

ATZ/MTZ-Fachbuch

Jörg Schäuuffele · Thomas Zurawka

# Automotive Software Engineering

Grundlagen, Prozesse, Methoden und  
Werkzeuge

*7. Auflage*

 Springer Vieweg

---

# **ATZ/MTZ-Fachbuch**

In der Reihe ATZ/MTZ-Fachbuch vermitteln Fachleute, Forscher und Entwickler aus Hochschule und Industrie Grundlagen, Theorien und Anwendungen der Fahrzeug- und Verkehrstechnik. Die komplexe Technik, die moderner Mobilität zugrunde liegt, bedarf eines immer größer werdenden Fundus an Informationen, um die Funktion und Arbeitsweise von Komponenten sowie Systemen zu verstehen. Fahrzeuge aller Verkehrsträger sind ebenso Teil der Reihe, wie Fragen zu Energieversorgung und Infrastruktur.

Das ATZ/MTZ-Fachbuch wendet sich an Ingenieure aller Mobilitätsfelder, an Studierende, Dozenten und Professoren. Die Reihe wendet sich auch an Praktiker aus der Fahrzeug- und Zulieferindustrie, an Gutachter und Sachverständige, aber auch an interessierte Laien, die anhand fundierter Informationen einen tiefen Einblick in die Fachgebiete der Mobilität bekommen wollen.

---

Jörg Schäuffele · Thomas Zurawka

# Automotive Software Engineering

Grundlagen, Prozesse, Methoden und  
Werkzeuge

7. Auflage

Jörg Schäuffele  
Erligheim, Deutschland

Thomas Zurawka  
Stuttgart, Deutschland

ISSN 2628-104X

ISSN 2628-1058 (electronic)

ATZ/MTZ-Fachbuch

ISBN 978-3-658-43542-4

ISBN 978-3-658-43543-1 (eBook)

<https://doi.org/10.1007/978-3-658-43543-1>

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <https://portal.dnb.de> abrufbar.

© Der/die Herausgeber bzw. der/die Autor(en), exklusiv lizenziert an Springer Fachmedien Wiesbaden GmbH, ein Teil von Springer Nature 2003, 2004, 2006, 2010, 2013, 2016, 2024

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsgesetz zugelassen ist, bedarf der vorherigen Zustimmung des Verlags. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die Wiedergabe von allgemein beschreibenden Bezeichnungen, Marken, Unternehmensnamen etc. in diesem Werk bedeutet nicht, dass diese frei durch jedermann benutzt werden dürfen. Die Berechtigung zur Benutzung unterliegt, auch ohne gesonderten Hinweis hierzu, den Regeln des Markenrechts. Die Rechte des jeweiligen Zeicheninhabers sind zu beachten.

Der Verlag, die Autoren und die Herausgeber gehen davon aus, dass die Angaben und Informationen in diesem Werk zum Zeitpunkt der Veröffentlichung vollständig und korrekt sind. Weder der Verlag noch die Autoren oder die Herausgeber übernehmen, ausdrücklich oder implizit, Gewähr für den Inhalt des Werkes, etwaige Fehler oder Äußerungen. Der Verlag bleibt im Hinblick auf geografische Zuordnungen und Gebietsbezeichnungen in veröffentlichten Karten und Institutionsadressen neutral.

Planung/Lektorat: Axel Garbers

Springer Vieweg ist ein Imprint der eingetragenen Gesellschaft Springer Fachmedien Wiesbaden GmbH und ist ein Teil von Springer Nature.

Die Anschrift der Gesellschaft ist: Abraham-Lincoln-Str. 46, 65189 Wiesbaden, Germany

Das Papier dieses Produkts ist recycelbar.

---

## Vorwort zur 7. Auflage

Wir freuen uns sehr, dass unser Buch bereits in der 7. Auflage erscheinen kann. Seit der 1. Auflage sind nun rund 20 Jahre vergangen und seit vielen Jahren liegen auch englische, japanische, koreanische und chinesische Ausgaben vor. Uns erfreut zudem, dass weltweit an zahlreichen Universitäten und Hochschulen Vorlesungen und Studiengänge angeboten werden, die dieses Buch als Grundlage für die Lehrveranstaltungen verwenden.

Nach seiner über 130-jährigen Geschichte ist das Kraftfahrzeug derzeit durch eine sehr rasante Weiterentwicklung gekennzeichnet, die von den Megatrends Elektrifizierung, Connectivity und Automatisiertes Fahren getrieben wird.

Seit Anfang der 1970er Jahre ist diese Entwicklung geprägt von einem anhaltenden Anstieg des Einsatzes von elektronischen Systemen und von Software im Fahrzeug und in Zukunft zunehmend auch im IT-Backend.

Im Mittelpunkt der Überarbeitung für die 7. Auflage stehen deshalb die Themen, die gerade zu sehr großen Umbrüchen der E/E-Architekturen führen: Service-orientierte Architekturen und service-orientierte Kommunikation, High-Performance-Computer, Zonen-Architekturen, Connectivity und Cloud Computing im IT-Backend. Schlagworte wie „Software-defined Vehicle“, „Vehicle Operating System“ (Vehicle OS), „DevOps“, „Continuous Integration/Continuous Delivery“ und „Software Updates over the Air“ (OTA) sind in aller Munde. Diese großen Themen müssen vor dem Hintergrund der stufenweisen Einführung des automatisierten Fahrens bei weiter zunehmenden Anforderungen an die Sicherheit – sowohl im Sinne von Funktionaler Sicherheit (Functional Safety) als auch im Sinne von Informationssicherheit (Cyber Security) – beherrscht werden. Insgesamt führen diese Veränderungen zu ganz neuartigen E/E-Architekturen und es werden völlig neue und erweiterte technische Freiheitsgrade für die Umsetzung von Fahrzeugfunktionen möglich. Außerdem werden bisher bestehende Einschränkungen und Zielkonflikte etwa bezüglich begrenzter Rechenleistung im Fahrzeug obsolet. Wir haben daher alle relevanten Kapitel aktualisiert und wo notwendig, neue Abschnitte und Kapitel aufgenommen.

Neben neuen Funktionen für den Fahrer und die Passagiere des Fahrzeugs führt dies auch zu gravierenden Veränderungen in der Entwicklung, in der Produktion und im

Service von Fahrzeugen. Die etablierten und bewährten Prozesse und Methoden müssen auf den Prüfstand gestellt werden, um die Potentiale des erweiterten Entwurfsraums für Fahrzeugfunktionen zukunftsicher zu heben. So wird die Software zum SOP des Fahrzeugs nicht mehr wie bisher fertiggestellt und dann nur noch bei schweren Fehlern korrigiert. Vielmehr wird die Software eines Fahrzeugs zum SOP in einer ersten Version ausgeliefert und dann ständig aktualisiert („always fresh“).

Zur Beherrschung der dadurch weiterwachsenden Komplexität sind weiterhin ein solides Grundlagenverständnis sowie Prozesse, Methoden und Werkzeuge unerlässlich, die die fahrzeugspezifischen Randbedingungen berücksichtigen.

Grundlegende charakteristische Anforderungen an Software im Automobil sind der Entwurf für eingebettete und verteilte Echtzeitsysteme bei hohen Sicherheits-, Robustheits- und Verfügbarkeitsanforderungen sowie Kostendruck und vergleichsweise lange Produktlebenszyklen. Diese Anforderungen stehen bei diesem Buch im Vordergrund.

Zur Entwicklung von Software für elektronische Systeme von Fahrzeugen wurden in den letzten 20 Jahren eine Reihe von Vorgehensweisen und Standards wie AUTOSAR Classic und AUTOSAR Adaptive, ISO 26262 und ISO 21448 für die funktionale Sicherheit, ISO 21434 für Cyber Security und Automotive SPICE entwickelt. Zahlreiche neue Initiativen zur Standardisierung im Bereich „Software-defined Vehicle“ wurden in jüngster Zeit gestartet. Diese Vorgehensweisen können wohl am besten unter dem Begriff „Automotive Software Engineering“ zusammengefasst werden.

