

Automotive Software Engineering Optimierte Entwicklungsmethoden

Dipl.-Ing. Tibor Farkas
Fraunhofer-Institut für
Offene Kommunikationssysteme (FOKUS)
Kaiserin-Augusta-Allee 31
D-10589 Berlin, Germany

Dieser Vortrag gibt zunächst eine kurze Übersicht über einige der Innovationstreiber im Umfeld von Car-To-X Anwendungen und erläutert die zukünftigen Herausforderungen für das Automotive Software Engineering. Wie sich klassische Entwicklungsprozesse im Automotive Software Engineering durch den Einsatz von modellbasierten Vorgehensweisen weiter optimierten lassen, wird vertiefend anhand von Beispielen aus der industriellen Forschung konkretisiert. Im Vortrag stehen Methoden zur Effizienzsteigerung und Kostenreduktion im Mittelpunkt, welche heutzutage die Spezifikation und Entwicklung von Onboard-Software entscheidend prägen. Fahrzeugfunktionsmodelle für mehr Sicherheit, Werkzeugkopplungen für durchgängige Prozesse und die virtuelle Produktentwicklung im multidisziplinären Umfeld sind hierbei wesentliche Aspekte. Abschließend wird ein werkzeugübergreifendes Verfahren zur Qualitätssicherung und ein Softwarewerkzeug zur Erhöhung von Qualität bei gleichzeitiger Kostenreduktion durch automatisierte Prüftechnologie im Automotive Software Engineering kurz vorgestellt.

Neue Innovationen und Komplexität

Software in modernen Fahrzeugen realisiert zunehmend Sicherheits- und Komfortfunktionen und wird zukünftig ein immer wichtigerer Innovationstreiber in der Automobilentwicklung. Kommende Fahrzeuggenerationen werden schon bald mit bis zu 1 Gigabyte Onboard-Software ausgerüstet sein. Aktive Sicherheit, Verbrauchsreduktion und Komforterhöhung realisiert softwareintensive Elektronik bereits schon im Mittelklasse- und Kleinwagen-Segment. Elektronische Systeme im Automobil werden für die Differenzierung der Marken vor dem Kunden ebenfalls immer wichtiger. Besonders den Fahrer unterstützende, die Fahrer- und Fahrzeugsicherheit erhöhende sowie kommunikations- und multimediale elektronische Innovationen prägen die Wahrnehmung der Kunden.



**Abbildung 1: Innovation durch Kommunikation -
Neue Anwendungen und Geschäftsmodelle durch Car-To-X.**

Um die immer größer werdende Anzahl von Steuergeräten und Systemen wirtschaftlich, effizient und robust miteinander zu vernetzen, bedarf es optimierte und flexible Entwicklungsprozesse für das Automotive Software Engineering. Eine frühzeitige Absicherung der Komplexitätsbewältigung solcher Elektroniksysteme sowie die Prozessoptimierung in der Entwicklungskette stellen neue Herausforderungen an den Produktenstehungsprozess. Von größter Wichtigkeit sind dabei sorgfältig ausgewählte Entwicklungsmethoden und Softwarewerkzeuge, da nur effiziente und fehlervermeidende Prozesse diesen Wettbewerbsvorteil generieren.

Die klassischen Methoden der Softwaretechnik zur Entwicklung anspruchsvoller Onboard-Softwaresysteme für Infotainment-Anwendungen oder komplexe Sicherheits-Anwendungen werden den immer steigenden Herausforderungen an die Entwicklung eingebetteter Steuergeräte im Automobil zukünftig nicht mehr gerecht (Abbildung 2).

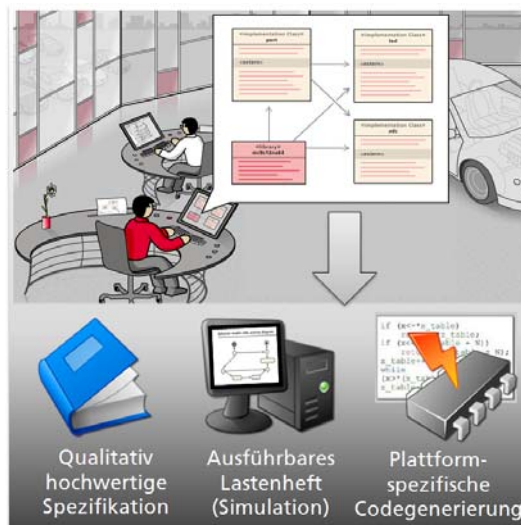
- [illegible]

Anforderungsdokumentation und Systemspezifikation nehmen in ihrem Umfang immer weiter zu. Ein anschauliches Beispiel: Gedruckt sind die Dokumente für komplexe Fahrzeugfunktionen, wie beispielsweise für den Antriebsstrang, schon bis zu 1,3 Meter hoch. Hieraus müssen

