobjekt (Anforderung, Modul) als verifiziert gilt.

#### 1.3.4.3 Testdokumentation nach IEEE 829 (30 Minuten)

Auch bei der Testdokumentation greift AUTOMOTIVE SPICE auf etablierte Industriestandards wie die IEEE 829 zurück. So werden Vorgaben für folgende Arbeitsprodukte der IEEE 829 entnommen: Testkonzept, Mastertestkonzept, Stufentestkonzept, Testspezifikation (Testfallspezifikation, Testentwurfsspezifikation, Testablaufspezifikation), Testprotokoll, Fehler- und Abweichungsbericht, Teststatusbericht, Stufentestbericht und Mastertestbericht.

### 1.3.5 Test- und Verifikationsaktivitäten in den Engineering- und Supporting-Prozessen nach AUTOMOTIVE SPICE (60 Minuten)

Die testorientierten Engineering-Prozesse

- Softwareintegrationstest,
- Softwaretest,
- Systemintegrationstest und
- Systemtest

und die Supporting-Prozesse

- Qualitätssicherung und
- Reviews

in AUTOMOTIVE SPICE beinhalten sowohl Test- als auch Qualitätssicherungsaktivitäten.

Die oben aufgeführten testorientierten Prozesse haben eine ähnliche Struktur, die sowohl auf den fundamentalen Testprozess nach ISTQB (siehe Foundation Level Lehrplan) als auch auf eine Testergebnisdokumentation nach IEEE 829 zurückgeführt werden kann.

## 1.4 Vergleich AUTOMOTIVE SPICE und ISO 26262 (K2) – [25 Minuten]

### **Begriffe**

Teststrategie, Testkonzept

LO-1.4.1 Sie sind in der Lage, die unterschiedlichen Zielsetzungen der Normen AUTOMOTIVE SPICE und ISO 26262 zu erkennen und einem Vergleich gegenüberzustellen. (K2)

Die beiden zuvor behandelten Normen bauen nicht aufeinander auf und verfolgen unterschiedliche Zielsetzungen. Hierdurch gehen ihre Forderungen nicht nahtlos ineinander über. Ihr Einsatz und konkurrierende Aspekte müssen daher entsprechend der Teststrategie des Unternehmens und der jeweiligen Projektziele aufeinander abgestimmt werden. Hierzu müssen Stärken und Schwächen beider Normen gegenübergestellt und in der Teststrategie miteinander kombiniert werden.

## 1.5 Übungen/Verständnisfragen/Fallstudien (K3) - [20 Minuten]

In diesem Abschnitt sind kapitelübergreifende Übungen und Verständnisfragen zu behandeln.

## 2 Baustein 2: Testen im E/E- Entwicklungsprozess - [380 Minuten]

### 2.1 Softwareentwicklung und -test im Automotive-Umfeld (K2) - [150 Minuten]

#### Begriffe

Automotive PEP, Musterstand, Integrationsstufenkonzept, E/E-Entwicklungsprozess

#### 2.1.1 Grundlagen des Softwaretestens im Automobilbau (30 Minuten)

LO-2.1.1 Sie können die Rahmenbedingungen in einem Automotive E/E-Entwicklungsprojekt insbesondere hinsichtlich Variantenvielfalt, Stückzahleneffekte, hardwarenaher Programmierung und die hieraus resultierenden Problemstellungen bezüglich des Testens beschreiben. (K2)

#### 2.1.2 Automotive PEP (30 Minuten)

LO-2.1.2 Sie können den prinzipiellen Aufbau automobiler Produktentstehungsprozesse (PEP) beschreiben und kennen ihre Herausforderungen. (K2)

#### 2.1.3 Musterstände (30 Minuten)

LO-2.1.3 Sie können die Musterstände A-Muster, B-Muster und C-Muster unterscheiden und wissen, welche Anforderungen an den Test und die Qualitätssicherung in welcher Musterstufe gestellt werden. (K2)

#### 2.1.4 Einbindung der Softwareentwicklung in den automobilen Produktentstehungsprozess (30 Minuten)

LO-2.1.4 Sie können die Herausforderungen des Testens in einem E/E-Entwicklungsprojekt hinsichtlich der Kommunikationswege erläutern. (K2)

#### 2.1.5 Integrationsstufenkonzept (30 Minuten)

LO-2.1.5 Sie können bezüglich einer Softwareentwicklung im Rahmen eines E/E-Entwicklungsprojektes den Einsatz von Integrationsstufen bzgl. des Testens erläutern. (K2)

#### 2.1.1 Grundlagen des Softwaretestens im Automobilbau (30 Minuten)

Die Geschäftsprozesse in der automobilen Produktentwicklung sind weitgehend durch die Erstellung physischer Bauteile geprägt. Die Erstellung der eingebetteten Software wird aus Unternehmenssicht als nachgelagerter Befähiger gesehen, obwohl anerkanntermaßen ein Großteil der heutigen Funktionalität ausschließlich oder zu großen Teilen durch Software ermöglicht wird. Diese funktionserbringende Software wird als „eingebettete Software“ bezeichnet.

Die Herausforderung bei der Erstellung von eingebetteter Software liegt

- an der Zugänglichkeit,
- zum anderen an den physikalischen und betriebswirtschaftlichen Randbedingungen des einbettenden Systems (z. B. durch harte Echtzeitanforderungen und Stückzahleneffekte) und
- an der Herausforderung trotz Serienproduktion dem Kunden eine möglichst große Produktindividualisierung zu ermöglichen.

Aus diesem Grund ist eine hardwarenahe Programmierung unter Einsatz von Sprachen gefordert, die dieses erlauben.

## 2.1.2 Automotive PEP (30 Minuten)

Gemäß den Produktentstehungsprozessen der Automobilbauer müssen in kürzester Zeit komplexe Produkte unterschiedlicher Disziplinen entwickelt werden. Hierzu müssen mehrere Prozesse, aufeinander abgestimmt, punktgenau Ergebnisse liefern. Neben der Entwicklung gehören hierzu beispielsweise betriebswirtschaftliche Prozesse, die Logistik, die Produktionsplanung oder die Serviceplanung. Diese Prozesse besitzen synchrone Meilensteine.

In Anlehnung an [VDA1998] besteht ein automobiler Produktentstehungsprozess (PEP) grob vereinfacht aus folgenden Phasen:

- Planung: Projektidee
- Definition: Produktplanung, Konzeptdefinition
- Realisierung: Serienentwicklung, Konstruktion, Absicherung (Verifikation)
- Produktionsplanung: Produktionsvorbereitung, Übergabe Serienproduktion

## 2.1.3 Musterstände (30 Minuten)

Die Qualitätssicherung im Entwicklungsprozess des Produktentstehungsprozesses erfolgt über Musterstände (A-, B- und C-Muster).

- A-Muster weisen die Tauglichkeit des Konzeptes nach,
- B-Muster weisen die Serientauglichkeit nach und
- C-Muster sind mit Serienwerkzeugen gefertigte Musterteile, die die Produzierbarkeit beweisen.