Damit ist eine komplexe Begriffswelt entstanden, mit der wir ständig konfrontiert werden. Es wird immer schwieriger, genau zu verstehen, was sich hinter den Vokabeln verbirgt. Erschwerend kommt hinzu, dass manche Begriffe mehrfach in unterschiedlichem Zusammenhang verwendet werden. So etwa der Begriff Prozess, der im Zusammenhang mit der Regelungstechnik, aber auch mit Echtzeitsystemen oder generell mit Vorgehensweisen in der Entwicklung benutzt wird. Nach einem Überblick zu Beginn werden deshalb in diesem Buch die wichtigsten Begriffe definiert und durchgängig so verwendet. In den folgenden Kapiteln stehen Prozesse, Methoden und Werkzeuge für die Entwicklung von Software für die elektronischen Systeme des Fahrzeugs und seines Umfelds im Mittelpunkt. Eine entscheidende Rolle spielen dabei die Wechselwirkungen zwischen der Software-Entwicklung als Fachdisziplin und der übergreifenden Systementwicklung, die alle beteiligten Komponenten berücksichtigen muss. Die dargestellten Prozesse haben Modellcharakter. Das bedeutet, die Prozesse sind ein abstraktes, idealisiertes Bild der täglichen Praxis. Sie können für verschiedene Entwicklungsprojekte als Orientierung dienen, müssen aber vor der Einführung in einem konkreten Projekt bewertet und eventuell angepasst werden. Auf eine klare und verständliche Darstellung haben wir deshalb großen Wert gelegt.

Wegen der Breite des Aufgabengebietes können nicht alle Themen in der Tiefe behandelt werden. Wir beschränken uns aus diesem Grund auf Gebiete mit automobilspezifischem Charakter und verweisen bei Bedarf auf die weiterführende Literatur.

---

## Beispiele aus der Praxis

Ein Prozess ist nur dann ein erfolgreiches Instrument für ein Entwicklungsteam, wenn die Vorteile der nachvollziehbaren Bearbeitung umfangreicher Aufgabenstellungen in der Praxis anerkannt werden. Dieses Buch soll daher kein theoretisches Lehrbuch sein, das sich jenseits der Praxis bewegt. Alle Anregungen basieren auf praktischen Anwendungsfällen, die wir anhand von Beispielen anschaulich darstellen. Die Vielzahl von Erfahrungen, die wir in enger Zusammenarbeit mit Fahrzeugherstellern und Zulieferern sammeln konnten, wurde dabei berücksichtigt. Dazu gehören Serienentwicklungen mit den dazugehörigen Produktions- und Serviceaspekten genauso wie Forschungs- und Vorentwicklungsprojekte.

---

## Leserkreis

Wir wollen mit dem vorliegenden Buch alle Mitarbeiter bei Fahrzeugherstellern, Zulieferern, Entwicklungspartnern, Software-Herstellern, Halbleiter-Herstellern, Kommunikationsdienstleistern und Cloud-Anbietern ansprechen, die in der Entwicklung, in der Produktion und im Service mit Software im Fahrzeug und zunehmend auch im IT-Backend beschäftigt sind. Wir hoffen, dass wir nützliche Anregungen weitergeben und zu einem gemeinsamen, ganzheitlichen Verständnis und Weltbild beitragen können.

Dieses Buch soll auch weiterhin eine Basis zur Ausbildung von Studierenden und zur Einarbeitung von neuen Mitarbeitern zur Verfügung stellen. Grundkenntnisse in der Steuerungs- und Regelungstechnik, in der Systemtheorie und in der Software-Technik sind von Vorteil, aber nicht Voraussetzung.

Sicherlich wird an einigen Stellen der Wunsch nach einer detaillierteren Darstellung aufkommen. Wir freuen uns deshalb zu allen behandelten Themen über Hinweise und Verbesserungsvorschläge. Zahlreiche konstruktive Rückmeldungen der Leser zur 6. Auflage wurden bei der 7. Auflage des Buches berücksichtigt.

---

## Danksagungen

An dieser Stelle möchten wir uns besonders bei allen unseren Kunden aus der Automobilindustrie für die jahrzehntelange, vertrauensvolle Zusammenarbeit bedanken. Ohne diesen Erfahrungsaustausch wäre diese 7. Auflage nicht möglich gewesen.

Des Weiteren bedanken wir uns bei der BMW Group für die freundliche Zustimmung, dass wir in diesem Buch Erfahrungen darstellen dürfen, die wir in BMW Projekten – im Falle des erstgenannten Autors auch als BMW Mitarbeiter – gesammelt haben. Dies schließt auch Empfehlungen für Serienprojekte bei BMW ein. Wir bedanken uns



besonders bei Herrn Heinz Merkle, Herrn Dr. Helmut Hochschwarzer, Herrn Dr. Maximilian Fuchs, Herrn Prof. Dr. Dieter Nazareth sowie allen ihren Mitarbeitern.

Viele Vorgehensweisen und Methoden entstanden während der jahrelangen, engen und vertrauensvollen Zusammenarbeit zwischen der Vector Informatik GmbH, der ETAS GmbH und der Robert Bosch GmbH. Die dabei entwickelten und inzwischen breit eingesetzten Vorgehensweisen finden sich in diesem Buch an vielen Stellen wieder. Wir sagen dafür den Mitarbeiterinnen und Mitarbeitern des Bereichs Mobility Solutions der Robert Bosch GmbH vielen Dank.

Wir bedanken uns außerdem herzlich bei Herrn Dr. Siegfried Dais, Herrn Dr. Klaus Grimm, Herrn Dr. Wolfgang Runge, Herrn Prof. Dr. Manfred Broy, Herrn Prof. Dr. Hans-Christian Reuss und Herrn Prof. Dr. Peter Göhner für ihre einleitenden Geleitworte.

Für die sorgfältige und kritische Durchsicht des Manuskripts bedanken wir uns insbesondere bei Mirko Ciecinski, Dr. Daniel Kanth, Roland Jeutter, Dr. Michael Nicolaou, Dr. Oliver Schlüter, Dr. Kai Werther, Hans-Jörg Wolff und Wolfgang Haug. Unser Dank gilt nicht zuletzt dem Springer Vieweg Verlag sowie Herrn Markus Braun für die wiederum sehr gute und äußerst geduldige Zusammenarbeit bei der Vorbereitung der 7. Auflage.

Stuttgart  
März 2024

Jörg Schäuuffele  
Dr. Thomas Zurawka

---

## Zur Bedeutung von Software im Automobil

### **Neue Entwicklungsmethoden zur Beherrschung der Komplexität erforderlich**

Auch in Zukunft ist Software die Technik, um komplexe Algorithmen im Kraftfahrzeug umzusetzen. Der nahezu exponentiell wachsende Software-Umfang wird getrieben durch die Funktionszunahme in den Feldern sicher, sauber, sparsam und seit einiger Zeit auch im Bereich der Fahrerassistenz. Die Beherrschung der daraus resultierenden Komplexität ist für Fahrzeughersteller wie Zulieferer eine Herausforderung, die nur mit leistungsfähigen Methoden und Werkzeugen möglich ist. Mit ihrer Hilfe gilt es, die sichere Funktion der Software und der Systeme zu gewährleisten. Das vorliegende Buch liefert dazu vielfältige Anregungen zur Gestaltung von Entwicklungsprozessen und zum Einsatz von Methoden und Werkzeugen.

*Dr. Siegfried Dais, vormals Stellvertretender Vorsitzender der Geschäftsführung der Robert Bosch GmbH, Stuttgart*

### **Standardisierung von Software und Entwicklungsmethoden**

Standardisierung ist ein wichtiger Erfolgsfaktor für die effiziente Entwicklung qualitativ hochwertiger und sicherer Elektrik-/Elektronik- und Software-Systeme. Neben AUTOSAR hat auch der Standard ISO 26262 zur Funktionssicherheit von Straßenfahrzeugen bei Fahrzeugherstellern und deren Zulieferern erheblichen Einfluss sowohl auf die Prozesse, Methoden und Werkzeuge als auch auf die Organisation. Eine Angleichung der Entwicklungsmethodik von der Anforderungsanalyse bis zum Systemtest ist dabei ein unverzichtbarer Schritt. Das vorliegende Buch adressiert dieses Thema in der 7. Auflage in anschaulicher und fundierter Weise.

