
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.

Fabian Wolf

Fahrzeuginformatik

Eine Einführung in die Software- und
Elektronikentwicklung aus der Praxis der
Automobilindustrie

 Springer Vieweg

Fabian Wolf
Clausthal University of Technology
Clausthal-Zellerfeld, Deutschland

ATZ/MTZ-Fachbuch
ISBN 978-3-658-21223-0
<https://doi.org/10.1007/978-3-658-21224-7>

ISBN 978-3-658-21224-7 (eBook)

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

Springer Vieweg

© Springer Fachmedien Wiesbaden GmbH, ein Teil von Springer Nature 2018

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, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften. 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.

Verantwortlich im Verlag: Markus Braun

Coverdesigner: Die Texte wurden eingesprochen von: Lee Rychter, Berlin

Gedruckt auf säurefreiem und chlorfrei gebleichtem Papier

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

*Für Pia, Maja, meine Familie und Freunde: Diese
Themen beschäftigen mich, wenn ich nicht bei
Euch bin.*

Vorwort

Der Einfluss der Digitalisierung aus dem Consumer-Bereich und der globalen Vernetzung, der Trend zum autonomen Fahren sowie die ökologische Notwendigkeit für neue Antriebskonzepte stellen die Automobilindustrie vor große Herausforderungen. Neue Mobilitätskonzepte und Produktideen implizieren einen radikalen Wandel der etablierten Konzepte für Technik und Vorgehensweisen, benötigen jedoch zuverlässige Plattformen zur nachhaltigen Umsetzung der Kundenwünsche. Hierbei muss die Evolution der gesamten Technologie des Fahrzeugs, das nach wie vor die Basis individueller Mobilität sein wird, betrachtet werden.

Der Weg zur vollständig elektronischen oder sogar digitalen Elektromobilität wird durch eine stetige Weiterentwicklung der aktuellen Elektronik und Antriebskonzepte im Fahrzeug geprägt sein. Keine verantwortungsbewusste Industrie, Volkswirtschaft oder Gesellschaft wird nachhaltig in der Lage sein, sowohl technisch als auch ökonomisch und damit letztendlich demografisch sozialverträglich der Geschwindigkeit der Produktvisionen zu folgen. Das teilweise notwendige Abschaffen etablierter, verankerter Konzepte erfordert einen praktikablen Ersatz und stabile Übergangsphasen.

Die „Fahrzeuginformatik“ spielt als Schnittstelle zwischen Produktvision und deren konkreter Umsetzung eine Schlüsselrolle in der Elektromobilität. Damit ist dieser Begriff rein sprachlich das deutsche Pendant zum „Automotive Software Engineering“ und definiert darüber hinaus ganzheitlich eine neue Disziplin in der Entwicklung der Fahrzeuggenerationen von morgen.



Einordnung des Begriffs Fahrzeuginformatik

Die Grundlagen für die Produkte und Mobilität von morgen sind in aktuellen Fahrzeugen zum Teil bereits vorhanden und können im Sinne einer pragmatischen Umsetzung neuer Produktideen weiterentwickelt werden. Fast alle Funktionen des Fahrzeugs werden bereits heute durch Software gesteuert, geregelt oder überwacht. Die vorhandenen Freiheitsgrade müssen jedoch Randbedingungen wie hohe Qualitäts- und Sicherheitsan-

forderungen, Standards, Gesetzgebung, kurze Entwicklungszeiten und die zunehmende Nachfrage nach einer Erweiterung der Fahrzeugfunktionen durch Softwareaktualisierung oder Updates im Lebenszyklus berücksichtigen.

Dieses Fach- und Lehrbuch enthält die Basis für die Umsetzung neuer Trends, wissenschaftliche Arbeiten und ökonomische Entscheidungen. Diese Basis sind unter anderem die heutigen Grundlagen und Sicht der Praxis zu Elektronik und Software im Fahrzeug, Elektronikarchitektur, Softwareentwicklung und Test sowie die zugehörigen Prozesse für zulassungsrelevante Systeme werden vorgestellt.

Die Zielgruppen sind in erster Linie angehende Fachkräfte der Mechatronik, Studierende und Berufseinsteiger im Bereich der Fahrzeugelektronik und Fahrzeuginformatik sowie sämtliche Mitarbeiter der Automobilindustrie, die sich auf den anstehenden Wandel nachhaltig vorbereiten wollen, um Entscheidungen auf der Basis von Fakten zu treffen. Damit empfiehlt sich dieses Werk auch als fachliche Basis für Entscheider außerhalb der Automobilindustrie, die den digitalen Wandel nachhaltig und fundiert vorantreiben oder diskutieren wollen.

Der Autor greift auf seine inhaltliche, organisatorische und leitende Erfahrung im Bereich der verteilten Entwicklung von Software für Motorsteuerungen sowie Lenkungselektronik zurück. Dazu kommt die Tätigkeit in internationalen Gremien sowie die über Jahre gereifte didaktische Aufbereitung der Inhalte in Form seiner Vorlesung „Fahrzeuginformatik“.