Eine Freigabe ab B-Muster bewirkt im Allgemeinen betriebswirtschaftliche Konsequenzen, da mit dieser Freigabe Verantwortungen vom Lieferanten an den Auftraggeber übergehen.

## 2.1.4 Einbindung der Softwareentwicklung in den automobilen Produktentstehungsprozess (30 Minuten)

Die Softwareentwicklung muss sich in den automobilen Produktentstehungsprozess (PEP) einordnen, der aus der klassischen konstruktiven Sichtweise auf das Fahrzeug entstanden ist. Die Technologie und die Bedeutung aber auch die Komplexität aufgrund zunehmender funktionaler Vernetzung wachsen schneller, als dies in den vorherrschenden Prozessdefinitionen berücksichtigt werden kann. So wird die Softwareentwicklung häufig den konstruktiven Tätigkeiten nachgelagert eingeordnet. Die Anforderungen an die Softwareentwicklung resultieren daher aus Abhängigkeiten aus den konstruktiven Tätigkeiten und den Rahmenbedingungen, die der PEP vorgibt. Entsprechend findet die Qualitätssicherung auf unterschiedlichen Ebenen statt. Viele der Software zugehörige Qualitätsmerkmale werden erst in der Fahrzeugerprobung abgesichert. Hierdurch entstehen gerade bei der Fehleranalyse und -erhebung (Debuggen) komplexe Kommunikationswege. So kann z.B. ein beim OEM im Fahrzeugtest gefundener Fehler anfangs nicht klar zugeordnet werden, ob es sich um einen Systemfehler, Hardwarefehler, Softwarefehler oder sonstiges handelt. Das Problem wird dann als Abweichung aufgenommen und über einen "Problemlösungsprozess" bis zur Schließung des Abweichungsberichts verfolgt. Nach der ersten Analyse der Abweichung wird die Abweichung zur Bearbeitung einer Kategorie "Hardware", "Software", "System" oder sonstiges zugeordnet. Der dabei entstehende Workflow beschreibt die Kommunikationswege in der Entwicklung.

## 2.1.5 Integrationsstufenkonzept (30 Minuten)

Um der Funktionssichtweise der Softwareentwicklung Rechnung zu tragen, setzt sich als Best Practice das Integrationsstufenkonzept durch. Dieses Konzept sieht gemäß vordefinierten Integrationsstufen die stufenweise Integration fertiggestellter Funktionen vor. Entsprechend leiten sich aus den Integrationsstufen die entsprechenden Teststufen ab.

## 2.2 Testen in virtueller Umgebung (K3) – [120 Minuten]

### Begriffe

Virtuelle Umgebung, Hardware in the Loop (HiL), Software in the Loop (SiL)

#### 2.2.1 Einführung und Zielsetzung des Testens in virtueller Umgebung (45 Minuten)

LO-2.2.1.1 Sie können einen Test in der virtuellen Umgebung spezifizieren und die Durchführung skizzieren. (K3)

#### 2.2.2 Hardware in the Loop (45 Minuten)

LO-2.2.2.1 Sie kennen Ziel und Zweck einer Hardware in the Loop-Simulation. (K1)

LO-2.2.2.2 Sie kennen die Vorteile von Hardware in the Loop und können diese stichpunktartig beschreiben. (K2)

LO-2.2.2.3 Sie wissen, aus welchen Komponenten der HiL-Prüfstand besteht, und kennen die Herausforderungen bei der Gewährleistung des reibungsfreien Zusammenspiels der Komponenten. (K2)

LO-2.2.2.4 Sie kennen den Anwendungsbereich von Hardware in the Loop (HiL), können die für HiL-Test spezifischen Testaktivitäten erläutern und können den Aufwand beim Vorgehen abschätzen. (K3)

#### 2.2.3 Software in the Loop (30 Minuten)

LO-2.2.3.1 Sie kennen die Ziele von Software in the Loop (SiL) und können die wesentlichen Unterschiede zum HiL charakterisieren. (K2)

LO-2.2.3.2 Sie können die wichtigsten Randbedingungen von Software in the Loop (SiL) benennen. (K1)

#### 2.2.1 Einführung und Zielsetzung des Testens in virtueller Umgebung (45 Minuten)

##### 2.2.1.1 Einführung

Software im Automobil ist immer abhängig von der einbettenden Hardware. Mit der Erstellung und Absicherung der Software kann jedoch die Fertigstellung der Hardware nicht abgewartet werden. Entsprechend sind virtuelle Umgebungen zu schaffen, die die reale Umgebung simulieren.

##### 2.2.1.2 Zielsetzung des Testens in virtueller Umgebung

Um auf fehlende Hardware bzw. fehlende Zielsysteme verzichten zu können, sind Methoden und Verfahren entwickelt worden, die eine Simulation der Umgebung zulassen. Hierunter fallen in erster Linie HiL und SiL, aber auch MiL.

Hinweis: Die in virtueller Umgebung durchgeführten Tests folgen dem bereits im Foundation Level eingeführten Testprozess mit entsprechender Testdokumentation gemäß IEEE 829.

Es hat sich somit eine Reihe von Testverfahren etabliert, bei denen der „Prüfling“ in einer virtuellen Umgebung betrieben wird, die die Realität (z. B. ein Fahrzeug) durch eine Simulation möglichst exakt nachbildet. Wenn sich hierbei ein geschlossener Regelkreis ergibt, so handelt es sich um ein „Closed loop“- bzw. „In the Loop“-Verfahren (vgl. [Baum2006]).

#### 2.2.2 Hardware in the Loop (45 Minuten)

##### 2.2.2.1 Zielsetzung HiL

Das Simulationsverfahren „Hardware in the Loop“ (HiL) bezeichnet den Test realer Komponenten aus dem Gesamtsystem, herausgelöst in virtueller Umgebung.

### 2.2.2.2 Vorteile und Risiken einer HiL-Simulation

Folgende Vorteile können durch eine HiL-Simulation erzielt werden:

- Simulation vor Verfügbarkeit des Fahrzeugs
- reproduzierbare Testfälle
- einfache Modifikation der Systemstruktur
- gefahrlose und zerstörungsfreie Erforschung extremer Zustände
- automatisierter Testablauf
- Unabhängigkeit von Randbedingungen, wie z. B. Umwelteinflüsse

Die Übertragbarkeit der Ergebnisse aus der Simulation in den realen Versuch ist aufgrund reduzierter Modelle zugunsten der Echtzeitfähigkeit nicht in jeder Hinsicht gegeben.

### 2.2.2.3 Voraussetzung zur HiL-Simulation

Die Einsatzfähigkeit im Sinne der Verfügbarkeit des Testobjekts inkl. Hardware ist Grundvoraussetzung für die Durchführung aussagekräftiger HiL-Simulationen. Dementsprechend muss im Entwicklungs- und Organisationsprozess berücksichtigt werden, dass Simulationshardware zum realitätsnahen Ersatz der Umgebung mit den elektrischen Schnittstellen zur Verfügung steht.