*Dr. Klaus Grimm, vormals Leiter E/E- und Software-Technologien, Daimler AG, Stuttgart*

### **Vom Kostentreiber zum Wettbewerbsvorteil**

Technischer Vorsprung in der Automobilindustrie kann nur durch eine Vorreiterrolle in der Software-Technik erreicht werden. Die erfolgreiche Zusammenarbeit von Ingenieuren unterschiedlicher Fachrichtungen in der Systementwicklung erfordert jedoch ein einheitliches Hintergrundwissen, eine gemeinsame Begriffswelt und ein geeignetes

Vorgehensmodell. Die in diesem Buch dargestellten Grundlagen und Methoden demonstrieren dies anhand von Beispielen aus der Praxis sehr eindrucksvoll. Neue Entwicklungen und hier vor allem vernetzte Fahrerassistenzsysteme tragen nicht unerheblich zu einer weiteren Komplexitätssteigerung der „Embedded Software“ bei. Strukturierte, nachvollziehbare Regeln und Prozesse – wie in diesem Buch beschrieben – werden dadurch immer unverzichtbarer.

*Dr.-Ing. Wolfgang Runge, Runge Consult, Ravensburg*

### **Innovationstreiber Software im Automobil**

Software im Automobil zeigt sich als zentraler Innovationstreiber und das nicht nur in den vergangenen 30 Jahren. Auch in den nächsten 10 bis 20 Jahren sind enorme Innovationsschübe zu erwarten. Möglich wird das nur auf Basis einer uneingeschränkten Beherrschung des Entwicklungsprozesses für Eingebettete Software im Automobil. Der exponentielle Zuwachs an Funktionalität geht einher mit einer entsprechenden Steigerung der Komplexität und der Softwareumfänge im Automobil. Das wirkt sich auch entsprechend auf die Kosten aus. Eine hohe Systematik und eine wissenschaftliche Durchdringung des Gegenstandes ist deshalb für eine Weiterentwicklung des Themas ein Muss. Dieses Buch liefert einen hervorragenden Beitrag in dieser Richtung, indem es mit großer Systematik alle Gesichtspunkte der Entwicklung von Software im Automobil aufzeigt.

*Prof. Dr. Dr. h.c. Manfred Broy, vormals Institut für Informatik, Technische Universität München*

### **Ein Grundlagenbuch mit starkem Praxisbezug**

Seit 2005 wird das Buch „Automotive Software Engineering“ mit gutem Erfolg als Vorlesungsskript für die gleichnamige Lehrveranstaltung im Studiengang Fahrzeug- und Motorentchnik an der Universität Stuttgart eingesetzt. Es ist ein Grundlagenbuch, das einen starken Praxisbezug hat. Gerade für Ingenieure ist es eine große Hilfe, da es in einen Umdenkprozess führt: weg vom Bottom-Up-Approach hin zum Top-Down-Prozess. Denn die Fortschritte in der Automobilentwicklung haben es mit sich gebracht, dass die Beherrschung der Komplexität zum wichtigsten Ziel geworden ist. Eine systematische und methodische Vorgehensweise bei der Funktionsentwicklung ist damit unabdingbar geworden. Ausgehend von der Architektur mechatronischer Systeme schlägt das Buch eine Brücke zum Kernprozess, der sich am V-Modell orientiert, um darauf aufbauend die nötigen Methoden und Werkzeuge vorzustellen. Regelmäßige Aktualisierungen machen das Buch zu einem Standardwerk für jeden, der sich mit Funktionsentwicklung in der Fahrzeugtechnik beschäftigt.

*Prof. Dr.-Ing. Hans-Christian Reuss, Institut für Verbrennungsmotoren und Kraftfahrwesen, Universität Stuttgart*

**Ausbildung als Chance und Herausforderung**

Gerade für den Raum Stuttgart spielt der Fahrzeugbau eine herausragende Rolle. Die Entwicklungszentren bedeutender Fahrzeughersteller und Zulieferer bieten hier viele Arbeitsplätze. Die Ausbildung in der Software-Technik ist ein fester Bestandteil eines Ingenieurstudienganges an der Universität Stuttgart. Dieses Buch bietet die Chance, praktische Erfahrungen in der Automobilindustrie bereits in der Ausbildung zu berücksichtigen. Viele der vorgestellten Vorgehensweisen können sogar Vorbildcharakter für andere Branchen haben.

*Prof. Dr.-Ing. Dr. h.c. Peter Göhner, vormals Institut für Automatisierungs- und Software-Technik, Universität Stuttgart*

---

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung und Überblick</b>	<b>1</b>
1.1	Das System Fahrer-Fahrzeug-Umwelt	2
1.1.1	Physikalisches und digitales Anwendererlebnis	3
1.1.2	Aufbau und Wirkungsweise elektronischer Systeme	3
1.1.3	Elektronische Systeme des Fahrzeugs und der Umwelt	6
1.2	Überblick über die elektronischen Systeme des Fahrzeugs	7
1.2.1	Elektronische Systeme des Antriebsstrangs	9
1.2.2	Elektronische Systeme des Fahrwerks	12
1.2.3	Elektronische Systeme der Karosserie	14
1.2.4	Infotainment-Systeme	16
1.2.5	Fahrerassistenzsysteme und automatisiertes Fahren	16
1.2.6	Zusammenfassung und Ausblick	20
1.3	Überblick über die logische Systemarchitektur	23
1.3.1	Funktions- und Steuergerätenetzwerk des Fahrzeugs	23
1.3.2	Logische Systemarchitektur für Steuerungs-/Regelungs- und Überwachungssysteme	23
1.4	Prozesse in der Fahrzeugentwicklung	25
1.4.1	Überblick über die Fahrzeugentwicklung	25
1.4.2	Überblick über die Entwicklung von elektronischen Systemen	26
1.4.3	Kernprozess zur Entwicklung von elektronischen Systemen und Software	29
1.4.4	Unterstützungsprozesse zur Entwicklung von elektronischen Systemen und Software	32
1.4.5	Produktion und Service von elektronischen Systemen und Software	35
1.5	Methoden und Werkzeuge in der Fahrzeugentwicklung	35
1.5.1	Modellbasierte Entwicklung	36
1.5.2	Integrierte Qualitätssicherung	38

1.5.3	Reduzierung des Entwicklungsrisikos . . . . .	39
1.5.4	Standardisierung und Automatisierung . . . . .	41
1.5.5	Entwicklungsschritte im Fahrzeug . . . . .	44
<b>2</b>	<b>Grundlagen . . . . .</b>	<b>45</b>
2.1	Steuerungs- und regelungstechnische Systeme . . . . .	46
2.1.1	Modellbildung . . . . .	46
2.1.2	Blockschaltbilder . . . . .	46
2.2	Diskrete Systeme . . . . .	51
2.2.1	Zeitdiskrete Systeme und Signale . . . . .	52
2.2.2	Wertdiskrete Systeme und Signale . . . . .	53
2.2.3	Zeit- und wertdiskrete Systeme und Signale . . . . .	54
2.2.4	Zustandsautomaten . . . . .	55
2.3	Eingebettete Systeme . . . . .	56
2.3.1	Aufbau von Mikrocontrollern . . . . .	58
2.3.2	Speichertechnologien . . . . .	60
2.3.3	Programmierung von Mikrocontrollern . . . . .	63
2.4	Echtzeitsysteme . . . . .	71
2.4.1	Festlegung von Tasks . . . . .	71
2.4.2	Festlegung von Echtzeitanforderungen . . . . .	73
2.4.3	Zustände von Tasks . . . . .	76
2.4.4	Strategien für die Zuteilung des Prozessors . . . . .	77
2.4.5	Aufbau von Echtzeitbetriebssystemen . . . . .	82
2.4.6	Interaktion zwischen Tasks . . . . .	83
2.5	Verteilte und vernetzte Systeme . . . . .	90
2.5.1	Logische und technische Systemarchitektur . . . . .	92
2.5.2	Festlegung der logischen Kommunikationsbeziehungen . . . . .	93
2.5.3	Festlegung der technischen Netzwerktopologie . . . . .	96
2.5.4	Strategien für die Zuteilung des Busses . . . . .	97
2.5.5	Signalorientierte Kommunikation . . . . .	99
2.5.6	Serviceorientierte Kommunikation . . . . .	106
2.6	Zuverlässigkeit, funktionale Sicherheit, Überwachung und Diagnose von Systemen . . . . .	116
2.6.1	Grundbegriffe . . . . .	117
2.6.2	Zuverlässigkeit und Verfügbarkeit von Systemen . . . . .	119
2.6.3	Funktionale Sicherheit von Systemen . . . . .	123
2.6.4	Überwachung und Diagnose von Systemen . . . . .	127
2.6.5	Aufbau des Überwachungssystems elektronischer Steuergeräte . . . . .	132
2.6.6	Aufbau des Diagnosesystems elektronischer Steuergeräte . . . . .	135
2.7	Produktlinienansatz und Systemvarianten . . . . .	139