Abgrenzungen

Der Fokus ist auf ein übergreifendes Lehrbuch und den Einstieg im Sinne einer Übersicht zu den Themenbereichen der Fahrzeuginformatik gesetzt. Es handelt sich weder um eine wissenschaftliche Arbeit, noch sollen die Lehrinhalte in voller fachlicher Tiefe erfasst werden. Darum wird der Inhalt nach didaktischen und praktischen Gesichtspunkten vermittelt. Die vorgestellten Themen werden in einem Grad detailliert, die einer Übersicht aus der Praxis angemessen sind. Wichtige Inhalte werden im Sinne der Geschlossenheit der Kapitel teilweise wiederholt. Bewertungen stellen die persönliche Sicht des Autors dar, die keine Rückschlüsse auf seinen aktuellen Arbeitgeber oder seinen Lehrauftrag zulassen.

Literaturangaben werden nur bei übernommenen Darstellungen, Zitaten oder zu Schlüsselquellen gegeben, da die Anzahl der Verweise pro Kapitel aufgrund des Übersichtscharakters des Buchs immens hoch wäre. Die Themen und Literatur entwickeln sich stetig weiter (z. B. Wikipedia) und können dort global recherchiert werden. Durch die damit kompakte und strukturierte Literaturliste besteht die Möglichkeit, die einzelnen Themen mit begrenztem Aufwand spezifisch zu vertiefen.

In diesem Buch wird der sprachlichen Einfachheit halber durchgehend die männliche Form der Rollenbeschreibungen verwendet. Dies drückt keine geringere Wertschätzung gegenüber den Damen und insbesondere interessierten Leserinnen dieses Buchs aus.

Anglizismen sind in der Fahrzeuginformatik nicht auszuschließen und werden nach Einschätzung des Autors dort verwendet, wo sie zur Klärung der Thematik beitragen können oder dem Sprachgebrauch entsprechen. Die Silbentrennung zusammengesetzter Begriffe ist nicht konsequent einhaltbar.

Die Kapitel werden jeweils mit einer Zusammenfassung und einem Teil abgeschlossen, in dem die wichtigsten inhaltlichen Grundsatzthemen noch einmal reflektiert oder hinterfragt werden. Dies soll der Nachhaltigkeit der Lerninhalte dienen, damit der Aufbau der Inhalte der Kapitel aufeinander gestützt und damit die Verständlichkeit des gesamten Buches sichergestellt wird. Es werden keine Musterlösungen zu den Fragen gegeben, die Antworten sind im jeweiligen Kapitel zu finden.

Danksagungen

In erster Linie möchte ich meinem Arbeitgeber Volkswagen für die Möglichkeit danken, dieses Buch zu schreiben. Dazu gehören der organisatorische Rahmen und die abwechslungsreichen Inhalte meiner Tätigkeit. Ein weiterer Dank gilt der Technischen Universität Clausthal für die Möglichkeit der Vorlesung der Inhalte dieses Buches und die damit verbundene Honorarprofessur.

Neben dem Gastbeitrag zum Variantenmanagement von Christoph Seidl haben etliche Personen bewusst oder unbewusst beim Zusammentragen der Inhalte dieses Buchs geholfen. Hervorzuheben sind Herr Marc Neubauer für die jahrelange Betreuung der Vorlesung und einige Darstellungen, die Buchstabensuppe von Udo Hallmann als Erläuterung für Modulbaukästen und die Umsetzung des Messverfahrens für Rechenzeiten von Andreas Schulze. Weiterhin danke ich etlichen Urhebern, stellvertretend den Herren Reif, Schäufele und Hofmann, für die Genehmigung zur Verwendung von Teilen ihrer Werke. Diese sind gekennzeichnet und als weiterführende Literatur empfohlen.

Eine große Hilfe war das permanente Hinterfragen aller Aspekte meiner Arbeit seit der Zeit meiner Promotion über die Tätigkeit bei Volkswagen und die Vorlesung bis zum Schreiben dieses Buchs. Dazu gehört auch meine Familie, die in den verschiedenen Phasen auf sehr unterschiedliche Weise positiv dazu beigetragen hat.

Braunschweig, April 2018

Einleitung

Die Kundenanforderungen, Trends und gesetzlichen Vorgaben zur nachhaltigen Mobilität erfordern neue Konzepte für die Elektromobilität, Abgasnachbehandlung, Digitalisierung, dem autonomen Fahren und neuen Antriebskonzepten. Dazu entwickelt die Automobilindustrie neue Technologien wie Vernetzung, Bedienkonzepte, Elektromotoren und Energiespeicher. Neben diesen Schlüsseltechnologien stellen die allgemeine, produktunabhängige Elektronik und Software des Fahrzeugs einen wesentlichen Beitrag zur Steuerung und damit dem effektiven und effizienten Einsatz dieser Innovationen dar.