Zulieferer die für sie essentiell relevanten Informationen zunächst einmal herausfinden. Aus diesen werden dann Funktionen in Software (je nach Mikrokontroller in Assembler, C oder C++) entwickelt, welche selbst an Codezeilen stark anwachsen. Qualität-Reviews sind hierdurch zeitaufwendig und mühsam, was insgesamt die Fehleranfälligkeit erhöht. Wiederverwendung geschieht von Generation zu Generation evolutionär. Durch diese Vorgehensweise ist wenig Innovation möglich. Unterschiedliche Varianten werden durch „Copy & Paste“ von Codezeilen oder Klassen hergestellt. Da immer noch Steuergeräte sehr plattformspezifisch (hardware- und ressourcenanhängig) programmiert werden müssen, ist wenig Flexibilität im Software Engineering gegeben. Hier wird AUTOSAR (AUTomotive Open System ARchitecture) neue Möglichkeiten bieten. AUTOSAR ist der neue Standard für Software-Architekturen, der sich in den kommenden Kraftfahrzeug-Generationen etablieren wird. Durch sein Architekturkonzept und die AUTOSAR-Laufzeitumgebung bietet er hier deutliche Vorteile. Jedoch bleiben viele Fragen bei der Einführung des Standards offen: „Wie geschieht eine erfolgreiche Umsetzung bzw. Migration von bestehenden Entwicklungen? Wie werden sich die Methoden und die Werkzeuge im Automotive Software Engineering ändern (müssen)?“

Als eine erfolgsversprechende Alternative wird daher seit einiger Zeit bei Herstellern und Zulieferern die modellbasierte Entwicklung von Fahrzeugfunktionen (Abbildung 3) angewendet. Diese Vorgehensweise unterstützt insbesondere das funktionsorientierte Design von Steuergerätesoftware. Hierbei wird auf Basis des ursprünglichen Lasten- und Pflichtenheftes ein ausführbares, digitales Modell erstellt, welches als zentraler Ausgangspunkt für alle nachfolgenden Entwicklungsschritte dient. In verschiedenen Verfeinerungsschritten können Verhaltensmodelle, Umgebungsmodelle, Test- und Implementierungsmodelle entwickelt werden, woraus sich der ausführbare Seriencode (vollständig oder teilweise) mit zertifizierten Compilern automatisch erzeugen sowie entsprechende Testbeschreibungen ableiten lassen.

- Der **Produktenstehungsprozess** wird zukünftig durch **Funktionen** bestimmt.
- Die Umsetzung der **Funktionslogik** bzw. Algorithmik wird in ausführbaren **Modellen** beschrieben (Lastenheft).
- **Abstraktion** von Plattformaspekten und Ressourcen schafft Flexibilität.
- Frühe **Fehlervermeidung** durch die Umsetzung des Lastenhefts als Modell schon im Vorfeld.
- **Wiederverwendung & Varianten** durch Komponenten und Patterns.
- Automatische **Codegenerierung** mit zertifizierten Compilern und Diagnosetools erhöhen die **Qualität**.



**Abbildung 3: Der funktionsorientierte Entwicklungsprozess -
Ideale Unterstützung durch modellbasierte Vorgehensweise.**

Durch Modelle sind Ressourcenbeschränkungen nicht mehr im Vordergrund; die Abstraktion von Plattformaspekten schafft hierdurch Flexibilität. Die Wiederverwendung von verschiedenen Funktionsvarianten wird durch Modell-Komponenten (partitionierte Funktionsbausteine) und Modell-Patterns (Design-Vorlagen) unterstützt.