### 2.2.2.4 Aufbau eines Versuchstands

Ein HiL-Prüfstand besteht aus folgenden Komponenten:

- Hardware inkl. Steuerung (Testobjekt)
- echtzeitfähige Simulationsumgebung
- Ausgabe- und Analyseeinheit

Diese Komponenten zeitnah zur Verfügung zu stellen, stellt die Herausforderung von HiL-Simulationen dar.

### 2.2.2.5 Ablauf der HiL-Simulation

Während der HiL-Simulation wird ein reales Steuergerät an einer Modellierung seiner späteren Umgebung angeschlossen und in dieser Art und Weise getestet. Damit ist eine frühe Analyse des entwickelten Gerätes möglich. Dabei wird das komplett integrierte System, bestehend aus Hardware und Software, mit einer Simulation der Umgebung mittels Signalflüssen gekoppelt und möglichst unter Echtzeitbedingungen ausgeführt. Der HiL-Test selbst wird in der Regel als Blackbox-Test durchgeführt.

## 2.2.3 Software in the Loop (30 Minuten)

### 2.2.3.1 Zielsetzung SiL

Im Vergleich zu HiL erfordert SiL keine Spezialhardware und bietet dadurch eine höhere Flexibilität für die Ausführung der Tests. Die SiL-Simulation lässt sich somit frühzeitig während der Softwareentwicklung einsetzen und bietet die Möglichkeit, Tests bereits vor der Fertigstellung der Hardware auszuführen.

### 2.2.3.2 Randbedingungen: Einsatzzeitpunkt und Aussagekraft

Das vollständige Abbild der einbettenden Hardware in ein Simulationsmodell kann zu einem aufwendigen Entwicklungsprozess mit entsprechender Absicherung führen.

Das Zeitverhalten der Software in der SiL-Simulation weicht oftmals vom Zeitverhalten auf der realen Hardware ab. Dies lässt sich durch Synchronisationsverfahren mit einer simulierten Echtzeituhr kompensieren, was jedoch die Komplexität der Simulation erhöht.

## 2.2.4 Abgrenzung zum Open-Loop-Test [informativ]

Die Begriffe Open-Loop-Test und Closed-Loop-Test werden in Analogie zur Regelungstechnik definiert. Beim Open-Loop-Verfahren werden die Eingänge des Steuergeräts unmittelbar mit Testvektoren (Stimuli) beaufschlagt. Die Ausgangssignale werden erfasst und mit den von der Testreferenz erzeugten Sollwerten verglichen. Im Gegensatz dazu werden beim Closed-Loop-Verfahren die ECU-Eingangssignale nicht unmittelbar vom Stimuli-Generator des Testsystems erzeugt, sondern von einem Simulationsmodell der Regelstrecke. Diesem Modell werden sowohl die vom Testsystem generierten Stimuli als auch die Ausgangsgrößen der ECU zugeführt. Durch die Rückführung ergibt sich ein geschlossener Regelkreis (vgl. [Baum2006]).

## 2.3 Besonderheiten in der automobilen Softwareentwicklung (K2) - [60 Minuten]

### Begriffe

AUTOSAR, Betriebsmodi, Variantenvielfalt

Die automobilen Softwareentwicklung unterliegt weiteren Besonderheiten, die hier lediglich als Ausblick erwähnt werden können. Diese reichen von neuen Initiativen zur Abstraktion der Funktion von der Hardware über verschiedene Betriebszustände, die nicht im primären Betrachtungsspektrum liegen bis hin zur Herausforderung, beliebig viele Varianten zu managen.

### 2.3.1 AUTOSAR (30 Minuten)

LO-2.3.1.1 Sie kennen die Zielsetzung von AUTOSAR allgemein. (K1)

LO-2.3.1.2 Sie können die Besonderheiten des Testens von AUTOSAR-basierten Systemen erklären. (K2) [informativ]

### 2.3.2 Betriebsmodi (10 Minuten)

LO-2.3.2.1 Sie können die Auswirkungen der Betriebsmodi auf das Testen erklären. (K2)

### 2.3.3 Variantenvielfalt (20 Minuten)

LO-2.3.3.1 Sie kennen die mit der Variantenvielfalt verbundenen Probleme bezüglich des Testens. (K1)

### 2.3.1 AUTOSAR (30 Minuten)

AUTOSAR (Automotive Open System ARchitecture) ist eine Initiative mehrerer Fahrzeughersteller zur Abstraktion der Software von der Hardware. Durch AUTOSAR erhalten Steuergeräte ein Betriebssystem mitsamt virtuellem Kommunikationsbus. Zur Initiative gehören darüber hinaus weitere Anforderungen mit Auswirkungen auf den Testprozess.

### 2.3.2 Betriebsmodi (10 Minuten)

Üblicherweise haben Automotive-Systeme verschiedene Betriebsmodi (z. B. im Motorelektronik-Steuergerät: init, predrive, drive, postdrive, diagnose). Weitere Betriebsmodi sind beispielsweise der Produktions- oder der Transportmodus. Die entsprechenden Funktionen und die Übergänge zwischen den Betriebsmodi müssen im Test berücksichtigt werden.

### 2.3.3 Variantenvielfalt (20 Minuten)

Die Kundenindividualisierbarkeit der Fahrzeuge wirkt sich ebenfalls auf die Variationsmöglichkeiten der Software aus. Es resultiert eine nahezu unendlich große Variantenvielfalt, in der nur ein Bruchteil der tatsächlichen Variationen getestet werden kann.

## 2.4 Übungen/Verständnisfragen/Fallstudien (K3) - [30 Minuten]

In diesem Abschnitt sind kapitelübergreifende Übungen und Verständnisfragen zu behandeln.



