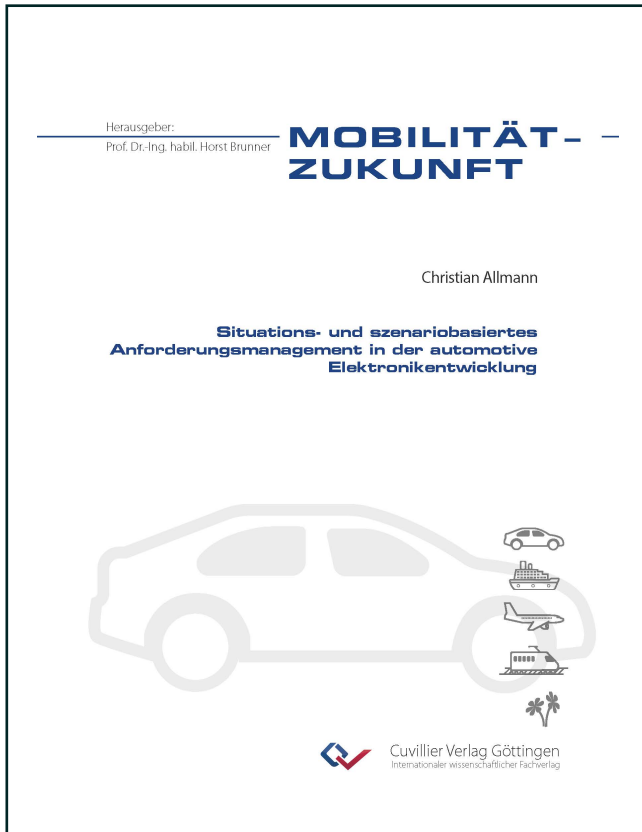




Christian Allmann (Autor)
**Situations- und szenariobasiertes
Anforderungsmanagement in der automotive
Elektronikentwicklung**



<https://cuvillier.de/de/shop/publications/8862>

Copyright:

Cuvillier Verlag, Inhaberin Annette Jentzsch-Cuvillier, Nonnenstieg 8, 37075 Göttingen,
Germany

Telefon: +49 (0)551 54724-0, E-Mail: info@cuvillier.de, Website: <https://cuvillier.de>

1. Einführung

Die Entwicklung in der Automobilindustrie ist seit mehreren Jahrzehnten gekennzeichnet durch den rasanten Anstieg an elektronischen Systemen. Diese, zum großen Teil eingebetteten Systeme tragen zu einem wachsenden Wertschöpfungsanteil des Automobils bei [BBC⁺03b, DK04, SS07a, BIT08]. Neben der Automobilindustrie spielen Eingebettete Systeme eine zentrale Rolle in vielen technologisch getriebenen Domänen (u.a. Aerospace, Verkehr, Maschinen- und Anlagenbau, Telekommunikationstechnik) [Wym07, BGS⁺07].

Als Eingebettetes System wird jener Verbund bezeichnet, dessen Struktur sich aus Mikroprozessoren, Sensoren und Aktuatoren zusammensetzt. In Kombination mit der dazugehörigen eingebetteten Software übernehmen diese Systeme Regelungs- und Steuerungsaufgaben. Die Besonderheiten von Eingebetteten Systemen gegenüber reinen Informationssystemen (siehe auch [ZA95, KK97, Dou99]) drücken sich insbesondere im Anforderungsmanagement durch den großen Umfang an zu berücksichtigenden Nichtfunktionalen Anforderungen und System Constraints aus. Die Berücksichtigung dieser Anforderungen (u.a. Ressourcenbeschränkung, Echtzeit, Zuverlässigkeit, Heterogenität) erfordert besondere Kompetenz der Entwickler und deren Unterstützung, u.a. durch angepasste Methoden und Prozesse für die Systementwicklung. [BBC⁺03b, FGP04, All07b, Hel05a, Hel05b]

Im Folgenden werden anhand von Beispielen aus dem Automotivesektor die Herausforderungen des Embedded Bereiches für aktuelle und zukünftige Systeme dargestellt. Die sich abzeichnenden Ursachen werden in den beiden darauffolgenden Kapiteln hinsichtlich ihrer Auswirkungen auf die Systementwicklung unter besonderer Berücksichtigung des Software und Requirements Engineering (RE) analysiert. Aus den daraus ermittelten Aufgaben an neuen Methoden insbesondere für den in dieser Ausarbeitung fokussierten Bereich des Anforderungsmanagement werden zum Abschluss die Ziele für die vorliegende Arbeit abgeleitet.