2.8	Moderne Elektrik-/Elektronik- und Software-Architekturen . . . . .	145
2.8.1	Domänen- und Zonenarchitektur . . . . .	145
2.8.2	Software-defined Vehicle . . . . .	147
2.8.3	Signalorientierte und serviceorientierte Teilarchitektur . . . . .	147
2.8.4	Unterschiedliche Software-Domänen, APIs und Integration auf den HPCs . . . . .	149
2.8.5	Vehicle Operating System . . . . .	151
<b>3</b>	<b>Unterstützungsprozesse zur Entwicklung von elektronischen Systemen und Software . . . . .</b>	<b>155</b>
3.1	Grundbegriffe der Systemtheorie . . . . .	155
3.2	Vorgehensmodelle und Standards . . . . .	158
3.3	Konfigurationsmanagement . . . . .	160
3.3.1	Produkt und Lebenszyklus . . . . .	160
3.3.2	Varianten und Skalierbarkeit . . . . .	161
3.3.3	Versionen und Konfigurationen . . . . .	161
3.4	Projektmanagement . . . . .	165
3.4.1	Projektplanung . . . . .	166
3.4.2	Projektverfolgung und Risikomanagement . . . . .	171
3.5	Lieferantenmanagement . . . . .	172
3.5.1	System- und Komponentenverantwortung . . . . .	172
3.5.2	Schnittstellen für die Spezifikation und Integration . . . . .	172
3.5.3	Festlegung des firmenübergreifenden Entwicklungsprozesses . . . . .	173
3.6	Anforderungsmanagement . . . . .	175
3.6.1	Erfassen der Benutzeranforderungen . . . . .	176
3.6.2	Verfolgen von Anforderungen . . . . .	179
3.7	Qualitätssicherung . . . . .	180
3.7.1	Integrations- und Testschritte . . . . .	182
3.7.2	Maßnahmen zur Qualitätssicherung von Software . . . . .	182
<b>4</b>	<b>Kernprozess zur Entwicklung von elektronischen Systemen und Software . . . . .</b>	<b>185</b>
4.1	Anforderungen und Randbedingungen . . . . .	187
4.1.1	System- und Komponentenverantwortung . . . . .	187
4.1.2	Abstimmung zwischen System- und Software-Entwicklung . . . . .	187
4.1.3	Modellbasierte Software-Entwicklung . . . . .	189
4.2	Grundbegriffe . . . . .	190
4.2.1	Prozesse . . . . .	190
4.2.2	Methoden und Werkzeuge . . . . .	191
4.3	Spezifikation der logischen Systemarchitektur . . . . .	192

4.4	Spezifikation der technischen Systemarchitektur . . . . .	196
4.4.1	Analyse und Spezifikation steuerungs- und regelungstechnischer Systeme . . . . .	199
4.4.2	Analyse und Spezifikation von Echtzeitsystemen . . . . .	200
4.4.3	Analyse und Spezifikation verteilter und vernetzter Systeme . . .	201
4.4.4	Analyse und Spezifikation zuverlässiger und sicherer Systeme . . . . .	202
4.5	Spezifikation der Software-Architektur . . . . .	203
4.5.1	Spezifikation der Software-Komponenten und ihrer Schnittstellen . . . . .	204
4.5.2	Spezifikation der Software-Schichten . . . . .	207
4.5.3	Spezifikation der Betriebszustände . . . . .	208
4.6	Spezifikation der Software-Komponenten . . . . .	210
4.6.1	Spezifikation des Datenmodells . . . . .	210
4.6.2	Spezifikation des Verhaltensmodells . . . . .	211
4.6.3	Spezifikation des Echtzeitmodells . . . . .	214
4.7	Design und Implementierung der Software-Komponenten . . . . .	216
4.7.1	Berücksichtigung der geforderten nichtfunktionalen Produkteigenschaften . . . . .	216
4.7.2	Design und Implementierung des Datenmodells . . . . .	219
4.7.3	Design und Implementierung des Verhaltensmodells . . . . .	220
4.7.4	Design und Implementierung des Echtzeitmodells . . . . .	222
4.8	Test der Software-Komponenten . . . . .	222
4.9	Integration der Software-Komponenten . . . . .	223
4.9.1	Erzeugung des Programm- und Datenstands . . . . .	224
4.9.2	Erzeugung der Beschreibungsdateien . . . . .	225
4.9.3	Erzeugung der Dokumentation . . . . .	226
4.10	Integrationstest der Software . . . . .	228
4.11	Integration der Systemkomponenten . . . . .	228
4.11.1	Integration von Software und Hardware . . . . .	229
4.11.2	Integration von Steuergeräten, Sollwertgebern, Sensoren und Aktuatoren . . . . .	230
4.12	Integrationstest des Systems . . . . .	232
4.13	Kalibrierung . . . . .	234
4.14	System- und Akzeptanztest . . . . .	235
<b>5</b>	<b>Methoden und Werkzeuge in der Entwicklung . . . . .</b>	<b>239</b>
5.1	Offboard-Schnittstelle zwischen Steuergerät und Werkzeug . . . . .	242
5.2	Spezifikation der technischen Systemarchitektur . . . . .	243
5.2.1	Analyse und Spezifikation steuerungs- und regelungstechnischer Systeme . . . . .	244
5.2.2	Analyse und Spezifikation von Echtzeitsystemen . . . . .	248



---

5.2.3	Analyse und Spezifikation verteilter und vernetzter Systeme .....	254
5.2.4	Analyse und Spezifikation zuverlässiger und sicherer Systeme .....	259
5.3	Spezifikation von Software-Funktionen und deren Validierung .....	267
5.3.1	Spezifikation der Software-Architektur und der Software-Komponenten .....	269
5.3.2	Spezifikation des Datenmodells .....	274
5.3.3	Spezifikation des Verhaltensmodells mit Blockdiagrammen .....	274
5.3.4	Spezifikation des Verhaltensmodells mit Entscheidungstabellen .....	277
5.3.5	Spezifikation des Verhaltensmodells mit Zustandsautomaten .....	280
5.3.6	Spezifikation des Verhaltensmodells mit Programmiersprachen .....	285
5.3.7	Spezifikation des Echtzeitmodells .....	286
5.3.8	Validierung der Spezifikation durch Simulation und Rapid-Prototyping .....	287
5.4	Design und Implementierung von Software-Funktionen .....	296
5.4.1	Berücksichtigung der geforderten nichtfunktionalen Produkteigenschaften .....	297
5.4.2	Design und Implementierung von Algorithmen in Festpunkt- und Gleitpunktarithmetik .....	306
5.4.3	Design und Implementierung der Software-Architektur .....	322
5.4.4	Design und Implementierung des Datenmodells .....	326
5.4.5	Design und Implementierung des Verhaltensmodells .....	330
5.5	Integration und Test von Software-Funktionen .....	332
5.5.1	Software-in-the-Loop-Simulationen .....	334
5.5.2	Laborfahrzeuge und Prüfstände .....	335
5.5.3	Experimental-, Prototypen- und Serienfahrzeuge .....	342
5.5.4	Design und Automatisierung von Experimenten .....	343
5.6	Kalibrierung von Software-Funktionen .....	344
5.6.1	Arbeitsweisen bei der Offline- und Online-Kalibrierung .....	346
5.6.2	Software-Update durch Flash-Programmierung .....	347
5.6.3	Synchrones Messen von Signalen des Mikrocontrollers und der Instrumentierung .....	349
5.6.4	Auslesen und Auswerten von Onboard-Diagnosedaten .....	349
5.6.5	Offline-Verstellen von Parametern .....	350
5.6.6	Online-Verstellen von Parametern .....	351
5.6.7	Klassifizierung der Offboard-Schnittstellen für das Online-Verstellen .....	351
5.6.8	Management des CAL-RAM .....	357