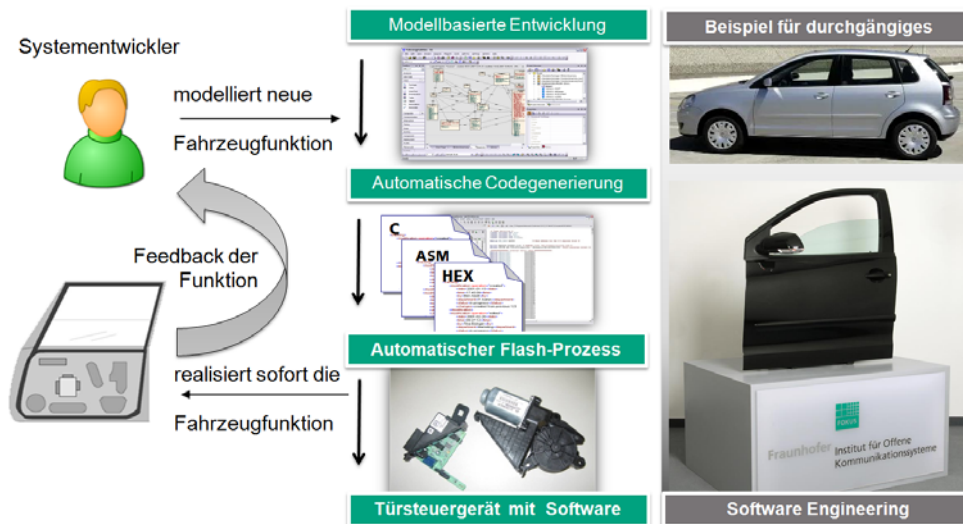


Abbildung 4: Durchgängigkeit mit integrierten Entwicklungswerkzeugen - Fahrzeugfunktionsentwicklung mit UML+Simulink+CodeWarrior.

Wesentliche Forschungsfragen in der modellbasierten Entwicklung betreffen die Definition und die Verwaltung der Anforderungen in Relation zum Modell, die Absicherung der Codegenerierung bei hochoptimierenden Codegeneratoren, geeignete Strategien für den entwicklungsbegleitenden modellbasierten Test, sowie eine prozessübergreifende Methodik, welche auf domänenspezifischen Informationsmodellen basiert.

Da im Automotive Software Engineering viele unterschiedliche Werkzeuge verschiedener Hersteller Anwendung finden, die teils durch die gewählte Hardwareplattform bestimmt sind (Compiler, Debugger, Flashtool etc.), können in der modellbasierten Entwicklung immer noch Lücken auftreten. Um dies zu vermeiden, wurden für eine durchgängige modellbasierte Entwicklung in den Forschungsarbeiten am Fraunhofer-Institut FOKUS Werkzeugketten unterschiedlicher Werkzeuge aufgebaut. Durch Tool-Adapter verbunden, erlauben diese einen reibungsfreien Entwicklungsprozess auch mit hochspeziellen Entwicklungsumgebungen, wie z.B. Freescale CodeWarrior® und HIS-konforme Flashtools (Abbildung 4).

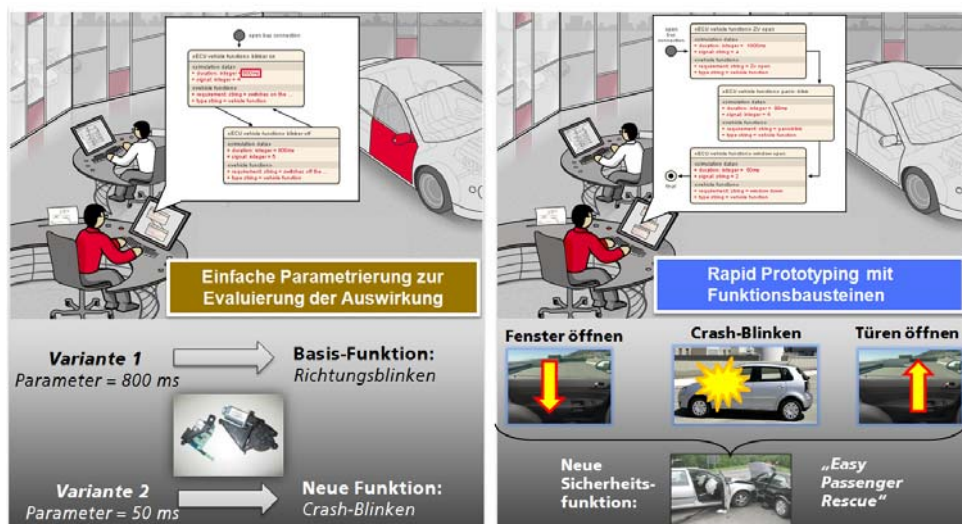


Abbildung 5: Verhaltensmodellierung von Sicherheitsfunktionen - Wechselwirkungsanalyse und Rapid Prototyping.

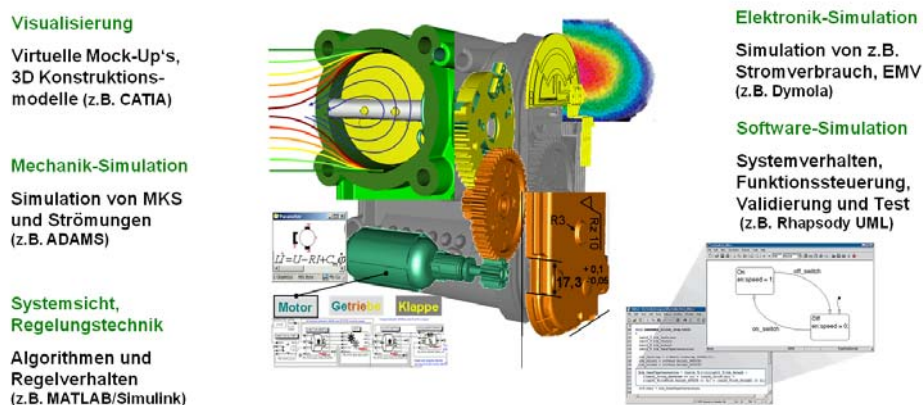
Neben proprietären Modellierungswerkzeugen wie MATLAB/Simulink® wurde insbesondere auch die offene Sprache UML (Unified Modeling Language) für das Architekturdesign der