1.1. Motivation

Wird heute versucht, den Handlungsbedarf für die Entwicklung Eingebetteter Systeme zu beschreiben, so wird meist auf den steigenden Bedarf an Speicherplatz, die hohe Anzahl an Steuergeräten oder die wachsenden Funktionsumfänge verwiesen [Sch01b, Ema05, Wym07, BKPS07]. Aus Entwicklersicht sind dies aber nur Symptome, die die eigentliche Herausforderung, nämlich die Beherrschung der wachsenden Systemkomplexität, nur schemenhaft beleuchten. Im Folgenden werden drei Ursachen dieser Systemkomplexität an Beispielen erläutert:

1. Vernetzung und Verteilung
2. Variantenbildung
3. Erweiterung von Systemgrenzen

Die erste Komplexitätsursache *Vernetzung und Verteilung* wird am Beispiel der Funktion „Rückwärts fahren“ erläutert. In heutigen Fahrzeugen sind verschiedene vom Kunden wahrnehmbare Fahrzeugfunktionen wie das „Rückwärts fahren“ über verschiedene Steuergeräte verteilt (hier u.a. Getriebe, Lichtmodul, etc.). Dabei kommunizieren diese Funktionen untereinander, um Daten auszutauschen, die u.a. neue Funktionen anstoßen (z.B. Einparkhilfe). Erst durch die Vernetzung von Funktionen ist es möglich, dass, wie in diesem Beispiel, erweiterte Funktionalitäten, wie das automatische abklappen des Außenspiegels beim Einlegen des Rückwärtsganges, realisiert werden können, ohne dass alle Funktionsumfänge auf einem Steuergerät integriert werden müssen.

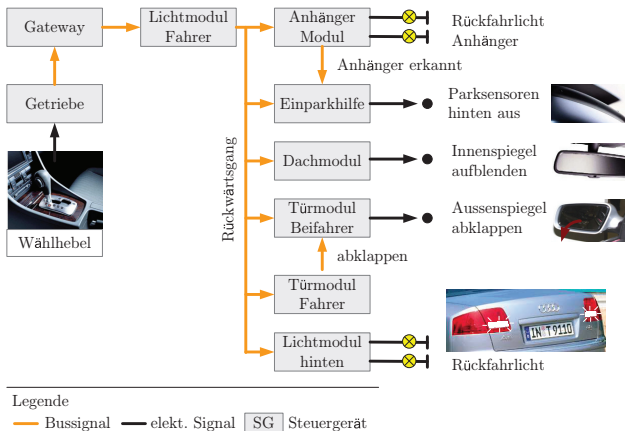


Abbildung 1.1.: Verteilte Funktionalität

Könnte auf der anderen Seite jede Funktion für sich autark, d.h. unabhängig gegenüber anderen Funktionen¹ spezifiziert und integriert werden, ohne dass sie mit anderen Funktionen Informationen austauschen müsste, so könnte sie gleichfalls unabhängig in ein Fahrzeug auf ein Steuergerät integriert werden. Wenn dem so wäre, würde jede in der Literatur aufgeführte steigende (oftmals exponentiell wachsende) Kurve für Speicherplatz, die Anzahl an verbauten Steuergeräten, etc. den Entwicklungsaufwand relativieren. Der Zuwachs an neuen Funktionen wäre proportional zum Entwicklungsaufwand, der

¹Designprinzip nach CONSTANTINE: *A structure is stable if cohesion is strong and coupling is low* [SMC74, Mye75, ER03]

mittels bewährten, etablierten Entwicklungsmethoden gelöst werden könnte. Das dieser Entwicklungsaufwand nicht unbedingt proportional steigt, wird am Beispiel der Funktion „Rückwärts fahren“ (siehe Abbildung 1.1) für die Ursache Vernetzung und Verteilung im Folgenden erläutert.