Solange sich Mobilität als physikalische Bewegung und Funktionen als Software auf elektronischen Steuergeräten darstellen, müssen die Grundsätze der Fahrdynamik und Elektronik sowie die damit zusammenhängenden Grenzen, die Eigenschaften von Materialien und mögliche Ausfälle berücksichtigt werden. Auch die Umsetzung gesetzlicher Vorgaben, z. B. zur übergangsweisen Weiterverwendung der Verbrennungsmotoren, muss sich bei allem Enthusiasmus auf der einen oder Stringenz der Vorgaben auf der anderen Seite entlang der nicht diskutablen oder modifizierbaren Naturwissenschaft und Physik sowie Verfügbarkeit von Ressourcen bewegen. Das gilt auch für die Visionen zur Digitalisierung im Fahrzeug oder die Strategien zur Elektromobilität.

Da bei allen politischen Forderungen, Vorgaben und ideologischen Wünschen die Automobilindustrie als wesentlicher volkswirtschaftlicher Bestandteil Deutschlands in Bezug auf Beschäftigung effizient arbeiten muss, stellt die Einhaltung der Vorgaben immer einen Kompromiss aus verwendeter Technik und gegebenenfalls subventioniertem oder sanktioniertem Verkaufspreis sowie Haltungskosten für den Kunden dar. Die Verletzung dieses Gleichgewichts kann zu nicht statthaften Abweichungen führen, wenn die Interessen einer der Seiten unangemessen beschnitten werden und damit Existenzen bedrohen.

Im Gegensatz zu den kurzen Lebenszyklen für Elektronik und Software im Bereich der IT-Systeme auf Arbeitsplatzrechnern oder Smartphones als Endprodukte sind die Lebenszyklen der Software im Fahrzeug oft an die Entwicklungsprozesse der Fahrzeugplattformen gekoppelt. Diese mehrjährigen Zyklen sind einerseits durch die Komplexität des Gesamtprodukts, unter anderem aber auch durch den Kunden vorgegeben. Dieser wechselt das Fahrzeug in der Regel seltener als sein Smartphone. Die Kundenerwartung an die Zuverlässigkeit des Fahrzeugs ist ebenfalls höher als an reine Softwareprodukte.

Die langen Entwicklungszyklen limitieren auf der einen Seite die Entwicklungsgeschwindigkeit für neue Produkttrends oder Kundenwünsche, stellen auf der anderen Seite allerdings die abgesicherte Funktionalität mit sämtlichen Seitenaspekten und für alle Varianten im Gesamtsystem Fahrzeug sicher. Die evolutionäre Weiterentwicklung neuer Funktionen verlangt die Kenntnis der Gesamtarchitektur des Fahrzeugs als System aus Mechanik, Elektronik und Software. Unter weiterer Berücksichtigung der Umwelteinflüsse und des Fahrers ergibt sich ein komplexes mechatronisches Regelsystem.

Ebenso kann es bei der Umsetzung von Kundenwünschen und Vorgaben in Software Fehler, Missverständnisse und menschliches Versagen geben, was entsprechende Prozesse zur Absicherung verlangt. In diesem Zusammenhang können Mechanik, Elektronik und Software nur begrenzt getrennt betrachtet werden. In Software implementierte Fahrzeugfunktionen werden als Eigenschaft des Gesamtprodukts im System entwickelt, freigegeben und verkauft.

Der immer stärker nachgefragte reine Verkauf von Funktionen in Form einer Softwareaktualisierung des bestehenden Fahrzeugs ist technisch eine Erweiterung der aktuellen gängigen Praxis der Reparatur des Fahrzeugs durch Software mittels sogenannter Updates. Er unterliegt damit den gleichen Rahmenbedingungen. Diese müssen genauso ohne eine physikalische Änderung in die Architektur des Fahrzeugs passen und in diesem Zusammenhang für alle Varianten entwickelt, getestet sowie freigegeben werden.

Um den Anforderungen dieses komplexen Umfelds gerecht zu werden, ist dieses Buch in Abschnitten strukturiert, die immer wieder ineinander greifen und aufgrund des komplexen Systems Fahrzeug nicht isoliert betrachtet werden können. Der Fokus wird auf Elektronik und Software gesetzt, die zugehörige Mechanik wird gegebenenfalls am Rande behandelt, wenn sie im Systemkontext eine Rolle spielt. Im Lebenszyklus der Fahrzeugelektronik wird der Schwerpunkt auf die Entwicklung gesetzt. Industrialisierung, Produktion und Betrieb werden nur am Rande betrachtet.