---

5.6.9	Management der Parameter und Datenstände . . . . .	360
5.6.10	Design und Automatisierung von Experimenten . . . . .	362
<b>6</b>	<b>Methoden und Werkzeuge in Produktion und Service . . . . .</b>	<b>363</b>
6.1	Offboard-Diagnose. . . . .	364
6.2	Parametrierung von Software-Funktionen. . . . .	365
6.3	Software-Update durch Flash-Programmierung . . . . .	367
6.3.1	Löschen und Programmieren von Flash-Speichern. . . . .	368
6.3.2	Flash-Programmierung über die Offboard- Diagnoseschnittstelle . . . . .	368
6.3.3	Sicherheitsanforderungen . . . . .	370
6.3.4	Verfügbarkeitsanforderungen. . . . .	372
6.3.5	Auslagerung und Flash-Programmierung des Boot-Blocks . . . . .	372
6.4	Inbetriebnahme und Prüfung elektronischer Systeme . . . . .	375
<b>7</b>	<b>Zusammenfassung und Ausblick. . . . .</b>	<b>377</b>
	<b>Literatur. . . . .</b>	<b>381</b>
	<b>Stichwortverzeichnis. . . . .</b>	<b>387</b>

---

# Abkürzungsverzeichnis

ABS	Antiblockiersystem
ACC	Adaptive cruise control
AFS	Active front steering
API	Application programming interface
ALU	Arithmetic and logic unit
CAL-RAM	Calibration RAM
CAN	Controller area network
CCP	CAN calibration protocol
CI/CD	Continuous integration and delivery
CMMI <sup>®</sup>	Capability Maturity Model Integration <sup>®</sup>
COVESA	Connected Vehicle Systems Alliance
CPU	Central processing unit
CSMA	Carrier sense multiple access
CSMA/CA	CSMA/collision avoidance
CSMA/CD	CSMA/collision detection
D/A	Digital-analog conversion
DDS	Data distribution service
DevOps	Development and Operations
DMA	Direct-memory I/O access
DRAM	Dynamic RAM
DTC	Diagnostic trouble code
ECU	Electronic control unit
E/E	Electric/electronic vehicle architecture
EEPROM	Electrical EPROM
EMV	Elektromagnetische Verträglichkeit
EPROM	Erasable PROM
ESP	Elektronisches Stabilitätsprogramm
FIFO	First in, first out
FMEA	Failure mode and effects analysis

FODA	Feature-oriented Domain Analysis
FO	Fail-operational (system)
FR	Fail-reduced (system)
FS	Fail-safe (system)
FTA	Fault-tree analysis
HAL	Hardware abstraction layer
HiL	Hardware-in-the-loop (simulation)
HPC	High Performance Computer
I/O	Input/output
IP	Internet Protocol
LOV	Line of visibility (diagrams)
MIL	Malfunction indicator light
MMI	Man/machine interface
MQTT	Message queuing telemetry transport protocol
MTTF	Mean time to failure
MTTR	Mean time to repair
NV-RAM	Nonvolatile RAM
OBD	Onboard diagnostics
OEM	Original Equipment Manufacturer
OS	Operating system
OSEK	Offene Systeme und deren Schnittstellen für die Elektronik im Kraftfahrzeug
PI	Proportional-plus-integral (Regler)
PPM	Parts per million
PROM	Programmable ROM
QoS	Quality of Service
RAM	Random access memory
ROM	Read-only memory
RPC	Remote procedure call
RTE	Runtime Environment
SAT	Satisfiability
SBC	Sensotronic Brake Control
SDV	Software defined Vehicle
SiL	Software-in-the-loop (simulation)
SOA	Service-oriented architecture
SOD	Start of Development
SOME/IP	Scalable Service-Oriented Middleware over IP
SOP	Start of Production
SPICE	Software Process Improvement and Capability Determination
SRAM	Static RAM
TCP	Transmission Control Protocol

---

TCS	Traction control system
TDMA	Time division multiple access
TTF	Time to failure
UDP	User Datagram Protocol
UML™	Unified Modeling Language™
UV	Ultraviolet
VECU	Virtual Electronic Control Unit
VDA	Verband der Automobilindustrie
VSS	Vehicle Signal Specification (defined by COVESA)
WCET	Worst-case execution time
WCRT	Worst-case response time
XCP	Extended calibration protocol



Die Erfüllung steigender Kundenansprüche und strenger gesetzlicher Vorgaben hinsichtlich

- der Verringerung von Verbrauch, CO<sub>2</sub>- und Schadstoffemissionen sowie
- der Erhöhung von Fahrsicherheit und Fahrkomfort

ist untrennbar mit dem Einzug der Elektronik in modernen Kraftfahrzeugen verbunden.

Das Automobil ist dadurch zum technisch komplexesten Konsumgut geworden. Die Anforderungen an die Automobilelektronik unterscheiden sich jedoch wesentlich von anderen Bereichen der Konsumgüterelektronik. Insbesondere hervorzuheben sind:

- der Einsatz unter oft rauen und wechselnden Umgebungsbedingungen in Bezug auf Temperaturbereich, Feuchtigkeit, Erschütterungen oder hohe Anforderungen an die elektromagnetische Verträglichkeit (EMV),
- hohe Anforderungen an die Zuverlässigkeit und Verfügbarkeit,
- hohe Anforderungen an die Sicherheit im Sinne von Functional Safety und Cyber Security sowie
- vergleichsweise lange Produktlebenszyklen.

Diese Anforderungen müssen bei begrenzten Kosten, kurzer Entwicklungszeit und großer Variantenvielfalt in Produkte umgesetzt werden, die in sehr großen Stückzahlen hergestellt, zuverlässig betrieben und weltweit gewartet werden können. Unter diesen Randbedingungen stellt die Umsetzung der zahlreichen Anforderungen an elektronische Systeme von Fahrzeugen eine Entwicklungsaufgabe von hohem Schwierigkeitsgrad dar.

In der Entwicklung von Fahrzeugelektronik ist neben der Beherrschung der zunehmenden Komplexität vor allem ein konsequentes Qualitäts-, Risiko- und Kostenmanagement eine wichtige Voraussetzung für den erfolgreichen Abschluss von Projekten. Ein grundlegendes Verständnis der Anforderungen und Trends in der Fahrzeugentwicklung ist daher die wichtigste Voraussetzung, um geeignete Methoden für Entwicklung, Produktion und Service von elektronischen Systemen zu entwickeln und durch praxistaugliche Standards und Werkzeuge unterstützen zu können. In diesem Übersichts Kapitel erfolgt ausgehend von einer Analyse der aktuellen Situation eine Darstellung der zukünftigen Perspektiven und Herausforderungen. Neben der Organisation der interdisziplinären und firmenübergreifenden Zusammenarbeit müssen auch viele Zielkonflikte gelöst werden.

Nach einem Überblick über die elektronischen Systeme des Fahrzeugs und deren Funktionen folgt eine Einführung in Vorgehensweisen zur Entwicklung von elektronischen Systemen und Software für Fahrzeuge. Dabei müssen zahlreiche Wechselwirkungen zwischen der Systementwicklung in der Automobilindustrie (engl. Automotive Systems Engineering) und der Software-Entwicklung (engl. Automotive Software Engineering) beachtet werden. Abschließend werden modellbasierte Entwicklungsmethoden vorgestellt, welche die verschiedenen Aspekte berücksichtigen.

In den weiteren Kapiteln des Buches erfolgt eine ausführliche Behandlung von Grundlagen, Prozessen, Methoden und Werkzeugen für Entwicklung, Produktion und Service von Software für die elektronischen Systeme von Fahrzeugen. Der Schwerpunkt liegt dabei auf den Fahrzeugsystemen oder funktionalen Domänen Antriebsstrang, Fahrwerk, Karosserie und Fahrerassistenz. Der Bereich der Infotainment-Systeme wird in einer Übersicht dargestellt und berücksichtigt, aber nicht ausführlicher behandelt.

---

## **1.1 Das System Fahrer-Fahrzeug-Umwelt**

Das Ziel jeder Entwicklung ist die Fertigstellung einer neuen oder die Verbesserung einer vorhandenen Funktion des Fahrzeugs. Unter Funktionen werden dabei alle Funktionsmerkmale des Fahrzeugs verstanden. Diese Funktionen werden vom Benutzer, etwa dem Fahrer oder Passagier des Fahrzeugs, direkt oder indirekt wahrgenommen und stellen einen Wert oder Nutzen für ihn dar und sind erlebbar (engl. User Experience).