Ausgehend vom Einlegen des Rückwärtsganges werden über diverse Kommunikationsmedien (hier Bussysteme und Signalleitungen) bis zu neun Steuergeräte angesprochen, um die Funktion „Rückwärts fahren“ zu realisieren. Vergleicht man den Funktionsumfang mit dem eines Fahrzeuges aus den 80ern, so aktiviert das Einlegen des Rückwärtsganges mehr als die Getriebewahl oder die Aktivierung der Rückleuchte. Die Beispielfunktion umfasst dabei mehrere Teilfunktionen, die abhängig von der Ausstattung des Fahrzeuges angesprochen werden. Hierunter fällt die Ansteuerung der Einparkhilfe (Park Distance Control (PDC)²), die in diesem Beispiel auf Grund des Anhängers deaktiviert ist. Eine weitere angesteuerte Teilfunktionalität ist das Abklappen des Außenspiegels, zur besseren Wahrnehmung des Fahrzeughecks beim Einparken.

Das Beispiel zeigt zum ersten, dass die entstehende Komplexität nicht nur für neue innovative Funktionen, sondern insbesondere für etablierte Funktionen in ihrer Verteilung und Vernetzung zu beherrschen gilt. Zweitens, neue Funktionalität entsteht u.a. durch verbesserte Wahrnehmung der Umgebung (hier: Anhänger) und durch Änderung des Fahrzeugumfeldes (Aktuatorik in Außen- und Innenspiegel). Drittens, die Komplexität einer Funktion steigt um den Grad, an der ihre Teilfunktionen verteilt auf Steuergeräten miteinander und mit anderen Funktionen kommunizieren³. Die hier betrachteten Funktionen auf Systemebene stellen an den Kunden ausgerichtete Anwendungen dar. Somit kann die Anwendung als Treiber der Systemkomplexität angesehen werden.

Das erste Beispiel zeigt die systemimmanente Funktionskomplexität, das zweite Beispiel in Abbildung 1.2 ergänzt diese um die Dimension der Varianten (hier: Länder-) am Beispiel der Verbraucherten eines Navigationssystems beziehungsweise (bzw.) Telefons. Navigationssysteme bzw. das Mensch-Maschine-Interface (MMI)⁴ ist aus Software-/Speicherplatzsicht eines der größten Systeme, das im Fahrzeug verbaut ist. Hier besteht die Herausforderung, all jene Varianten in der Entwicklung zu berücksichtigen, die eine Platzierung des Produktes in den verschiedenen Märkten ermöglichen, ohne ein späteres vollständiges Reengineering des Systems durchzuführen. Hier gilt es, nicht nur die Software (SW), sondern auch die Hardware (HW) zu berücksichtigen. So ist z.B. zu bedenken, dass in Amerika in großen Teilen Mittelwelle zum Empfang des Radiosenders genutzt wird; hierfür muss der Empfänger/Antenne entsprechend ausgelegt werden. Die Gründe für eine Anpassung von Systemen sind vielfältig (u.a. neue Märkte, neue Systemumgebung), sie führen entweder zu einem Redesign des Systems oder zu einer Variantenbildung des

²Assistenzsystem, das den Fahrer akustisch oder visuell vor erkannten Hindernissen beim Einparken warnt

³Eine Metrik zur Bestimmung des hier erwähnten Komplexitätsgrades ist auf Basis von Allokations- und Partitionierungsalgorithmen für Funktionen ableitbar (siehe [Nis97])

⁴siehe auch Human Machine Interface (HMI)

aktuellen Systems. Für die Entwicklung neuer Systeme und Funktionen gilt es dabei zu berücksichtigen, dass sowohl HW wie SW als denkbare Variationspunkte des Systems anzusehen sind.