Als entscheidende Basis gilt die in Kap. 1 vorgestellte Fahrzeugelektronik. Neben einer Erläuterung der Gesamtarchitektur und des mechatronischen Regelkreises Fahrzeug werden Sensoren eingeführt. Dabei wird auf die physikalischen Grundlagen und die Begriffe der Quantisierung, Wandlerketten und Sensorpartitionierung eingegangen. Hierzu werden Beispiele aus der Praxis erläutert. Es erfolgt eine Einführung der Aktoren und Bussysteme mit deren prominenten Vertretern sowie die zugehörigen Zugriffsverfahren. Prozessoren, Speicher, Controller und Schaltungskonzepte werden erläutert. Diese werden zu Steuergeräten zusammengefasst und es wird auf den Begriff der Hardwarebeschreibungssprachen mit Beispielen eingegangen.

In Kap. 2 zur Software im Fahrzeug werden verschiedene Topologien mit Vor- und Nachteilen vorgestellt sowie mit einer Referenz zur Fahrzeug-Gesamtarchitektur und Softwarearchitektur die Begriffe der Domänen und der Funktionsabbildung erläutert, also der Umsetzung gewünschter Eigenschaften durch die Architektur des Fahrzeugs. Es werden Treiber, Basissoftware und Echtzeitbetriebssysteme eingeführt. Neben der Eigendiagnose im Fahrzeug wird die Werkstattdiagnose und Softwareaktualisierung zur Fehlerbehebung oder Erweiterung der Fahrzeugfunktionalität durch Flashprogrammierung (Updates) erläutert, bevor auf die Netzwerksoftware eingegangen wird. Die Funktionssoftware wird

an mehreren Beispielen illustriert, unter anderem an der Lenkung. Das in der Automobilindustrie etablierte Sicherheitskonzept zum Schutz von Leib und Lebens der Insassen und des Umfelds wird detailliert. Softwarestandards wie OSEK, ASAM und AUTOSAR werden in Form einer Übersicht vorgestellt.

Für die Softwareentwicklung in Kap. 3 wird als unumgängliche Basis auf das allgemeine Konzept der Eindeutigkeit und Durchgängigkeit von Anforderungen eingegangen. Die konkreten Schritte der praxisnahen Softwareentwicklung werden im Detail vorgestellt. Das beinhaltet auch die für die industrielle Erstellung von Software notwendigen begleitenden Prozesse aus den Bereichen der Qualitätssicherung und des Projektmanagements sowie das Änderungs- und Konfigurationsmanagement. Der Weg von der Spezifikation bis zur Codierung in konkreten Programmiersprachen wird beschrieben. Hierbei wird auf Codierungsrichtlinien für Hochsprachen und die modellbasierte Entwicklung eingegangen. Eine Auswahl in der Praxis eingesetzter Entwicklungswerkzeuge und IT-Infrastruktur wird exemplarisch vorgestellt. Aus der Praxis wird die Umsetzung eines Baukastens erläutert, der die Entnahme fertiger Softwaremodule ermöglicht. Den Abschluss bildet die konkrete Darstellung eines industriell eingesetzten hybriden Verfahrens aus Analyse und Messung für die Ermittlung der Rechenzeit von Software.

Der Softwaretest in Kap. 4 stellt eine besondere Disziplin in der Automobilindustrie dar, da die Nachweisführung für Kundenanforderungen und Gesetzesvorgaben von der abstrakten Ebene des Gesamtfahrzeugbetriebs bis auf das einzelne Bit der Software bilateral notwendig ist. Hierzu werden die etablierten und teilweise vorgeschriebenen Testverfahren und Werkzeuge vom bitweisen Software-Debugger bis zum Fahrzeug im Prüfgelände eingeführt. Es werden Beispiele aus der Praxis, unter anderem aus dem Bereich der Codeanalyse und konkrete Testkataloge für die Abnahmen von Fahrzeugen vor Produktionsstart gegeben.

Prozessmodelle in Kap. 5 stellen als Erweiterung zur Erläuterung des konkreten Vorgehens der Softwareentwicklung und des Softwaretests in den entsprechenden Kapiteln eine Besonderheit dar. Sie bilden eine Klammerfunktion in Form einer Beschreibung für die dargestellten Konzepte. Es werden grundsätzliche Prozessmodelle wie unter anderem das etablierte Wasserfallmodell und das Vorgehensmodell eingeführt, bevor auf Reifegradmodelle aus der Software-Qualitätssicherung und normative Entwicklungsprozesse aus dem Bereich der funktionalen Sicherheit sowie die zugehörigen Assessments eingegangen wird. Der Stand zu den im Rahmen der Digitalisierung zunehmenden agilen Methoden der Softwareentwicklung wird vorgestellt und bewertet. Die praktische Umsetzung einer Prozessmodellierung als konkrete Arbeitshilfe für Entwickler wird gezeigt.