Die technische Realisierung einer Funktion, ob es sich also letztendlich um ein mechanisches, hydraulisches, elektrisches oder elektronisches System im Fahrzeug oder auch im IT-Backend handelt, hat dabei zunächst eine untergeordnete Bedeutung.

Elektronische, programmierbare Komponenten in Kombination mit mechanischen, elektrischen oder hydraulischen Bauteilen für Sensoren und Aktuatoren bieten jedoch bei der technischen Realisierung viele Vorteile, etwa in Bezug auf die Freiheitsgrade bei der Realisierung, die erreichbare Zuverlässigkeit, das Gewicht, den benötigten Bauraum oder die Kosten. Wie keine andere Technologie ist deshalb heute die Elektronik – Hardware und Software – die Schlüsseltechnologie zur Realisierung vieler Innovationen im Fahrzeugbau.

### 1.1.1 Physikalisches und digitales Anwendererlebnis

Fast alle Funktionen des Fahrzeugs werden mittlerweile elektronisch gesteuert, geregelt, diagnostiziert oder überwacht (engl.: control, diagnostics and monitoring). Diese Funktionen beeinflussen über Sensoren und Aktuatoren den physikalischen Zustand des Fahrzeugs; sie führen zu einem physikalischen Anwendererlebnis (engl. Physical User Experience) [126].

Daneben ist eine neue Klasse von Funktionen am Entstehen, die zu einem reinen digitalen Anwendererlebnis führt (engl. Digital User Experience). Diese Funktionen haben keine eigene Sensorik und Aktuatorik, sondern sind reine Software-Funktionen – wie beispielsweise die Suche und Buchung einer naheliegenden Ladesäule, die Suche und Buchung eines naheliegenden Parkplatzes, das automatisierte Bezahlen von Parkgebühren oder auch die nutzerprofilabhängige Versicherung des Fahrzeugs [126]. In den folgenden Kapiteln liegt der Fokus zunächst auf den physikalisch erlebbaren Fahrzeugfunktionen, die durch elektronische Systeme realisiert werden.

### 1.1.2 Aufbau und Wirkungsweise elektronischer Systeme

Auf Aufbau und Wirkungsweise elektronischer Systeme des Fahrzeugs soll daher am Beispiel des elektrohydraulischen Bremssystems näher eingegangen werden.

#### Beispiel

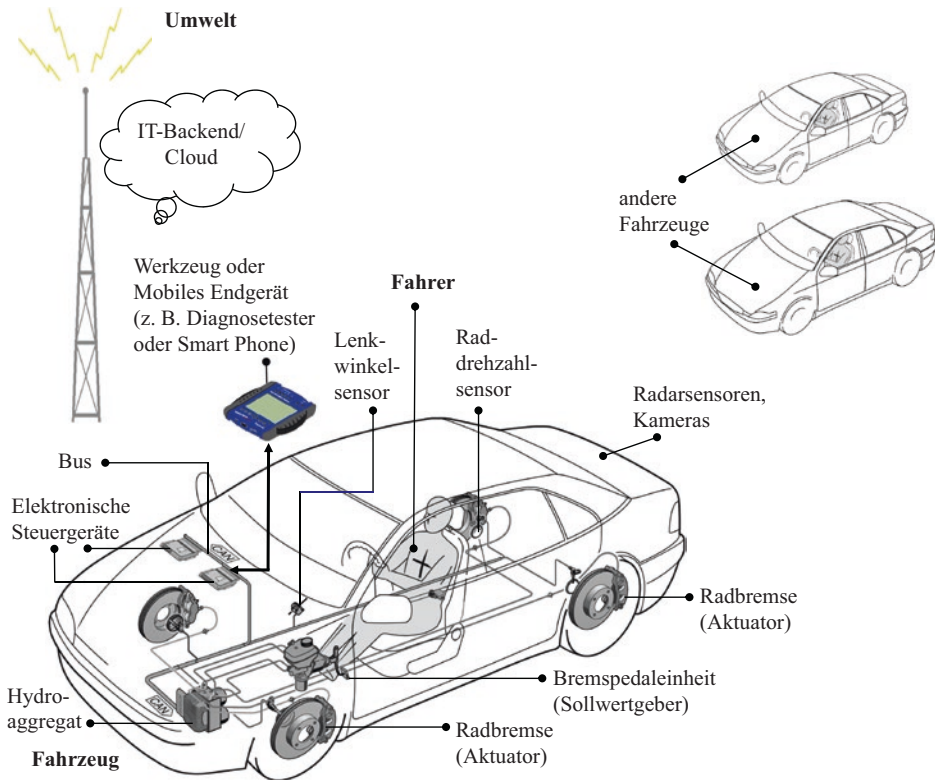
##### **Aufbau des elektrohydraulischen Bremssystems [1]**

In Abb. 1.1 ist der Aufbau des elektrohydraulischen Bremssystems von Bosch [1] dargestellt. Die elektrohydraulische Bremse vereint die Funktionen des Bremskraftverstärkers, des Antiblockiersystems (ABS) und der Fahrdynamikregelung, auch elektronisches Stabilitätsprogramm (ESP) genannt. ◀

Die mechanische Betätigung des Bremspedals durch den Fahrer wird in der Bremspedaleinheit erfasst und elektrisch an das so genannte elektronische Steuergerät übertragen. In diesem Steuergerät werden unter Verwendung dieses Sollwertes und verschiedener Sensorsignale, wie z. B. dem Lenkwinkelsignal oder den Raddrehzahlensignalen, Ausgangsgrößen berechnet, die wiederum elektrisch zum Hydroaggregat übertragen werden und dort durch Druckmodulation in Stellgrößen für die Radbremsen umgesetzt werden. Über die Radbremsen wird das Fahrverhalten des Fahrzeugs, die so genannte Regel- oder Steuerstrecke, beeinflusst. Die Radbremsen werden daher als Aktuatoren bezeichnet.

Das Steuergerät kommuniziert über einen Bus, etwa über CAN [2], mit anderen Steuergeräten des Fahrzeugs. Dadurch können Funktionen realisiert werden, die über die





**Abb. 1.1** Aufbau der elektrohydraulischen Bremse. (Robert Bosch GmbH [1])

bisher genannten Funktionen hinausgehen. Ein Beispiel dafür ist die Antriebsschlupfregelung (ASR), die eine übergreifende Funktion zwischen Motorsteuerung und Bremssystem darstellt.

Der am Beispiel der elektrohydraulischen Bremse dargestellte Systemaufbau ist typisch für alle elektronischen Steuer-, Regelungs- und Überwachungssysteme des Fahrzeugs. Im Allgemeinen kann zwischen den folgenden Komponenten des Fahrzeugs unterschieden werden, die zur Realisierung notwendig sind: Sollwertgeber, Sensoren, Aktuatoren und elektronische Steuergeräte. Diese Komponenten steuern die Strecke bzw. regeln die Regelstrecke. Die elektronischen Steuergeräte sind miteinander vernetzt und können so durch Kommunikation Daten untereinander austauschen.

Fahrer und Umwelt können das Verhalten des Fahrzeugs beeinflussen und sind Komponenten des übergeordneten Systems Fahrer-Fahrzeug-Umwelt.

Für sich allein genommen ist ein elektronisches Steuergerät also lediglich ein Mittel zum Zweck. Alleine stellt es für den Fahrzeugbenutzer keinen Wert dar.