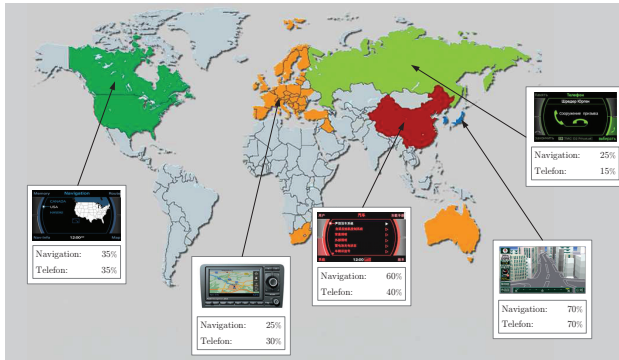


Abbildung 1.2.: Ländervarianten

Auch wenn diese beiden Beispiele nur auszugsweise die aktuellen Ursachen der Herausforderung im Embedded Umfeld beschreiben, so lässt sich zusammenfassend Folgendes festhalten: Komplexität entsteht durch verteilte, vernetzte Funktionen, die über verschiedene Kommunikationsmedien unter definierten Qualitäts- (u.a. Security, Safety) und Laufzeiteigenschaften (z.T. harte Echtzeit) miteinander Informationen austauschen. Jede der aufgeführten Funktionsbeispiele muss heute für unterschiedliche Zielmärkte verfügbar sein, was die Systemkomplexität durch die Beherrschung ihrer jeweiligen Varianten ansteigen lässt. Die Systeme der Zukunft zeigen eine Verschärfung der genannten Komplexitätsursachen, durch eine stärkere Vernetzung des Fahrzeuges nach innen und nach außen (d.h. einer detaillierteren Erfassung der nahen und fernen Umwelt) und eine wettbewerbsdifferenzierende Umsetzung neuer Funktionen (siehe auch [JSJF98, CB05, SFT05, BR05, FPZ05, HTG05]). Das Beispiel Car-to-X (C2X) soll hier als Vertreter diese Entwicklung verdeutlichen.

In Abbildung 1.3 sind auszugsweise Funktionen aufgeführt, die mit Hilfe der Technologie von C2X realisiert werden können. Die Funktionen können in fast allen denkbaren Bereichen von Fahrzeugsicherheit (u.a. Kreuzungsassistent, Fußgängerschutz, Unfallwarnung), Komfort (u.a. Point of Interest (POI)-Informationen, Parkleitsysteme, virtueller Stadtführer) bis hin zu präziseren, aktuelleren Verkehrsinformationen für Routenplanungen auf Straßen inner- und außerorts eingeordnet werden.

01. Traffic signal / Stop sign violation warning	25. Blind spot warning / lane change warning
02. General in-vehicle signage	26. Wrong way driver warning
03. Left-turn assistant	27. Rail collision warning
04. Intersection collision warning	28. Highway merge assist
05. Pedestrian crossing information	29. Visibility enhancer
06. Emergency vehicle approaching warning /signal pre-emption	30. Cooperative adaptive cruise control
07. Vehicle safety inspection	31. Cooperative platooning
08. Electronic licence plate	32. Cooperative glare reduction
09. Electronic driver's licence	33. Adaptive drivetrain management
10. In-vehicle Amber alert	34. Intelligent traffic flow control
11. Stolen vehicle tracking	35. Road surface conditions to TOC
12. Post-crash / breakdown warning	36. Crash data to TOC
13. SOS services	37. Parking spot locator
14. Pre-crash sensing	38. Enhanced route guidance and navigation
15. Event data recording	39. Map download / update
16. Work zone warning	40. GPS correction
17. Curve-speed warning	41. Cooperative positioning improvement
18. Vehicle-based / Infrastructure-based road condition warning	42. Instant messaging
19. Safety recall notice	43. POI notification
20. Just-in-time repair notification	44. Internet service provisioning
21. Wireless diagnostics	45. Mobile access to vehicle data
22. Software update / flashing	46. Fleet management
23. Cooperative forward collision warning	47. Area access control
24. Emergency electronic brake lights	48. Electronic payment
	49. Rental car processing
	50. Hazardous material tracking

Abbildung 1.3.: Auswahl an C2X Funktionen

Die Systemgrenzen, die bisher durch das Fahrzeug selbst definiert wurden, werden durch die Vernetzung nach außen zu anderen Fahrzeugen, Infrastrukturanlagen (z.B. Ampeln) oder anderen Informationsquellen erweitert und neu definiert. Der Vernetzungsgrad würde sich nicht weiterhin nur auf das Fahrzeuges und seine Funktionen beziehen, sondern würde um jene Funktionen (u.a. Internet-Services) ergänzt werden, die außerhalb des Fahrzeuges realisiert sind. Neue Funktionsverteilungen wären denkbar, bei denen Teile einer Funktion bzw. Daten außerhalb des Fahrzeuges (z.B. Straßenkarten) sich befinden.