Das Kap. 6 zur Software Variabilität hat eine Sonderstellung. Es handelt sich um einen Gastbeitrag, in dem gezeigt wird, wie sich die Variabilität und die Varianten in der Automobilindustrie mit wissenschaftlichen Methoden fassen lassen. Das Thema wird spätestens mit der schnell kommenden Digitalisierung und den damit verbundenen zusätzlichen Software-Varianten sowie über die verbundenen mobilen Endgeräte der Nutzer Einzug in die Entwicklungsprozesse und letztendlich Produkte der Automobilindustrie halten müssen. Der Fokus dieses Beitrags wird in den Kontext des gesamten Buches gesetzt und stellt damit eine optimale Ergänzung dar.

Inhaltsverzeichnis

1	Fahrzeugelektronik	1
1.1	Gesamtfahrzeugarchitektur	1
1.2	Steuergeräte	3
1.3	Bordnetz	5
1.4	Elektrik-/Elektronikarchitektur	6
1.4.1	Funktionale Anforderungen	7
1.4.2	Technologie	8
1.4.3	Topologie	8
1.5	Entwurfsprozess der Elektrik-/Elektronikarchitektur	9
1.6	Digitale Bussysteme	10
1.6.1	Protokolle	13
1.6.2	Topologien	14
1.6.2.1	Sterntopologie	14
1.6.2.2	Bustopologie	14
1.6.2.3	Ringtopologie	15
1.6.2.4	Baumtopologie	15
1.6.3	Digitale Bussysteme – CAN	15
1.6.3.1	Allgemeine Eigenschaften	15
1.6.3.2	Buszugriff: CSMA/CA	16
1.6.3.3	CAN – Bus: Struktur der Nachricht	17
1.6.3.4	Bitstuffing	17
1.6.3.5	Kollisionsvermeidung	18
1.6.3.6	Fehlererkennung	18
1.6.3.7	High Speed CAN Buspegel	19
1.6.3.8	Low Speed CAN Buspegel	20
1.6.3.9	CAN-Bus: Weitere Anwendungen	20
1.6.3.10	CAN-Bus: Grenzen	20
1.6.4	Digitale Bussysteme – Flexray	21
1.6.5	Digitale Bussysteme – LIN	23
1.6.6	Digitale Bussysteme – MOST	25
1.6.7	Digitale Bussysteme – Vergleich	25

1.6.8	Digitale Bussysteme – Kombination	26
1.6.9	Unterschiede der Zugriffsverfahren	26
1.7	Sensoren	27
1.7.1	Wandlerketten	28
1.7.2	Sensorkennlinien	28
1.7.3	Abstraten	29
1.7.4	Sensorpartitionierung	30
1.7.5	Beispiele für Sensoren	32
1.7.6	Schnittstellenbeschreibung	32
1.8	Aktoren	33
1.8.1	Schnittstellenbeschreibung	33
1.8.2	Beispiele für Aktoren	34
1.9	Mikrocontroller	34
1.10	Schaltungstechnik	35
1.10.1	Programmierung einer Schaltung durch Verdrahtung	35
1.11	Hardwarebeschreibungssprachen	36
1.12	Speicher	38
1.13	Energie	39
1.14	Zusammenfassung	39
1.15	Lernkontrollen	40
1.15.1	Bordnetz	40
1.15.2	Bussysteme	40
1.15.3	Sensoren und Aktoren	40
1.15.4	Steuergeräte	40
2	Software im Fahrzeug	41
2.1	Durchgängigkeit der Anforderungen	42
2.2	Abbildung Funktion auf Architektur	43
2.3	Softwarearchitektur	44
2.4	Echtzeitbetriebssysteme	45
2.4.1	Echtzeitanforderungen	46
2.4.2	Prozess und Prozesszustände	47
2.4.3	Kontextwechsel	47
2.4.4	Scheduling	48
2.5	Diagnose	49
2.5.1	Diagnose in der Automobiltechnik	50
2.5.2	Eigendiagnose: Onboard-Diagnose OBD	51
2.5.2.1	Beispiel Motorsteuergerät	51
2.5.2.2	Fehlererkennung	52
2.5.2.3	Fehlerbehandlung	53
2.5.2.4	Ereignisspeicher	53