### 3 Literaturverzeichnis

- [ASPICE\_2010A] Automotive SIG (ED.): AUTOMOTIVE SPICE, Prozess Assessment Model, V2.5, 2010.
- [ASPICE\_2010B] Automotive SIG (ED.): AUTOMOTIVE SPICE, Prozess Reference Model, V4.5, 2010.
- [Baum2006] Baumann, Gerd: „Was verstehen wir unter Test? Abstraktionsebenen, Begriffe und Definitionen“, 1. Autotest-Fachkonferenz, Stuttgart 2006.  
URL: [http://www.fkfs.de/uploads/publikationen/Baumann\\_Autotest\\_2006\\_Test\\_Einfuehrung\\_Paper.pdf](http://www.fkfs.de/uploads/publikationen/Baumann_Autotest_2006_Test_Einfuehrung_Paper.pdf).
- [Borg2014] Borgeest, Kai: Elektronik in der Fahrzeugtechnik. ATZ/MTZ-Fachbuch, 2014.
- [BMW95014] BMW Group Standard: Embedded Software Entwicklung, 2007.
- [DIN61508] DIN 61508: Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme; 2010.
- [HOEP2013a] Hoepke, E.; Breuer, S. (Hrsg.): Nutzfahrzeugtechnik, ATZ/MTZ-Fachbuch, Vieweg+Teubner Verlag | Springer Fachmedien, Wiesbaden 2013.
- [IEEE 829-2008] Standard for Software and System Test Documentation, 2008.
- [ISO 12207] ISO/IEC 12207 „Systems and software engineering - Software life cycle processes“, 2008.
- [ISO 15504] ISO/IEC 15504 „Information technology - Process assessment“, 2003-2013.
- [ISO 26262-2011] ISO 26262 Roadvehicles Functional Safety, Part 1 - 10, 2011/2012.
- [ISO 29119-2013] ISO/IEC/IEEE 29119-1, IS 09/2013; ISO/IEC/IEEE 29119-2, IS 09/2013; ISO/IEC/IEEE 29119-3, IS 09/2013; ISO 29119-4 – DIS.
- [ISTQB®2011] ISTQB® Foundation Level Lehrplan 2011.  
URL: <http://www.german-testing-board.info/>.
- [ISTQB®2013] ISTQB® Foundation Level Syllabus Overview 2011, 2013, V0.3.  
URL: <http://www.istqb.org/downloads/finish/16/136.html>.
- [ISTQB®2012a] ISTQB® Advanced Level 2012 Overview (deutschsprachig).  
URL: <http://www.german-testing-board.info>.
- [ISTQB®2012b] ISTQB® Advanced Level 2012 Testmanager Syllabus (deutschsprachig).  
URL: <http://www.german-testing-board.info>.
- [ISTQB®2012c] ISTQB® Advanced Level 2012 Test Analyst Syllabus (deutschsprachig).  
URL: <http://www.german-testing-board.info>.
- [ISTQB®2012d] ISTQB® Advanced Level 2012 Technical Test Analyst Syllabus (deutschsprachig).  
URL: <http://www.german-testing-board.info>.
- [ISTQB®2012e] ISTQB® Glossary 2.3, deutsch-englischsprachige Ausgabe; 2014.  
URL: <http://www.german-testing-board.info>.
- [Kind2009] Kindel, Olaf; Friedrich, Mario: Softwareentwicklung mit AUTOSAR. Grundlagen, Engineering, Management für die Praxis, dpunkt.verlag, 2009.
- [MISRA2012] MISRA: The Motor Industry Software Reliability Association; 2012.  
URL: <http://www.misra.org.uk>.

- [Muell2007] Müller, Markus; Hörmann, Klaus; Dittmann, Lars; Zimmer, Jörg: AUTOMOTIVE SPICE in der Praxis 1. Auflage, dpunkt Verlag Heidelberg, 2007.
- [Noere2012] Nörenberg, Ralf: Effizienter Regressionstest von E/E-Systemen nach ISO 26262, KIT Scientific Publishing, 2012.
- [Reif2011] Reif, K. (Hrsg.): Bosch Autoelektrik und Autoelektronik, Springer Fachmedien, Wiesbaden GmbH, 2011.
- [Saul2011] Sauler, Jürgen; Kriso, Stefan: „ISO 26262 - Die zukünftige Norm zur funktionalen Sicherheit von Straßenfahrzeugen“; elektronik-praxis; 2011.
- [SPILL12a] Spillner, Andreas; Linz, Tilo: Basiswissen Softwaretest, dpunkt Verlag, Sept. 2012.
- [VDA1998] VDA-Empfehlung 4, Teil 3 „Qualitätsmanagement in der Automobilindustrie, Sicherung der Qualität vor Serieneinsatz - Projektplanung“, 1. Aufl. 1998.
- [VDA2004] VDA Band 13: Entwicklung softwarebestimmter Systeme, Forderungen an Prozesse und Produkte, Systematische Softwareentwicklung, Entwicklungsmodell, Vorgehensweisen, Praktiken, Systempartnerschaft, 1. Auflage 2004 (Rotdruck).
- [VDA2008] Automotive SPICE Prozessassessmentmodell, VDA, 1. Aufl., Nov. 2008.
- [VDA2013] VDA Qualitätsmanagement Center: AUTOMOTIVE SPICE, 2013. URL: <http://vda-qmc.de/software-prozesse/AUTOMOTIVE-spice/>.
- [Winter1999] Winter, Mario: Qualitätssicherung für objektorientierte Software - Anforderungsermittlung und Test gegen die Anforderungsspezifikation, 1999. URL: <ftp://ftp.fernuni-hagen.de/pub/fachb/inf/pri3/papers/winter/diss.pdf>.

Zeitschriften mit Fachartikeln zum „Testen von E/E-Systemen“ (im deutschsprachigen Raum):

- ATZ-elektronik: <http://www.springerprofessional.de/servlet/segment/springer-professional/35658---atzelektronik/2845040.html>.
- Hanser Automotive: <http://www.hanser-automotive.de/index.php?id=heftarchiv>.
- Automobil Elektronik: <http://www.automobil-elektronik.de>.

## 4 Anhang A – Glossar und Abkürzungsverzeichnis

### 4.1 Glossar CTFL-EXTENSION “Certified Automotive Software Tester (CTFL®-CAST)”

Die folgenden Begriffe sind als komplementär zum ISTQB® Glossar ab V2.2 zu verstehen.

Begriff	Bedeutung
<b>Absicherung</b>	Absicherung ist eine systematische Prüfung, ob die Arbeitsprodukte aus einer bestimmten Entwurfsphase der Softwareentwicklung den zuvor aufgestellten Anforderungen bezüglich Richtigkeit, Vollständigkeit und Konsistenz genügen (nach [BMW95014]).
<b>Analyseeinheit HiL</b>	Eine Analyseeinheit im HiL-Stand verarbeitet die Eingangssignale und berechnet die Zwischenwerte bzw. Ausgangssignale anhand von vorhandenen Informationen.
<b>Anforderungsbasierter Test</b>	Ein anforderungsbasierter Test ist ein Test, der auf den Anforderungen basiert. Beim anforderungsbasierten Testen werden die freigegebenen Anforderungen als Testbasis herangezogen. Zu jeder Anforderung wird mindestens ein Testfall abgeleitet und in der Testspezifikation dokumentiert. In der Regel wird mehr als ein Testfall benötigt, um eine Anforderung zu testen [SPILL12a].
<b>AUTOMOTIVE Safety Integrity Level (ASIL)</b>	1. Methodik zur Einstufung von sicherheitskritischen Systemen in Sicherheitsstufen im Automotive. 2. (AUTOMOTIVE) Sicherheitsintegritätslevel = diskrete Stufen zur Spezifikation der Anforderung für die Sicherheitsintegrität der Sicherheitsfunktionen.
<b>AUTOMOTIVE SPICE</b>	AUTOMOTIVE SPICE ist eine domänenspezifische Variante des internationalen Standards ISO/IEC 15504 (SPICE). Der Zweck von AUTOMOTIVE SPICE ist die Bewertung der Leistungsfähigkeit der Entwicklungsprozesse von Steuergerätlieferanten in der Automobilindustrie. Anmerkung: Für die Durchführung von konformen AUTOMOTIVE SPICE Assessments gelten die Anforderungen aus ISO/IEC 15504-2.
<b>Assessment</b>	Erfassung von Merkmalen der Leistungsfähigkeit und Prozesse einer Organisation gegenüber einem Modell mit dem Ziel, Prozesse bzw. Prozessfähigkeit zu bewerten und zu verbessern.
<b>Ein-/Ausgabeeinheit HiL</b>	Die Ein-/Ausgabeeinheit stellt die Verbindung zur Außenwelt in Form des Austausches von Daten mit der virtuellen Umgebung des HiLs her. Die Signale zwischen reeller Spezialhardware (Simulationshardware) und Rechneinheit werden mittels Ein-/Ausgabeeinheit