Selbst wenn man die Herausforderungen der technischen Umsetzbarkeit für einen Augenblick nicht berücksichtigt, dann bleiben weiterhin Fragen der Kundenakzeptanz, Marktumsetzung, Realisierung etc. unbeantwortet. Da diese, wie in den Beispielen skizziert, zu großen Teilen von der Vernetzung, Verteilung und Variantenbildung der Funktionen und Systeme abhängen. Ergänzend hierzu, müssten Fragen der Standardisierung, Quality of Service (QoS), der Verfügbarkeit, (Daten-) Sicherheit, Safety, etc. sowohl aus wirtschaftlicher wie technischer Sicht geklärt werden.

Letztendlich entscheiden eine Reihe von Faktoren (u.a. Umsetzbarkeit, Wirtschaftlichkeit, Kundenakzeptanz) über den Serienentscheid, also dem Übergang einer neuen Anwendung in ein Zielfahrzeug. Die Herausforderung besteht nun darin, diese Faktoren und ihre Abhängigkeiten zu kennen und zu analysieren. Das vorliegende Beispiel zeigt, dass gerade für die Funktionen bei C2X neben den technischen Faktoren eine Größe für

die Beurteilung ganz entscheidend ist, das Einführungsszenario. Alle Funktionen von C2X basieren auf Informationen, die sich das System extern (außerhalb des Fahrzeuges) beschaffen muss. Werden diese nicht durch andere Quellen (z.B. Fahrzeuge - Car-to-Car (C2C)) zur Verfügung gestellt, da die zu Grunde liegende Technologie nicht zur Ausstattung gehört, fehlt der Anwendung die Basis. Technologien, die eine große Fülle an neuen Anwendungen ermöglichen, haben den Nachteil, dass deren Machbarkeit von vielen unterschiedlichen Faktoren abhängt. Am Beispiel C2X zeigt eine Prognose der Marktdurchdringung nichttechnische Grenzen der Umsetzbarkeit auf.

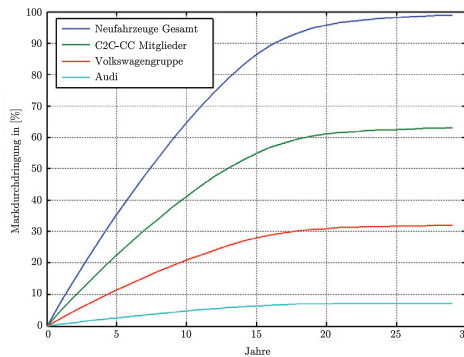


Abbildung 1.4.: Prognose der Marktdurchdringung für C2C in Deutschland

Abbildung 1.4 zeigt exemplarisch die initiale Dauer in Jahren bis zur Erreichung einer von der Anwendung geforderten Ausstattungsrate bei Ausstattung eines jeden Neufahrzeuges (vier Kurven) mit der notwendigen Technologie. Durch diese Sicht auf eine Technologie und unter Berücksichtigung des jeweiligen Geschäftsmodells (z.B. First in Market) reduzieren sich die möglichen Funktionen in Abbildung 1.3 auf eine überschaubare Anzahl, deren (Vor-)Entwicklung nun gründlicher begegnet werden kann.

Technologien wie C2X oder Ubiquitäre Systeme, Pervasive Computing, Ambient Intelligence, Augmented Reality, Context Aware Computing etc. schaffen neue Anwendungsfelder für Eingebettete Systeme im Automobil [BGS⁺07, End07, Bul04, Krä07]. Sie zeigen aber auch, dass mit ihnen die Systemgrenzen des Fahrzeuges erweitert werden. Die Komplexität, mit der sich der Entwickler beschäftigen muss, steigt. Er ist auf Unterstützung durch Prozesse, Methoden und Tools angewiesen, die es ihm erlauben, diese Komplexität zu beherrschen.