2.5.3	Werkstattdiagnose: Offboard-Diagnose	54
2.5.4	Flashen im Kundendienst	54
2.5.5	Flashen im Fahrzeug-Lebenszyklus	55
2.6	Netzwerksoftware	56
2.6.1	Implementierung von Netzwerkprotokollen	56
2.6.2	Kommunikation und funktionale Vernetzung (Konnektivität)	56
2.7	Funktionssoftware	57
2.7.1	Fragmentierung von Steuergeräten	57
2.7.2	Steuergerätedomänen	58
2.7.3	Klimasteuerung	59
2.7.4	Motorsteuerung	60
2.7.5	Lenkungssteuerung	62
2.7.6	Türsteuerung	63
2.7.7	Verteilte Funktionen	63
2.8	Überwachungskonzepte für sicherheitsrelevante Systeme	64
2.8.1	Anforderungen aus den Normen	64
2.8.2	Abgestufte Funktionseinschränkung	65
2.8.3	Diversitäre Programmierung	65
2.8.4	Redundanz in der Elektronik	66
2.8.5	Watchdog und Drei-Ebenen-Konzept	66
2.8.5.1	Funktionsrechner	67
2.8.5.2	Überwachungsrechner	68
2.8.5.3	Ablauf der Überwachung mittels Fehlerzähler	68
2.9	Herstellerübergreifende Softwarestandards	68
2.9.1	Historie	69
2.9.2	Beispiel OSEK/VDX: Betriebssysteme	70
2.9.3	Beispiel ASAM-MDX: Verteilte Softwareentwicklung	71
2.9.4	Beispiel AUTOSAR: Systemarchitektur	72
2.10	Zusammenfassung	76
2.11	Lernkontrollen	76
2.11.1	Architektur	76
2.11.2	Software	77
2.11.3	Echtzeitbetriebssystem	77
2.11.4	Sicherheitskonzept	77
2.11.5	Standards	77
3	Softwareentwicklung in der Automobilindustrie	79
3.1	Stand der Technik	81
3.2	Anforderungen und Architekturentwurf	81
3.2.1	Anforderungserhebung	82
3.2.2	Systemanforderungsanalyse	83
3.2.3	Systemarchitekturentwurf	85

3.2.4	Komponentenanforderungsanalyse	86
3.2.5	Komponentenarchitekturentwurf	87
3.3	Mechanik und Hardware/Elektronik	88
3.4	Softwareentwicklung	90
3.4.1	Softwareanforderungsanalyse	90
3.4.2	Softwareentwurf	91
3.4.3	Funktionssoftwareerstellung	92
3.4.4	Sicherheitssoftwareerstellung	93
3.4.5	Softwareintegrationstest	95
3.4.6	Softwaretest	95
3.5	Integrationstests für Komponenten und System	96
3.5.1	Komponententest	96
3.5.2	Komponentenintegrationstest	97
3.5.3	Systemintegrationstest	98
3.5.4	Systemtest	98
3.6	Übergreifende Prozesse der Softwareentwicklung	99
3.6.1	Qualitätssicherung	100
3.6.1.1	QM-Planung	101
3.6.1.2	QS-Planung	101
3.6.1.3	QS-Durchführung	102
3.6.2	Funktionale Sicherheit	103
3.6.2.1	Konzeption	103
3.6.2.2	Produktentwicklung	104
3.6.2.3	Funktionale Absicherung	105
3.6.3	Projektmanagement	106
3.6.3.1	Projektstart	106
3.6.3.2	Projektplanung	107
3.6.3.3	Projektdurchführung	107
3.6.3.4	Projektabschluss	108
3.6.4	Risikomanagement	109
3.6.4.1	Identifizieren und Bewerten der einzelnen Risiken	110
3.6.4.2	Analyse von Risiken	110
3.6.4.3	Risikobewertung	111
3.6.4.4	Priorisierung	111
3.6.4.5	Festlegung von Maßnahmen	112
3.6.4.6	Risikoverfolgungsliste	112
3.6.5	Lieferantenmanagement	112
3.6.5.1	Lieferantenauswahl	113
3.6.5.2	Lieferantensteuerung	114
3.6.5.3	Lieferantenausphasung	115
3.6.6	Änderungsmanagement	115
3.6.6.1	Vorbereitung Änderungsmanagement	116

3.6.6.2	Änderung	116
3.6.7	Konfigurationsmanagement	117
3.6.7.1	Konfigurationsmanagement Aufbau	117
3.6.7.2	Konfigurationsmanagement Betrieb	118
3.6.8	Problemlösungsmanagement	119
3.6.8.1	Vorbereitung Problemlösungsmanagement	119
3.6.8.2	Problemlösung	120
3.6.8.3	Trendberichterstellung	121
3.6.9	Freigabemanagement	122
3.6.9.1	Freigabevorbereitung	122
3.6.9.2	Freigabedurchführung	122
3.7	Handcodierung in der Programmiersprache C	123
3.8	Modellbasierte Entwicklung	125
3.8.1	Modelle der Elektronik	125
3.8.2	Modelle der Regelstrecke	126
3.8.3	Modelle der Software	127
3.8.4	Modellbasierte Codegenerierung	127
3.8.5	Generierung von Schnittstellencode	128
3.9	Entwicklungswerkzeuge	131
3.9.1	Sicherheitsanalysen	131
3.9.2	Handcodierung von Steuergerätesoftware	131
3.9.3	Prüfung von Handcode/Codierungsrichtlinien	131
3.9.4	Modellbasierte Entwicklung	131
3.9.5	Test	132
3.9.6	Kommunikation	132
3.9.7	Sonstiges/IT-Infrastruktur	132
3.10	Modulbaukasten für Plattformsoftware	132
3.11	Rechenzeitanalyse von Softwarefunktionen	142
3.11.1	Stand der Technik	143
3.11.2	Rechenzeitbestimmende Faktoren	143
3.11.3	Anforderungen an das Messverfahren	144
3.11.4	Verfahren der hybriden Rechenzeitanalyse	145
3.11.5	Softwareumgebung	146
3.11.6	Analytische Betrachtung der Softwarefunktion	146
3.11.6.1	Kontrollflussgraph	147
3.11.6.2	Codeanalyse	147
3.11.6.3	Softwareabschnitte für die Laufzeitanalyse	147
3.11.7	Instrumentierung	148
3.11.7.1	Instrumentierung des Quellcodes	149
3.11.7.2	Instrumentierung der Funktionen	149
3.11.7.3	Instrumentierung der bedingten Verzweigungen	150
3.11.7.4	Erzeugen von Stimuli-Daten	151