	ausgetauscht.
<b>AUTOSAR</b>	AUTOSAR (AUTOMOTIVE Open System ARchitecture) ist ein internationaler Standard der Automobilindustrie. Er beschreibt eine offene und standardisierte Softwarearchitektur für die Fahrzeugentwicklung, die gemeinsam von Automobilherstellern, Automobilzulieferern und Werkzeugherstellern entwickelt und getragen wird.
<b>Automobiler Produktentstehungsprozess (Automotive-PEP)</b>	Der automobiler Produktentstehungsprozess beschreibt die Arbeitsabläufe von der Idee für ein neues Fahrzeug bis zu dessen Entwicklung, Produktion und Verkauf.
<b>Betriebsmodus (Automotive)</b>	Betriebsmodus definiert ein spezielles Verhalten, welches für eine bestimmte Aufgabe (z. B. Transport oder Produktion) optimiert ist.
<b>E/E-Entwicklungsprozess</b>	Elektrischer und/oder Elektronischer Entwicklungsprozess.
<b>E/E-System</b>	Elektrisch/Elektronisches System: Ein E/E-System besteht generell aus einer vernetzten Menge weitgehend autonomer, aus Hardware- und Softwareanteilen bestehenden Steuergeräten, welche als Verbund die Funktionalität bereitstellen [Noere2012, S. 5].
<b>Echtzeitfähigkeit</b>	Echtzeitfähigkeit eines Systems besagt, dass ein System auf ein Ereignis innerhalb eines vorgegebenen Zeitrahmens reagieren muss.
<b>Elektrik</b>	Unter Elektrik verstehen wir hier die Gesamtheit aller elektrischen Funktionen und der zugehörigen Komponenten eines Fahrzeuges [HOEP2013a].
<b>Elektronik</b>	Elektrische Funktionen, die aus technischen oder wirtschaftlichen Gründen statt in konventioneller Elektrotechnik mit Hilfe elektronischer Schaltungen oder Mikroprozessortechnik und Software realisiert werden [HOEP2013a].
<b>Erprobung</b>	Erprobung ist eine Methode, bei der das Testobjekt im Hinblick auf die in der Entwicklung bzw. Produktion zugrunde liegenden Zielvorstellungen überprüft wird. Der Produktentstehungsprozess muss entsprechend fortgeschritten sein.
<b>Freigabe</b>	Freigabe ist eine Formale Entscheidung auf der Basis der Projektergebnisse über den Reifegrad eines Produktes/Objektes zur offiziellen Übergabe an den nächsten Prozessschritt [VDA2004].
<b>FTA (Fault Tree Analysis, Fehlerbaumanalyse)</b>	Die Fehlerbaumanalyse (FTA) ist eine Methode der Zuverlässigkeits- und Sicherheitsanalyse. Das Ziel ist, mögliche Kombinationen von Ursachen zu bestimmen, die zu bestimmten unerwünschten Ereignissen führen können. Bei der Durchführung von FTA wird eine grafisch-logische Baumstruktur zum Verständnis der Zusammenhänge erstellt.
<b>Funktionale Sicherheit</b>	Funktionale Sicherheit ist der Teil der Sicherheit, der

	davon abhängig ist, dass ein elektrisch bzw. elektronisch geregeltes System korrekte Reaktionen auf seine Eingänge liefert.
<b>Hardware in the Loop (HiL)</b>	<ol style="list-style-type: none"> <li>1) HiL ist ein Simulationsverfahren, bei dem ein reales elektronisches Steuergerät oder eine mechatronische Komponente über seine Ein- und Ausgänge in einer virtuellen Umgebung des Systems ausgeführt wird.</li> <li>2) Bei Hardware in the Loop (HiL) wird das vollständige Steuergerät über die I/O-Schnittstellen getestet (nach [Reif2011]).</li> <li>3) Echte Hardware wird in einer simulierten Umgebung getestet (nach [VDA2008]).</li> </ol>
<b>Integrationsstufe (I-Stufen)</b>	<ol style="list-style-type: none"> <li>1) Eine Integrationsstufe ist ein fertig entwickeltes, erprobungsfähiges Funktionspaket.</li> <li>2) Stufenweise Planung der Integration von Softwarefunktionen.</li> <li>3) Integrationsstufen sind feste Termine, zu denen vorher definierte und mit allen Fachbereichen abgestimmte Funktionen im Gesamtfahrzeug zur Verfügung stehen (nach Volkswagen).</li> </ol>
<b>Integrationsstufenkonzept</b>	Das Integrationsstufenkonzept sieht gemäß vordefinierten Integrationsstufen die stufenweise Integration fertiggestellter Funktionen vor.
<b>MiL</b>	<ol style="list-style-type: none"> <li>1) Bei Model in the Loop (MiL) wird das Funktionsmodell der Software getestet. Das Funktionsmodell läuft auf einem Entwicklungsrechner [Reif2011].</li> <li>2) Das Simulationsmodell des Systems wird in einer simulierten Umgebung dynamisch getestet (nach [VDA2008]).</li> </ol>
<b>MISRA-C</b>	MISRA-C ist ein C-Programmierstandard aus der Automobilindustrie, der von der MISRA (Motor Industry Software Reliability Association) erarbeitet wurde.
<b>Musterstand</b>	<ol style="list-style-type: none"> <li>1) Definition des Konstruktions-, Software- und Hardwarestands eines Prototyps.</li> <li>2) Die Qualitätssicherung im Entwicklungsprozess des Produktentstehungsprozesses erfolgt über Musterstände (A-, B-, C-, D-Muster).</li> </ol>
<b>NPLF Bewertungsskala</b>	4-stufige Skala zur Bewertung der Prozessattribute nach ISO 15504-2.
<b>Produktentstehung</b>	Prozess, der alle Tätigkeiten von der ersten Produktidee bis zur Herstellung umfasst.
<b>Prozessdimension</b>	Eine Prozessdimension enthält für alle relevanten Prozesse die Indikatoren bezogen auf den Prozesszweck und die Prozessergebnisse. Die Prozesse sind in Prozessgruppen zusammengefasst [Muell2007].
<b>Reifegrad</b>	Der Reifegrad ist die Bewertung der Prozessfähigkeit einer