Erst ein vollständiges elektronisches System aus Steuergeräten, Sollwertgebern, Sensoren und Aktuatoren beeinflusst oder überwacht die Strecke und erfüllt so die

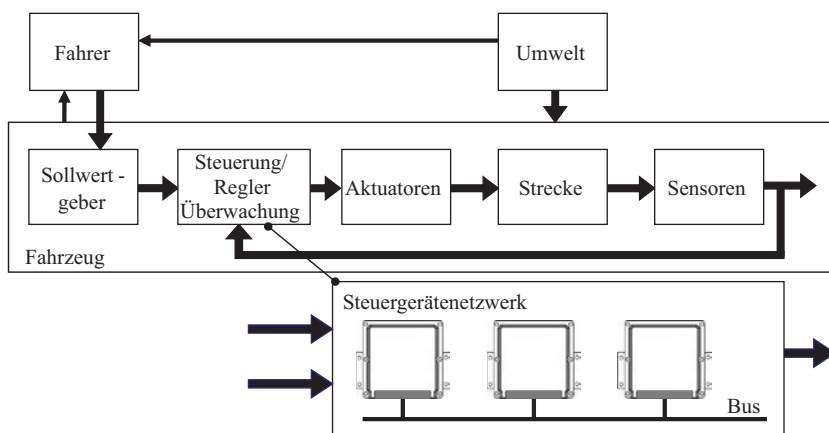
Benutzererwartungen. In vielen Fällen ist die Realisierung mittels Elektronik für den Fahrzeugbenutzer nicht einmal sichtbar, wenn es sich – wie häufig – um so genannte eingebettete Systeme handelt.

Steuerungs-, Regelungs- und Überwachungssysteme des Fahrzeugs können übersichtlich in Form eines Blockdiagramms gezeichnet werden, wie in Abb. 1.2 dargestellt. Die Komponenten werden dabei als Blöcke und die zwischen ihnen bestehenden Signalflüsse als Pfeile dargestellt. Eine Einführung in die Grundlagen und Begriffe der Steuerungs-, Regelungs- und Überwachungstechnik erfolgt in den Abschn. 2.1 und 2.6.

Wie in Abb. 1.2 ersichtlich, können zwischen den Komponenten Fahrer, Fahrzeug und Umwelt zahlreiche Signalflüsse bestehen. Der Fahrer steht in dieser Darstellung stellvertretend auch für alle anderen Benutzer einer Fahrzeugfunktion, etwa für weitere Passagiere.

Zur Umwelt zählen auch andere Fahrzeuge oder elektronische Systeme in der Umgebung des Fahrzeugs oder im IT-Backend – auch Werkzeuge, wie Diagnosetester, die in der Servicewerkstatt mit den elektronischen Systemen des Fahrzeugs verbunden werden (Abb. 1.1).

Neue Technologien zum Informationsaustausch zwischen Fahrer und Fahrzeug, zwischen Fahrer und Umwelt sowie zwischen Fahrzeug und Umwelt ermöglichen viele innovative Funktionen – etwa durch eine Vernetzung über die Fahrzeuggrenze hinweg mit drahtlosen Übertragungssystemen „over the air“. Daran kann der Übergang zu einer neuen Klasse von Systemen erkannt werden – zu Systemen, die eine Vernetzung von Fahrer und Fahrzeug mit der Umwelt nutzen und damit die Grundlage bilden für viele den Fahrer unterstützende Funktionen wie Fahrerassistenzsysteme. Insbesondere auch im Bereich der Infotainment-Systeme wurden viele Funktionen eingeführt, die erst durch diese Vernetzung von Fahrzeug und Umwelt möglich wurden. Ein Beispiel ist



**Abb. 1.2** Blockdiagramm zur Darstellung von Steuerungs-, Regelungs- und Überwachungssystemen

die dynamische Navigation, die Informationen der Umwelt – wie Staumeldungen oder Umweltinformationen anderer Verkehrsteilnehmer – bei der Routenberechnung berücksichtigt.

Auch der Bereich der Schnittstellen zwischen Fahrer bzw. Passagieren und Fahrzeug, die so genannten Benutzerschnittstellen, ist von vielen Innovationen geprägt. So basieren Bedien- und Anzeigesysteme inzwischen auf Sprachein- und Sprachausgabesystemen und modernen display- oder gestengestützten Bedienkonzepten.

Der Begriff der Vernetzung soll deshalb in diesem Buch nicht auf die elektronischen Steuergeräte des Fahrzeugs beschränkt, sondern weiter gefasst werden. Hilfreich ist dazu die Unterscheidung zwischen der Vernetzung der Systeme innerhalb des Fahrzeugs und der Vernetzung über die Fahrzeuggrenze hinweg.

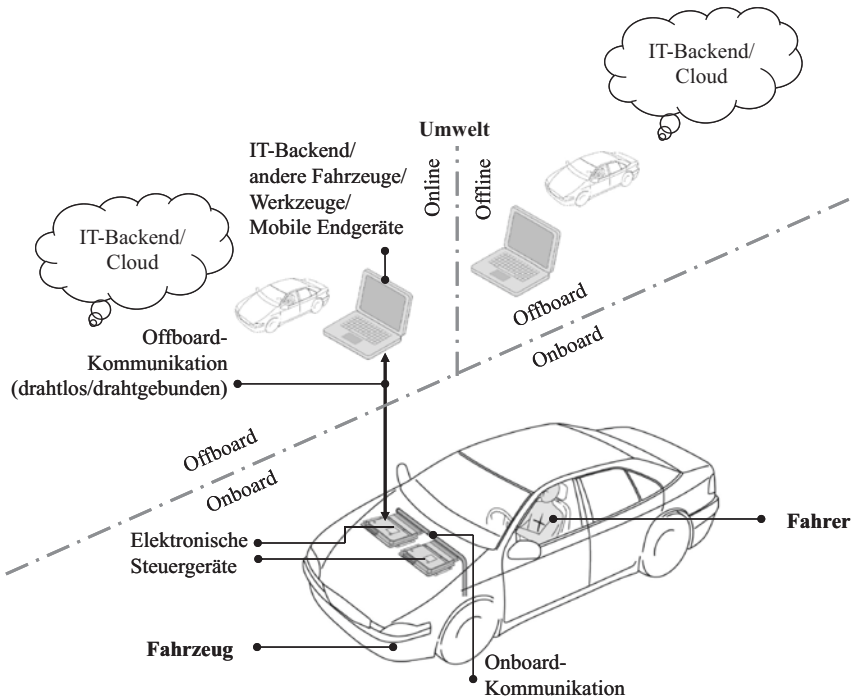
### **1.1.3 Elektronische Systeme des Fahrzeugs und der Umwelt**

In den folgenden Kapiteln wird die Kommunikation zwischen elektronischen Systemen des Fahrzeugs als Onboard-Kommunikation bezeichnet; die Kommunikation zwischen Systemen des Fahrzeugs und Systemen der Umwelt als Offboard-Kommunikation. Bei den Schnittstellen der elektronischen Systeme des Fahrzeugs wird entsprechend zwischen Onboard- und Offboard-Schnittstellen unterschieden. Einen Überblick zeigt Abb. 1.3.

Bei den Funktionen der elektronischen Systeme wird zwischen Funktionen, die von Systemen des Fahrzeugs ausgeführt werden, den so genannten Onboard-Funktionen, und Funktionen, die von Systemen der Umwelt ausgeführt werden, den so genannten Offboard-Funktionen, unterschieden. Auch die Aufteilung von Funktionen in Teilfunktionen und deren verteilte Realisierung durch Systeme im Fahrzeug und außerhalb des Fahrzeugs, z. B. im IT-Backend, ist möglich.

Ein weiteres Unterscheidungsmerkmal ist der Zeitpunkt der Ausführung einer Funktion durch ein System der Umwelt in Bezug zur Ausführung einer Funktion eines Fahrzeugsystems. Hier wird zwischen der synchronisierten Ausführung von Funktionen, auch Online-Ausführung genannt, und der nicht synchronisierten Ausführung einer Funktion, auch als Offline-Ausführung bezeichnet, unterschieden.

Diese beiden Unterscheidungskriterien onboard/offboard und online/offline betreffen schon seit vielen Jahren die Systeme zur Diagnose von Fahrzeugfunktionen, so etwa die Unterscheidung zwischen Onboard- und Offboard-Diagnose, und in ähnlicher Weise auch viele in der Entwicklung elektronischer Systeme eingesetzte Entwurfsmethoden und Werkzeuge. Mit der Connectivity des Fahrzeugs zum IT-Backend stehen diese Freiheitsgrade auch für die Entwicklung von Funktionen zur Verfügung.