3.11.7.5	Messaufbau	151
3.11.7.6	Durchführung der Messung	152
3.11.7.7	Genauigkeit der Messmethodik	152
3.11.8	Statische Analyse für Worst Case und Best Case	154
3.11.9	Zusammenfassung des Messverfahrens	155
3.12	Zusammenfassung	156
3.13	Lernkontrollen	156
3.13.1	Softwareentwicklung	156
3.13.2	Programmierung	157
3.13.3	Modulbaukasten	157
3.13.4	Rechenzeitanalyse	157
4	Softwaretest	159
4.1	Softwarefehler	159
4.1.1	Ursachen von Softwarefehlern	159
4.1.2	Schäden durch Softwarefehler	160
4.1.3	Prominente Beispiele	161
4.2	Grundlagen des Softwaretests	162
4.2.1	Testbegriff	162
4.2.2	Ablauf von Tests	162
4.2.3	Fehlerbegriff	163
4.2.4	Ziel des Tests	163
4.2.5	Maße für Software	164
4.2.5.1	Zyklomatische Komplexität	165
4.2.5.2	Halstead-Maße	165
4.2.5.3	Live-Variables	167
4.2.6	Vorgehen und Strategie im Softwaretest	167
4.2.7	Testaufwand	168
4.2.8	Weitere Begriffe und Definitionen	169
4.2.8.1	Verifikation und Validierung	169
4.2.8.2	Analysemethoden	169
4.2.8.3	Statische und dynamische Analyse	170
4.2.8.4	Testumgebung und Testfall	170
4.2.8.5	Testarten in der Automobilindustrie	171
4.3	Merkmalsräume von Softwaretests	172
4.3.1	Prüfebene	172
4.3.1.1	Unit-Tests und Modultests	175
4.3.1.2	Softwareintegrationstests	176
4.3.1.3	Systemintegrationstest	178
4.3.1.4	Abnahmetest	179
4.3.2	Prüfkriterium	180
4.3.2.1	Funktionstest	180

4.3.2.2	Trivialtest	181
4.3.2.3	Crashtest	181
4.3.2.4	Kompatibilitätstest	181
4.3.2.5	Zufallstest	182
4.3.2.6	Installationstest	182
4.3.2.7	Ergonomietest	182
4.3.2.8	Sicherheitstests	183
4.3.2.9	Komplexitätstest	183
4.3.2.10	Laufzeittests	184
4.3.2.11	Lasttest	184
4.3.2.12	Stresstest	185
4.3.3	Prüfmethodik und Prüftechniken: Black Box Tests	185
4.3.3.1	Äquivalenzklassentest	186
4.3.3.2	Zustandsbasierter Softwaretest	188
4.3.3.3	Use Case Test	190
4.3.3.4	Entscheidungstabellenbasierter Test	190
4.3.3.5	Back-to-Back-Test	191
4.3.3.6	Regressionstest	192
4.3.3.7	Mutationstest	193
4.3.4	Prüfmethodik und Prüftechniken: White Box Tests	193
4.3.4.1	Anweisungüberdeckung (C0)	195
4.3.4.2	Kantenüberdeckung (C1)	196
4.3.4.3	Bedingungsüberdeckung (C2, C3)	197
4.3.4.4	Pfadüberdeckung (C4)	198
4.3.5	Testmetriken und Grenzen des Softwaretests	199
4.4	Hardware-in-the-Loop in der Automobilelektronik	201
4.4.1	Historie	202
4.4.2	Automatisierung der Tests	202
4.5	Zusammenfassung	204
4.6	Lernkontrollen	204
4.6.1	Fehlerbegriff	204
4.6.2	Software-Maße	205
4.6.3	Testverfahren	205
5	Prozessmodellierung	207
5.1	Der Softwareentwicklungsprozess	209
5.2	Das Wasserfallmodell	211
5.3	Das V-Modell	212
5.4	Anwendungsbezogene Prozessmodelle	214
5.4.1	Das Prototypenmodell	214
5.4.2	Das evolutionäre Modell	215
5.4.3	Das inkrementelle Modell	216