	Organisation.
<b>Reifegraddimension</b>	Die Reifegraddimension entspricht den sechs Prozessreifegraden, wie sie in der ISO 15504 definiert sind.
<b>Reifegradstufe</b>	Eine Reifegradstufe besteht aus einer Menge von Prozessattributen, die im Zusammenspiel eine deutliche Verbesserung der Fähigkeit zur Durchführung eines Prozesses vorsehen (vgl. [ISO 15504]).
<b>Software in the Loop (SiL)</b>	<ol style="list-style-type: none"> <li>1) SiL ist ein Closed-Loop-Verfahren, bei dem ECU-Software im Labor über Ein- und Ausgänge an eine virtuelle Umgebung angeschlossen wird.</li> <li>2) Bei Software in the Loop (SiL) wird der Softwarecode getestet. Er läuft auf einem Entwicklungsrechner (nach [Reif2011]).</li> <li>3) Echte Software wird in einer simulierten Umgebung getestet.</li> </ol>
<b>Sicherheit</b>	Sicherheit ist die Freiheit von unververtretbaren Risiken der Verletzung oder Schädigung der Gesundheit von Menschen, entweder direkt oder indirekt als ein Ergebnis von Schäden an Gütern oder der Umwelt.
<b>Sicherheitsanforderung (nach ISO)</b>	Anforderung, die verlangt, dass ein gefährlicher Zustand durch Maßnahmen vermieden werden soll (engl. safety requirement).
<b>Sicherheitsnachweis (synonym: Safety Case)</b>	<ol style="list-style-type: none"> <li>1) Liste aller Arbeitsprodukte und die Arbeitsprodukte selbst des Sicherheitsprozesses, die nachweisen, dass die Sicherheitsanforderungen vollständig sind und erfüllt sind. Liefert eine klare verständliche und nachvollziehbare Argumentation, warum ein System akzeptabel sicher ist, um in einem gegebenen Umfeld zu arbeiten.</li> <li>2) Ziel des Sicherheitsnachweises ist es, zusammenfassend und nachvollziehbar darzustellen, dass das System so entwickelt wurde, dass es frei von nicht akzeptablen Risiken ist.</li> </ol>
<b>Sicherheitsziel (nach ISO)</b>	Spezielle Sicherheitsanforderung auf oberster Ebene (Fahrzeug).
<b>Sicherheitsfunktion (nach IEC)</b>	Funktion eines Systems, die zur Risikominderung ausgeführt wird, mit dem Ziel, unter Berücksichtigung eines festgelegten gefährlichen Vorfalls einen sicheren Zustand für das System selbst zu erreichen oder aufrecht zu halten (DIN IEC 61508).
<b>Sicherheitsintegrität</b>	Wahrscheinlichkeit, dass ein sicherheitsbezogenes System die geforderten Sicherheitsfunktionen unter allen festgelegten Bedingungen innerhalb eines festgelegten Zeitraums anforderungsgemäß ausführt.
<b>System</b>	Ein System besteht aus Elementen mit gegenseitiger Wechselwirkung. Das Gesamtsystem besteht aus Sub-

	bzw. Teilsystemen, die wiederum aus Komponenten bestehen.
<b>Systemstruktur</b>	Die Systemstruktur ergibt sich aus den Grenzen des Systems, den Beziehungen zwischen den Elementen und den Wechselbeziehungen der Systemelemente zur Umwelt.
<b>Testbarkeitsanalyse</b>	Zweck der Testbarkeitsanalyse in einem E/E-Entwicklungsprojekt ist die Sicherstellung der Testbarkeit des jeweils aktuellen Softwarestands als Voraussetzung für die Spezifikation der anforderungsbasierten Testfälle.
<b>Traceability</b>	Traceability ist der Grad, bis zu dem eine Beziehung zwischen mindestens zwei Produkten des Entwicklungsprozesses ermittelt werden kann, insbesondere bei Produkten, die in einer Vorgänger-Nachfolger-Beziehung oder in einer Beziehung von Überordnung/Unterordnung zueinander stehen [ASPICE_2010A].
<b>Verifikationskriterium</b>	<ol style="list-style-type: none"> <li>1) Spezifikation, was erfüllt sein muss, damit ein Prüf- bzw. Testobjekt (z. B. Anforderung, Unit) als erfolgreich verifiziert gilt. Die Verifikation kann z. B. durch ein Review oder durch einen Test erfolgen. Beispiel für ein Verifikationskriterium: 100%-MISRA-2004-Compliance.</li> <li>2) Die Verifikationskriterien definieren die qualitativen und quantitativen Kriterien für die Verifikation einer Anforderung. Die Verifikationskriterien zeigen, dass eine Anforderung innerhalb vereinbarter Restriktionen verifiziert werden kann[ASPICE_2010A].</li> </ol>
<b>Virtuelle Umgebung</b>	Eine virtuelle Umgebung ist eine gedachte, von einem Computer simulierte Umgebung zur Nachbildung realer Welt.

## 4.2 Abkürzungen

Abkürzung	Bedeutung
ABS	Antiblockiersystem
ASIL	<b>AUTOMOTIVE Safety Integrity Level</b>
ASPICE	<b>AUTOMOTIVE Software Process Improvement and Capability Determination</b>
BP	<b>BasisPraktik</b> (in Automotive SPICE)
CCB	<b>Change Control Board</b>
ECU	<b>Electronic Control Unit</b>
E/E	<b>Elektrisch und/oder Elektronisch</b>
ENG	<b>Engineering-Prozessgruppe</b>
HAZOP	<b>HAZard and OPerability Study</b>
HiL	<b>Hardware in the Loop</b>
HIS	<b>Herstellerinitiative Software</b>
IEEE	Institute of <b>Electrical and Electronics Engineers</b>
GP	<b>Generische Praktik</b> (in AUTOMOTIVE SPICE)
IEC	<b>International Electrotechnical Commission</b>
ISO	<b>International Organization for Standardization</b>
ISTQB®	<b>International Software Testing Qualifications Board</b>
MAN	Prozessdomäne <b>MAN</b> agement im AUTOMOTIVE SPICE
MiL	<b>Model in the Loop</b>
MISRA	<b>Motor Industry Software Reliability Association</b>
OEM	<b>Original Equipment Manufacturer</b>
PEP	<b>ProduktEntstehungsProzess</b>
SPICE	<b>Software Process Improvement and Capability Determination</b>
SiL	<b>Software in the Loop</b>
SUP	Supporting-Prozessgruppe
VDA	<b>Verband der Automobilindustrie</b>



## 5 Anhang B – Hintergrundinformation zum Lehrplan

### 5.1 Zur Geschichte dieses Dokuments

Der Lehrplan 1.0 wurde 2010/2011 im Auftrag des Global Association for Software Quality AISBL (gasq) von Dr. Hendrik Dettmering entwickelt.

Zum Review des Dokuments wurden ausgewählte Experten deutscher OEMs berufen, durch die die Qualität und die Zielsetzung des Lehrplans geprüft und für geeignet bewertet wurden. Damit stellt dieses Dokument den Lehrplan für die Zertifizierung zum Automotive Software Tester dar und ist gleichermaßen die Basis für Schulungsunterlagen als auch für Prüfungsfragen zur Zertifizierung.