1.2. Herausforderungen im Automotive Systems Engineering

Die beschriebenen Anwendungen zeigen Symptome der wachsenden Systemkomplexität und den damit verbundenen Herausforderungen in der Entwicklung Eingebetteter Systeme⁵. Es ist daher für das Verständnis und die Transparenz des später in dieser Arbeit entwickelten Lösungsansatzes notwendig, die aktuellen Themenfelder sowohl in Hinblick der Herausforderungen als auch der Lösungsansätze zu skizzieren. Hierzu muss die eingangs formulierte These „*Die Anwendung ist Treiber der (Funktions-)Komplexität*“ weiter präzisiert werden. Neue, innovative Anwendungen im Embedded Bereich entstehen in erster Linie durch die Nutzung effektiverer/effizienterer Technologien u.a. für Aktuatoren, Sensoren und Kommunikation⁶. Diese Technologien für den automotive Sektor zu entwickeln, verlangt neben der Abwägung vieler Faktoren auch die Berücksichtigung der erwähnten nichtfunktionalen Anforderungen u.a. Sicherheit, Verfügbarkeit, Robustheit, Leistung, Integrität, Benutzerfreundlichkeit. Erst in zweiter Linie entstehen neue Anwendungen durch die intelligente (Wieder-) Verwendung existierender Infrastrukturen (Hard- und Software) mittels verbindenden, neuen Softwarefunktionen⁷. Vor diesem Hintergrund steht die Entwicklung von Technologie geprägter und getriebener Anwendungen vor der Herausforderung, eine wachsende Funktionskomplexität methodisch zu beherrschen, um eine hohe Software- und Systemqualität zu garantieren [BDP06]. Die Liste an zu berücksichtigenden Einflussgrößen (intrinsisch⁸ wie extrinsisch⁹) auf die Entwicklung ist auf Grund der heute meist offenen Systeme schwierig zu überblicken.

Ein wesentlicher Schritt zur Lösung muss es daher sein, eine einheitliche Systematik zur umfassenden Beherrschung multifunktionaler Anwendungen (siehe C2X) zu etablieren, beginnend mit der Anforderungserhebung [Gri03, WW02, FGP04, HS06, BKPS07, BGS⁺07, Fle08]. Diese Systematik muss dabei die folgenden drei Handlungsfelder berücksichtigen:

Paradigmen im Entwicklungsprozess:

- geographisch verteilte Entwicklung entlang der Zuliefererkette [Hef00, AWK06a, AOK08]
- Integration von Reifegradmodellen zwischen OEM und Zulieferer(-kette) [SZ06, HW05a]
- Verständnis des Zulieferers als Funktions- oder Plattformlieferanten [BBC⁺03b, SZ06, HH04, Spi08]
- Wahrung des Interlectual Property (IP)-Schutzes von Original Equipment Manufacturer (OEM) und Zulieferer [DK04, AOK08, BKPS07]

⁵siehe WYMAN: Innovationstrend 2015 im Automobil [Wym07], KRÄMER [Krä07]

⁶siehe Fallbeispiele Kapitel 2

⁷Beispiel: PreCrash-Systeme, die den Fahrer in die ideale Sitzposition fahren, Fenster schließen und den Gurt vorstraffen [Krä07]

⁸durch die Systemauslegung (z.B. Motorraum, Schutz der Elektronik u.a. vor Temperatur)

⁹durch die Fahrzeugumwelt (z.B. Salzwasser, Schutz der Elektronik u.a. vor Korrosion)

- Rolle des OEMs sowohl als Systemintegrator und Konzeptentwicklers [HH04, AWK06a, BKPS07, SS07a]

Domänenübergreifende Konzepte:

- Entwicklung mechatronischer Systeme [Buc00, Sta01, Fra06]
- Berücksichtigung unterschiedlicher Lebenszyklen (Innovations-) von Hardware und Software [BBC⁺03b, BKPS07]
- Variantenkomplexität durch Markt, Ausstattung und Produkt-Lebens-Zyklus [BKPS07]
- Sicherstellung der Robustheit und Sicherheit (Safety) [x-b98, Mah00, Ehr03, WNSSL04, AM07, AOK08]
- Verbindung unterschiedlicher Ingenieurskulturen (Kontrollfluss z.B. Simulink vs. Informationsfluss, z.B. UML) [Hoh03, CCM⁺03, BKPS07]

Funktionsentwicklung:

- Modularisierung von Software (System- und Applikationssoftware) [BBC⁺03b, Sal02]
- Planmäßige Wiederverwendung von Softwareartefakten (Systemkomponenten) [Tra05, Kub07, Fri08]
- Kompatibilitätsmanagement [Sta04, BBDS06]
- Vermeidung nicht beabsichtigter Funktionsinteraktionen (Feature-Interaction) [BDC⁺89, Zav93]
- Validierung und Verifikation von Software, Hardware in ihren Integrationsphasen [Kut04, Web06, Met04]
- Durchgängige Toolkette und automatische Codegenerierung [ESB01, BOJ04, FGG⁺05, SCFD06, Int08]

Die aufgeführten drei Handlungsfelder zeigen die Ideen und Konzepte zur Bewältigung der eingangs erwähnten Beherrschung der Systemkomplexität und ihren Ursachen. Für viele diese Ideen existieren bereits methodische Ansätze, die in Entwicklungswerkzeugen Einzug gehalten haben. In der Fachwelt [Bro06, BPSS06, KNP04] werden insbesondere die folgenden Ansätze des Systems und Software Engineering diskutiert, die eine Anpassung etablierter Methoden für eine durchgängige Systematik im Embedded Umfeld versprechen:

- modellbasierte Entwicklung [Kri00, Rau01, KNP04, Met04, AKMP05, ZBF⁺05]
- modellbasierte Codegenerierung [BBC⁺03b, SCFD06]
- standardisierte Softwarekonzepte (u.a. AUTOSAR, OSEK)[Kon06, 20004c, 20004b, 20004a, 20005]

- Software Produktlinien (SPL)[PR02, KFAK05, FKKSP05, KKSP05, All05, Kub05, Kol06, Kub07]
- domänenspezifische Sprachen (u.a.EAST-ADL)[MT00, LSNT04, SS04, FLS⁺04]

Die Aufführung der Methoden und Techniken zeigt, dass ihr Hauptanwendungsgebiet die Entwicklung von eingebetteter Software ist. Wie bereits in den vorangegangenen Kapiteln aufgeführt, gibt es mehrere Zusammenhänge zwischen Anwendung, Funktion, System und Technologie. Jegliche Form der Anpassung, der Änderung der Frage, welche Technologie möglich wäre, wirft Fragen der Entwicklungsintegration und -grenzen auf ¹⁰. Methodische und technologische Lösungen müssen in diesem Zusammenhang beide in eine gemeinsame Systematik eingebettet werden, mit dem Ziel nicht separate Lösungen für einzelne Fragen zu liefern, sondern im Sinne einer durchgängigen Toolkette [Han05] den Entwickler zu jeder Zeit methodisch zu unterstützen.

1.3. Abgeleitete Herausforderungen für das Anforderungsmanagement

Wie der vorangegangene Abschnitt zeigt, sind für eine Reihe an Herausforderungen in der Embedded Entwicklung abgestimmte Konzepte und Technologien entwickelt worden, die eine nachweisliche Reduzierung der Entwicklungskomplexität aufweisen können (siehe aufgeführte Quellen). Die Herausforderung besteht nun darin, die einzelnen Ansätze miteinander zu verbinden. Eine solche abgestimmte Systematik hat das hehre Ziel, den Entwickler bei der Lösung seiner Aufgaben methodisch zu jedem Zeitpunkt in der Entwicklung zu unterstützen. Viele der genannten Ansätze setzen Grundlagen voraus, seien es Artefakte (z.B. Feature, Modelle, Formalisierung) oder Fachwissen, die nur bedingt von Fachexperten geliefert werden können, und dies nur ab bestimmten Zeitpunkten in der Entwicklung. Gerade bei Neuentwicklungen und technischen Innovationen ist diese Grundlage nicht gegeben. Es fehlt ein Konzept, das einen methodischen, roten Faden von den ersten Analysen (Forschung) über die ersten Prototypen (Vorentwicklung) bis zur letztendlichen Serienumsetzung (Serienentwicklung) im Fahrzeug und späteren Fahrzeugvarianten, -versionen legt. Im Gegensatz zu BROY[BKPS07] besteht die Herausforderung nicht unmittelbar darin, eine höhere Qualität und Zuverlässigkeit zu erzielen, sondern vielmehr diese frühzeitiger und präziser zu bestimmen. Die zu entwickelnde Systematik hat somit die dringliche Aufgabe, die hierfür notwendigen Anforderungen auf ihren jeweiligen Abstraktionsebenen (u.a. Funktions- und Bauteilebene)¹¹ zu identifizieren und die Verfolgbarkeit entlang der Entwicklungsphasen sicherzustellen.