5.4.4	Das nebenläufige Modell	216
5.4.5	Das Spiralmodell	217
5.5	Zusammenfassung der klassischen Prozessmodelle	218
5.6	Alternativen zu klassischen Prozessmodellen	219
5.6.1	Rational Unified Process	219
5.6.2	Extreme Programming	221
5.6.3	ROPES	222
5.7	Einsatz der vorgestellten Prozessmodelle	222
5.8	Verbesserung von Entwicklungsprozessen: Reifegrade	223
5.9	CMMI	223
5.9.1	Reifegrade in CMMI	224
5.9.2	Prozessgebiete	225
5.9.3	Spezifische und generische Ziele	226
5.9.4	Zertifizierung der Organisation	227
5.10	SPICE	228
5.10.1	Referenz- und Assessmentmodell	228
5.10.2	Prozessdimension	230
5.10.3	Beispiel zur Prozessdimension:	231
5.10.4	Reifegraddimension und Reifegradstufen	232
5.10.5	Beispiel zu Reifegraddimensionen:	234
5.11	Vorteile und Nachteile von SPICE	234
5.12	Automotive SPICE	235
5.12.1	Assessments – Prinzipien	238
5.12.2	Assessments – Ablauf	242
5.13	Funktionale Sicherheit	244
5.13.1	IEC 61508	244
5.13.2	Safety Integrity Level	246
5.13.3	Failure in Time	246
5.13.4	Gefahren- und Risikoanalyse: Gefahrenanalyse	247
5.13.5	Gefahren- und Risikoanalyse: Risikoanalyse	252
5.13.6	IEC 61508 Assessment	253
5.13.7	ISO 26262	254
5.14	Agile Methoden	255
5.15	Reifegrade und Prozessmodellierung in der Praxis	257
5.16	Zusammenfassung	258
5.17	Lernkontrollen	258
5.17.1	Prozessmodelle	258
5.17.2	Reifegradmodelle	259
5.17.3	Funktionale Sicherheit	259

6	Software-Variabilität in der Automobilindustrie	261
6.1	Das Body Comfort System mit variabler Fenstersteuerung als laufendes Beispiel	263
6.2	Grundlagen der Software-Produktlinien	266
6.2.1	Erstellung einer Software-Produktlinie	267
6.2.2	Verwendung einer Software-Produktlinie	270
6.3	Variabilitätsmodelle	273
6.3.1	Feature Models mit Cross-Tree Constraints	273
6.3.2	Decision Models mit Resolution-Constraints	276
6.3.3	Orthogonal Variability Models (OVMs) mit Constraints	279
6.4	Variabilitätsrealisierungsmechanismen	281
6.4.1	Annotativer Variabilitätsrealisierungsmechanismus	281
6.4.2	Kompositionaler Variabilitätsrealisierungsmechanismus	284
6.4.3	Transformativer Variabilitätsrealisierungsmechanismus	287
6.5	Wahl und Verwendung geeigneter Technologien für den Praxiseinsatz von Software-Produktlinien	290
6.5.1	Wahl und Verwendung des Variabilitätsmodells	290
6.5.2	Wahl und Verwendung des Variabilitätsrealisierungsmechanismus	291
6.6	Weiterführende SPL Konzepte und Techniken	295
6.6.1	Prozesse zur Aufnahme der Entwicklung von Software- Produktlinien	295
6.6.1.1	Proaktive Entwicklung von Software-Produktlinien	295
6.6.1.2	Reaktive Entwicklung von Software-Produktlinien	296
6.6.1.3	Extraktive Entwicklung von Software-Produktlinien	297
6.6.2	Spezialisierte Notationen für Feature Models	298
6.6.2.1	Kardinalitätsbasierte Feature Models	298
6.6.2.2	Attributierte Feature Models	299
6.6.2.3	Hyper-Feature Models	300
6.6.2.4	Temporal Feature Models	300
6.6.3	Multi-Software-Produktlinien (MSPLs)	301
6.6.4	Staged Configuration	304
6.7	Zusammenfassung	306
7	Zusammenfassung und Abschluss	307
	Literatur	309
	Sachverzeichnis	311

Autor



Professor Dr.-Ing. Fabian Wolf Jahrgang 1971, hat von 1996 bis 2001 im Bereich der Software-Entwicklungswerkzeuge im Mobilfunkbereich gearbeitet und über die Rechenzeitanalyse eingebetteter Software an der Technischen Universität Braunschweig promoviert.

Von 2001 bis 2014 hat er die verteilte Entwicklung, den Test und die Prozesse für Software in Motorsteuerungen sowie der Lenkungselektronik in verschiedenen inhaltlichen, organisatorischen und leitenden Positionen bei Volkswagen vorangetrieben. Seit 2014 ist er für die Prozesse der Elektronikentwicklung von Fahrzeugkomponenten im Konzern verantwortlich.

Seit 2009 lehrt er parallel das Fach Fahrzeuginformatik an der Technischen Universität Clausthal, die ihn 2016 zum Honorarprofessor bestellt hat.