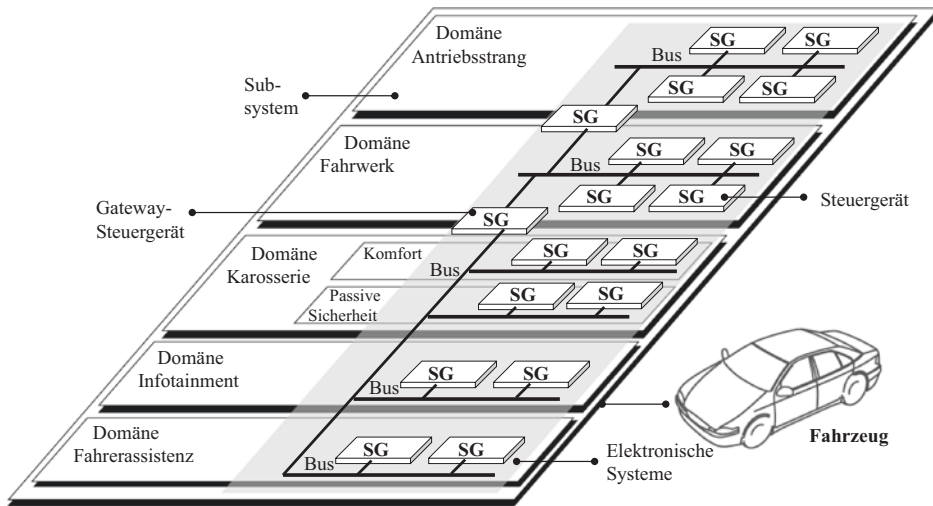
**Abb. 1.3** Elektronische Systeme des Fahrzeugs und der Umwelt

## 1.2 Überblick über die elektronischen Systeme des Fahrzeugs

Zunächst sollen die verschiedenen elektronischen Systeme eines Fahrzeugs in einem Überblick dargestellt werden.

Zu Beginn des Einsatzes von Elektronik im Fahrzeug arbeiteten die verschiedenen elektronischen Steuergeräte weitgehend autonom und isoliert ohne gegenseitige Wechselwirkungen. Die Funktionen konnten deshalb eindeutig einem Steuergerät und meist eindeutig den funktionalen Domänen wie Antriebsstrang, Fahrwerk, Karosserie, Infotainment und Fahrerassistenz zugeordnet werden (Abb. 1.4).

Die Motor- und Getriebesteuerung werden z. B. als klassische Systeme dem Antriebsstrang zugeordnet, das Anti-Blockier-System (ABS) dem Fahrwerksbereich, die Heizungs- und Klimatisierungssteuerung, Zentralverriegelung, Sitz- und Spiegelverstellung dem Komfortbereich der Karosseriesysteme. Die Airbag- und Rückhaltesysteme tragen zur Erhöhung der Sicherheit der Fahrzeuginsassen im Falle eines Unfalls bei und zählen zu den passiven Sicherheitssystemen. Radio, Navigationssystem oder Telefon gehören zu



**Abb. 1.4** Zuordnung der elektronischen Steuergeräte zu den funktionalen Domänen des Fahrzeugs

den Infotainment-Systemen. Das Adaptive Cruise Control System (ACC) oder der Spurhalteassistent gehört zu den Fahrerassistenzsystemen.

Die anhaltenden Technologie- und Leistungssprünge in der Hardware und in den Kommunikationstechnologien erlauben die Realisierung zahlreicher neuer und immer leistungsfähigerer Funktionen des Fahrzeugs durch Software. Diese Funktionen werden als Software-Funktionen bezeichnet.

Mit der Vernetzung der elektronischen Systeme im zweiten Schritt etwa ab Beginn der 1990er Jahre durch die Einführung leistungsfähiger Bussysteme im Fahrzeug, wie CAN [2], wurden die Realisierung neuer, übergeordneter Software-Funktionen sowie Kosteneinsparungen möglich. Kosteneinsparungen sind beispielsweise durch die Verwendung von Sensorsignalen in mehreren Systemen ohne aufwändigen Verkabelungsaufwand realisierbar.

Wirken diese übergeordneten Software-Funktionen weiter innerhalb der funktionalen Domänen, so wird dieser ganzheitliche Ansatz auch als integriertes Antriebsstrang-, integriertes Fahrwerks-, integriertes Karosserie- oder integriertes Sicherheitsmanagement bezeichnet. In solchen verteilten und vernetzten Systemen können Software-Funktionen oft nicht mehr einem einzigen Steuergerät zugeordnet werden. Software-Funktionen werden dann in Teilfunktionen aufgeteilt, die dann in verschiedenen Steuergeräten realisiert werden.

Wirken die übergeordneten Software-Funktionen über die Grenzen einer funktionalen Domäne hinaus im Netzwerk, dann können sie auch nicht mehr einer bestimmten Domäne zugeordnet werden. Wie bereits im einführenden Beispiel angesprochen, ist die Antriebsschlupfregelung eine antriebsstrang- und fahrwerkübergreifende Funktion. Aber auch viele Fahrerassistenzsysteme, wie die verkehrsflussangepasste Geschwindig-

keitsregelung (ACC), fallen in diese Kategorie. Ebenso gibt es im Komfort- und passiven Sicherheitsbereich viele domänenübergreifende Funktionen. Ein Beispiel ist das Fahrzeugzugangssystem einschließlich der Schließanlage und der Diebstahlsicherung. Aus Funktionssicht werden die Übergänge zwischen den Domänen damit fließend. Die Voraussetzung zur Realisierung derartiger Funktionen ist die Vernetzung der Steuergeräte über Domänengrenzen hinweg. Dies erfolgte in der Vergangenheit häufig durch so genannte Gateway-Steuergeräte (Abb. 1.4). Derartige E/E-Architekturen werden daher auch als Domänenarchitekturen bezeichnet.

Ist die systemübergreifende Verwendung von Sensorsignalen durch wohl definierte Systemschnittstellen problemlos umsetzbar, so stellt der „konkurrierende“ Zugriff mehrerer Software-Funktionen auf dieselben Aktuatoren jedoch größere Herausforderungen dar – insbesondere an eine geeignete Spezifikationsmethodik, die auch die Prioritäten der verschiedenen Funktionen definiert. Dabei muss die Definition eindeutiger Daten- und Auftragschnittstellen zwischen den verschiedenen Funktionen und Systemen gleichermaßen berücksichtigt werden. Die verschiedenen Aufträge etwa an Aktuatoren müssen koordiniert werden. Einen Beitrag zu diesem Themengebiet liefert beispielsweise AUTOSAR [3].

Nach einem Überblick über die elektronischen Systeme des Antriebsstrangs, des Fahrwerks, der Karosserie und der Fahrerassistenzsysteme werden in diesem Abschnitt auch die Infotainment-Systeme dargestellt. Diese werden im Rahmen dieses Buches zwar nicht weiter berücksichtigt, eine Übersicht ist dennoch zur Abgrenzung gegenüber den anderen Anwendungsbereichen hilfreich. Dabei erfolgt eine Klassifizierung der Systeme nach typischen Merkmalen wie Benutzerschnittstellen und Sollwertgebern, Sensoren und Aktuatoren, Software-Funktionen, Bauraum sowie Varianten und Skalierbarkeit. Auch aktuell erkennbare Trends werden berücksichtigt.

Bei der technischen Realisierung müssen in vielen Fällen eine Reihe von gesetzlichen Vorgaben berücksichtigt werden. So stehen etwa im Bereich des Antriebsstrangs, insbesondere bei den Software-Funktionen der Steuergeräte für Verbrennungsmotoren, häufig Vorgaben bezüglich Kraftstoffverbrauchs, CO<sub>2</sub>- und Schadstoffemissionen im Vordergrund. Bei den Fahrwerks- und Karosseriefunktionen spielen dagegen Sicherheits- und Komfortanforderungen eine große Rolle.

Dieser Abschnitt beschränkt sich auf einen Überblick über die elektronischen Fahrzeugsysteme und ihre Funktionen. Einzelne Aspekte werden anhand von Beispielen in den weiteren Kapiteln des Buches wieder aufgegriffen. Für eine ausführliche Darstellung wird auf die umfangreiche Fachliteratur, z. B. [4], verwiesen.

### 1.2.1 Elektronische Systeme des Antriebsstrangs

Der Antriebsstrang eines Fahrzeuges umfasst folgende Aggregate und Komponenten:

Antrieb – also Elektromotor [24, 63], Verbrennungsmotor, Hybridantrieb, Batterie, Brennstoffzelle –,