Mit dem 01.01.2014 übernimmt die Arbeitsgruppe „Automotive Testen (CAST)“ des German Testing Board (GTB) die Weiterentwicklung des Lehrplans, um der schnellen Entwicklung der Thematik folgen zu können und dem Bedürfnis der Industrie zu entsprechen, neben dem branchenunabhängigen CORE-Lehrplan auch die automotivspezifischen Aspekte als Ergänzung zum bewährten ISTQB Foundation Level zur Verfügung zu haben.

### 5.2 Voraussetzungen für die Ergänzung (Extension) zur Basisstufe

Voraussetzung für die Ergänzungsprüfung zum Certified Automotive Software Tester (CTFL®-CAST) ist das erworbene Zertifikat zum ISTQB Certified Tester Foundation Level (CTFL®) und das Interesse der Kandidaten am Testen im Rahmen von AUTOMOTIVE E/E-Entwicklungsprojekten. Es empfiehlt sich für die Kandidaten,

- zumindest ein minimales Hintergrundwissen im Bereich Softwareentwicklung oder Softwaretest zu haben (z. B. sechs Monate Erfahrung als System- oder Abnahmetester oder als Entwickler)
- oder einen Kurs besucht zu haben, der nach dem ISTQB®-Standard (durch ein ISTQB®-Mitglieds-Board) akkreditiert ist und/oder
- erste Erfahrungen im Testen in E/E-Entwicklungsprojekten in der Automotive-Branche gesammelt zu haben.

## 6 Anhang C – Lernziele/Kognitive Ebenen des Wissens<sup>6</sup>

Die folgende Taxonomie für Lernziele bildet die Grundlage des Lehrplans. Jeder Inhalt wird entsprechend den zugeordneten Lernzielen geprüft.

### 6.1 Taxonomiestufe 1: Kennen (K1)

Der Lernende ruft im Gedächtnis gespeicherte Informationen (z. B. Begriffe, isolierte Fakten, Abfolgen, Prinzipien, Mittel und Wege) ab. Typische beobachtbare Leistungen sind „erkennen“, „nennen“, „bezeichnen“.

**Schlüsselworte:**

sich erinnern, erkennen, wiedergeben, kennen

**Beispiel:**

Erkennt die Definition von Fehlerwirkung (engl. „Failure“) als

- „Nicht-Erfüllen einer definierten Leistung gegenüber einem Anwender oder sonstigen Stakeholder“ oder
- „die tatsächliche Abweichung einer Komponente oder eines Systems von der erwarteten bzw. vereinbarten Lieferung, einer Dienstleistung oder einem Ergebnis“.

### 6.2 Taxonomiestufe 2: Verstehen (K2)

Der Lernende begründet oder erläutert Aussagen zum Thema. Typische beobachtbare Leistungen sind „beschreiben“, „zusammenfassen“, „vergleichen“, „klassifizieren“, „begründen“, „erklären“, „Beispiele für Testkonzepte nennen“.

**Schlüsselworte:**

zusammenfassen, verallgemeinern, abstrahieren, klassifizieren, vergleichen, auf etwas übertragen, etwas gegenüberstellen, erläutern, interpretieren, übersetzen, darstellen, rückschließen, folgern, kategorisieren, Modelle konstruieren, erklären, Beispiele geben, begründen, verstehen

**Beispiel:**

Beschreibt Gemeinsamkeiten und Unterschiede zwischen Integrations- und Systemtest:

- Gemeinsamkeiten: Mehr als eine Komponente wird getestet und kann nicht-funktionale Aspekte umfassen.
- Unterschiede: Der Integrationstest konzentriert sich auf Schnittstellen zwischen und Interaktion von Komponenten; der Systemtest ist auf den Aspekten des ganzen Systems und End-to-End-Verarbeitung ausgerichtet.

Kann erklären, warum Tests so früh wie möglich entworfen werden sollen:

- Fehlerzustände finden, wenn die Behebung billiger ist
- die wichtigsten Fehlerzustände zuerst finden

---

<sup>6</sup> Übernommen aus: ISTQB Certified Tester Foundation Level Lehrplan, V1.01, deutschsprachige Ausgabe vom 19.04.2013.

### 6.3 Taxonomiestufe 3: Anwenden (K3)

Der Lernende überträgt erworbenes Wissen auf gegebene neue Situationen oder wendet sie zur Problemlösung an. Typische beobachtbare Leistungen sind „ausführen“, „anwenden“, „beurteilen“, „ermitteln“, „entwerfen“, „analysieren“.

**Schlüsselworte:**

anwenden, einsetzen, ausführen, nutzen, Verfahren verstehen, Verfahren anwenden

**Beispiel:**

- identifiziert Grenzwerte für gültige bzw. ungültige Äquivalenzklassen
- selektiert aus einem Zustandsdiagramm die notwendigen Testfälle zur Überdeckung aller Statusübergänge

### 6.4 Taxonomiestufe 4: Analysieren (K4)

Der Lernende kann zum besseren Verständnis die Informationen bezüglich eines Vorgehens oder Ablaufs in ihre Einzelbestandteile aufschlüsseln und zwischen Sachverhalten und abgeleiteten Schlussfolgerungen unterscheiden. Typische Anwendungen sind die Analyse eines Dokuments, einer Software oder Projektsituation und der Vorschlag angemessener Maßnahmen zur Problemlösung.

**Schlüsselworte:**

analysieren, organisieren, Zusammenhänge erkennen, integrieren, kurz skizzieren, zergliedern, strukturieren, zuordnen, zerlegen, differenzieren, unterschiedlich behandeln, unterscheiden, fokussieren, auswählen

**Beispiel:**

- Analysieren Sie Produktrisiken und schlagen Sie vorbeugende oder korrigierende Maßnahmen vor.
- Beschreiben Sie, welche Teile eines Abweichungsberichts einen Sachverhalt darstellen und bei welchen es sich um Schlussfolgerungen aus den Ergebnissen handelt.

**Referenz (für die kognitiven Ebenen von Lernzielen)**

Anderson, L. W. and Krathwohl, D. R. (eds): "A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives", Allyn& Bacon: Columbus, Ohio, 2001.

## 7 Anhang D – Verwendete Regeln bei der Erstellung des Lehrplans<sup>7</sup>

Die Arbeitsgruppe wendete die unten aufgeführten Regeln auf die Erstellung und Prüfung des Lehrplans an.