Nach GRIMM und WEBER [Gri03, WW02] ist dieses Konzept eines roten Fadens bisher daran gescheitert, dass ein zugeschnittener RE-Prozess als zentrales Bindeglied zwischen modellbasierter Entwicklung, Wiederverwendungskonzepten und Technologieplattformen fehlte. Dieser Prozess hätte die Aufgabe, die frühen Phasen der Entwicklung (Forschung,

¹⁰Beispiel: Grenzen und Möglichkeiten von automotive Ethernet [AFH⁺03]

¹¹siehe Ebenenmodelle [RFH⁺05, All07a, BKPS07]

Vor- und Konzeptentwicklung) untereinander und, bei erfolgreicher Evaluierung des Konzeptes, über die Konzeptentwicklung mit der Serienentwicklung zu verbinden¹². Durch die regelmäßigen Berichte der Gartner¹³ und Standish Group¹⁴ wird bereits seit Jahren darauf hingewiesen, dass dem RE eine zentrale Rolle zur Sicherstellung der System- und Software-Qualität zukommt. Bisherige Arbeiten konzentrierten sich auf die Phase der Serienentwicklung, die durch einen hohen Prozentsatz an Übernahmemüßigkeiten, Bauteilen, Funktionen und Spezifikationen gekennzeichnet ist [HMH03, HH04]. Für die geforderte einheitliche Systematik fehlt die Anbindung an die frühen Entwicklungsphasen. In diesem Abschnitt fehlt die Erfahrung mit der eingesetzten Technologie, Fragen nach Funktionsumfang, Details über Systemreaktionen, Schnittstellen zwischen Systemen und Benutzer müssen erst grundlegend analysiert werden. Erste methodische Ansätze zu Lösungen dieser Fragen wurden in [Dei01, FHP⁺05, Gei05, MkFN05, ABS⁺07] bearbeitet. Aber auch hier fehlt eine verbindende Systematik für die erwähnten Herausforderungen und Lösungskonzepte (Kapitel 1.2) von den frühen Phasen der Entwicklung bis zur Serienentwicklung - Start of Production (SOP). Für die frühen Phasen gelten zudem nach [WA07, AO08, AOK08] folgende Rahmenbedingungen:

1. Anforderungsumfang ist nicht abschätzbar
2. Technische Machbarkeit ist nicht nachweisbar
3. Hard- und Softwareumfang ist nicht definierbar
4. relevante Stakeholder sind nicht bekannt
5. Kundenstudien (Akzeptanztests) können nicht durchgeführt werden
6. Markteinsatz ist auf Grund fehlender Studien nicht kalkulierbar
7. Prozesse und Methoden sind nicht etabliert
8. Aufwand für die Erstellung und Wartung von Dokumenten ist zu hoch

Die hier aufgeführten Rahmenbedingungen beschreiben ein Spannungsverhältnis zwischen Praxis und Theorie [Som07, SS97]. Dieser Sachverhalt ist eine besondere Herausforderung für die zu entwickelnde Systematik, die gleichfalls eine Antwort auf die Frage nach der Notwendigkeit des Einsatzes von Prozessen und Methoden in den frühen Phasen beantworten wird. [All08]. Eine dritte Kraft im Spannungsverhältnis zwischen Software Engineering Theorie und gelebter Praxis ist die Notwendigkeit eines Leitfadens. Dieser Leitfaden wird von den Entwicklern benötigt, um bei der Entwicklung neuer Applikationen das Spannungsverhältnis zwischen Nachweis des Anwendungsnutzens und den Risiken der Entwicklung abzuwägen. Analog der skizzierten C2X-Einführung müssen Kundennutzen (Hat der Kunde einen Mehrwert, für den er bezahlen würde?) und Entwicklungsrisiken

¹²Prozessphasen siehe Anhang A.1, Abbildung A.1

¹³Gartner Group: 75% heutiger Projekte scheitern auf Grund von Fehlern in der Analyse [Ins08]

¹⁴Standish Group: Chaos Report 1995-2008 siehe www.standishgroup.com