### 7.1 Allgemeine Regeln

- SG1. Der Lehrplan soll durch Personen mit null bis sechs (oder mehr) Monaten Testerfahrung verstanden und aufgenommen werden können. (6 MONATE)
- SG2. Der Lehrplan ist praxisorientiert und nicht theoretisch. (PRAKTISCH)
- SG3. Der Lehrplan soll für den angesprochenen Leserkreis klar und eindeutig sein. (KLAR)
- SG4. Der Lehrplan soll für Personen in verschiedenen Ländern verständlich sein und soll leicht in verschiedene Sprachen übersetzt werden können. (ÜBERSETZBAR)
- SG5. Das Original des Lehrplans ist in deutscher Sprache erstellt. (HOCHDEUTSCH)

### 7.2 Aktualität

- SC1. Der Lehrplan berücksichtigt die neuen Entwicklungen im Bereich Testen und gibt die aktuellen, allgemein anerkannten und bewährten Verfahren des Softwaretestens wieder. Der Lehrplan soll alle drei bis fünf Jahre einem Review unterzogen werden. (AKTUELL)
- SC2. Der Lehrplan soll möglichst zeitlos erstellt und von Marktströmungen unabhängig sein, um eine Lebensdauer von drei bis fünf Jahren zu ermöglichen. (LEBENSDAUER)

### 7.3 Lernziele

- LO1. Lernziele unterscheiden zwischen Lerninhalten, welche erkannt (kognitive Stufe K1), verstanden (K2) bzw. auf neue Aufgaben angewendet (K3) werden müssen und die dazu dienen, ein Dokument, eine Software oder Projektsituation im Zusammenhang zu analysieren (K4). (WISSENSEBENE)
- LO2. Die Beschreibung der Themen soll konsistent zu den Lernzielen formuliert sein. (LERNZIEL-KONSISTENZ)
- LO3. Um Lernziele zu illustrieren, sollen für alle Hauptteile des Lehrplans Beispielprüfungsfragen bereitgestellt werden (LO-PRÜFUNG)

### 7.4 Gesamtstruktur

- ST1. Die Struktur des Lehrplans soll klar sein und Querverweise zwischen einzelnen Teilen, zu Prüfungsfragen und zu anderen relevanten Dokumenten erlauben. (CROSS-REF)
- ST2. Inhaltliche Überschneidungen zwischen einzelnen Kapiteln sollen minimiert sein. (ÜBERSCHNEIDUNG)
- ST3. Die Abschnitte des Lehrplans sind einheitlich aufgebaut. (STRUKTUR-KONSISTENT)

---

<sup>7</sup> Übernommen aus: ISTQB Certified Tester Foundation Level Lehrplan, V1.01, deutschsprachige Ausgabe vom 19.04.2013.

- ST4. Der Lehrplan enthält Version, Freigabedatum und Seitennummer auf jeder Seite. (VERSION)
- ST5. Der Lehrplan enthält als Leitfaden die jedem Kapitel zugewiesene Zeit (um die relative Bedeutung jedes Themas widerzuspiegeln). (ZEITDAUER)

## 7.5 Referenzen<sup>8</sup>

- SR1. Quellen und Referenzen zu Konzepten im Lehrplan erlauben es den Ausbildungsanbietern, mehr Informationen zum Thema zu finden. (REFS)
- SR2. Wo es keine leicht bestimmbareren und klaren Quellen gibt, enthält der Lehrplan mehr Details. Die Definitionen von Begriffen beispielsweise stehen im Glossar, deshalb enthält der Lehrplan ausschließlich den Begriff. (KEIN-REF DETAIL)

## 7.6 Informationsquellen

Die verwendeten Begriffe im CORE-Lehrplan sind im „ISTQB Glossary of Terms used in Software Testing“ definiert, das von der ISTQB-Webseite ([www.istqb.org](http://www.istqb.org)) heruntergeladen werden kann.

Eine entsprechende englisch-/deutschsprachige Fassung dieses Glossars mit der Bezeichnung „ISTQB®/GTB Standard Glossar der Testbegriffe“ wird auf der Webseite des GTB e.V. ([www.german-testing-board.info](http://www.german-testing-board.info)) bzw. anderer deutschsprachiger Testing Boards zur Verfügung gestellt.

Eine Liste empfohlener Bücher wird mit dem Lehrplan publiziert. Die direkten Referenzen befinden sich jeweils im Abschnitt Referenzen.

---

<sup>8</sup> Übernommen aus: ISTQB Certified Tester Foundation Level Lehrplan, V1.01, deutschsprachige Ausgabe vom 19.04.2013.

## **8 Anhang E – Hinweise für Ausbildungsanbieter<sup>9</sup>**

Jeder Überschrift eines Hauptkapitels im Lehrplan ist die vorgegebene Unterrichtsdauer in Minuten zugeordnet. Diese Angabe dient als Leitfaden für die relative, zeitliche Gewichtung der Kapitel in einem akkreditierten Kurs. Zusätzlich legt diese Zahl die minimale Zeitdauer für ein Kapitel fest. Ausbildungsanbieter können mehr Zeit darauf verwenden, und Kandidaten können mehr Zeit als vorgegeben in ihrem Studium und Analyse verwenden. Ein Lehrprogramm eines Kurses kann eine andere Reihenfolge der Kapitel als im Lehrplan enthalten.

Der Lehrplan enthält Referenzen zu gängigen Standards, welche für die Vorbereitung der Ausbildungsunterlagen verwendet werden müssen. Jeder Standard muss in der im Lehrplan referenzierten Version verwendet werden. Andere nicht referenzierte Publikationen, Dokumentvorlagen oder Standards können ebenso verwendet werden, sind aber nicht Gegenstand der Prüfung.

Alle K3- und K4-Lernziele machen praktische Übungen erforderlich, die in den Trainingsunterlagen enthalten sein müssen.

---

<sup>9</sup> Übernommen aus: ISTQB Certified Tester Foundation Level Lehrplan, V1.01, deutschsprachige Ausgabe vom 19.04.2013.

## 9 Anhang F – Release Notes

Siehe Änderungsverzeichnis

## 10 Index

Anforderungsbasierter Test	28
Assessment	17, 28
<b>Automobiler Produktentstehungsprozess</b>	29
Automotive Safety Integrity Level (ASIL)	28
Automotive SPICE	28
AUTOMOTIVE-PEP	4, 20, 21
AUTOSAR	28
<b>Betriebsmodus(Automotive)</b>	29
<b>E/E-Entwicklungsprozess</b>	29
<b>E/E-System</b>	29
<b>Echtzeitfähigkeit</b>	29
<b>Freigabe</b>	29
<b>Funktionale Sicherheit</b>	29
<b>Hardware-in-the-Loop (HiL)</b>	30
<b>Integrationsstufe</b>	30
<b>Integrationsstufenkonzept</b>	30
<b>MISRA-C</b>	30
<b>Musterstand</b>	30
<b>Produktentstehung</b>	30
<b>Reifegrad</b>	30
<b>Reifegraddimension</b>	31
<b>Reifegradstufe</b>	31
<b>Sicherheitsanforderung (nach ISO)</b>	31
<b>Sicherheitsfunktion (nach IEC)</b>	31
<b>Sicherheitsintegrität</b>	31
<b>Sicherheitsnachweis</b>	31
<b>Sicherheitsziel (nach ISO)</b>	31
<b>Software-in-the-Loop (SiL)</b>	31
<b>System</b>	31
<b>Systemstruktur</b>	32
<b>Testbarkeitsanalyse</b>	32
<b>Verifikationskriterium</b>	32
<b>Virtuelle Umgebung</b>	32