Automotive Software angewandt. Bereits existierende C Code Funktionen oder Bibliotheken für den gewählten Mikrocontroller-Typ können in das Modell eingebunden (referenziert) werden. Aus den Modellen entsteht eine plattformspezifische C/C++ Code Generierung vollständig automatisiert. Die Kopplung von UML-Modellierungswerkzeug mit mikrocontrollerspezifischem Linker und Compiler sowie der Software für den Flash-Prozess, ermöglicht das sofortige Aufspielen der modellierten Fahrzeugfunktion auf das Steuergerät. Dieses modellbasierte Verfahren erlaubt die schrittweise Migration auf die modellbasierte Entwicklung, die Migration zu AUTOSAR und auch die Anwendung von offenen Standards, wie der UML.

Mittels Simulationsmodellen kann die geflashte Fahrzeugfunktion realitätsnah am eingebetteten System – hier am Beispiel Türsteuergerät – durch einfache Parametrierung evaluiert werden. Auswirkungen und Wechselwirkungsanalysen sind somit direkt am Steuergerät erkennbar (Abbildung 5). Neue Funktionen können hierdurch mit weniger Zeitaufwand frühzeitig erprobt werden: Durch Parameteränderungen kann z.B. ein Richtungsblinker als ein ‚Crash-Blinker‘ abgeändert werden und vorhandene Funktionen, wie ‚Fenster öffnen‘, ‚Crash-Blinker‘ und ‚Zentralverriegelung öffnen‘ können modellbasiert zu einer neuen Funktion kombiniert werden (hier als fiktives Beispiel: *Easy Passenger Rescue - EPR*). Die so entstandene, neue Sicherheitsfunktion EPR kann beispielsweise zur einfacheren Bergung der Insassen aus einem verunfallten Fahrzeug eingesetzt werden, nachdem die Airbags in einem Crash ausgelöst wurden: Tür entriegelt, falls verschlossen; Fenster fährt schnell hinab; Crash-Blinker warnt auffahrende Fahrzeuge. Der Systementwickler erhält sofort die Möglichkeit, seinen Entwurf in der realen Umgebung bzw. an Aktuatoren – angeschlossen an einem Steuergerät – sichtbar zu machen.

Sicherheitsrelevante Funktionen sind oft in mechatronischen Systemkomponenten implementiert. Die mechanischen Vorgänge werden von Software gesteuert, die auf elektronischen Bauteilen ablaufen. Immer komplexer werdende elektronische Produkte, wie Drive-By-Wire-Systeme, führen in den Entwicklungsprozessen zum vermehrten Einsatz unterschiedlichster Simulationswerkzeuge. Diese ermöglichen auf der Basis von Modellen eine frühzeitige Absicherung von Produkteigenschaften im Entwicklungsprozess und somit einen Zeitvorteil zur Minimierung von Entwicklungskosten.

Motivation für eine **frühzeitige Simulation mit virtuellen Prototypen** ist es, einen Zeitvorteil und damit eine gravierende Kostenreduktion zu erwirken.



Quelle: FunctionalDMU – Initiative der Fraunhofer Gesellschaft, URL: www.functionaldmu.org

Abbildung 6: Functional Digital Mock-Up - Virtuelle Produkte werden erlebbar.

Für die einzelnen Domänen, wie Mechanik-, Elektronik- und Software-Engineering, kommen unterschiedliche und spezialisierte Simulatoren zum Einsatz (Abbildung 6). Erst eine Kopplung dieser Simulatoren ermöglicht eine domänenübergreifende Simulation der mechatronischen Komponente.

Zur geometrischen Integration von Produktmodellen haben sich dreidimensionale Konstruktionsmodelle – Digital Mock-Up's (DMU) – in der industriellen Praxis als fester Bestandteil der virtuellen Produktentwicklung etabliert (Abbildung 7). Allerdings sind DMU und

die dazugehörigen Softwarewerkzeuge heute weitgehend auf die Integration der Geometrie beschränkt; es fehlt hier an Möglichkeiten zur funktionalen Integration.

Die **digitale Produktentwicklung** ermöglicht es, Software-Verhaltensmodelle von Fahrzeugfunktionen multidisziplinär durch kooperative Simulation zu evaluieren.

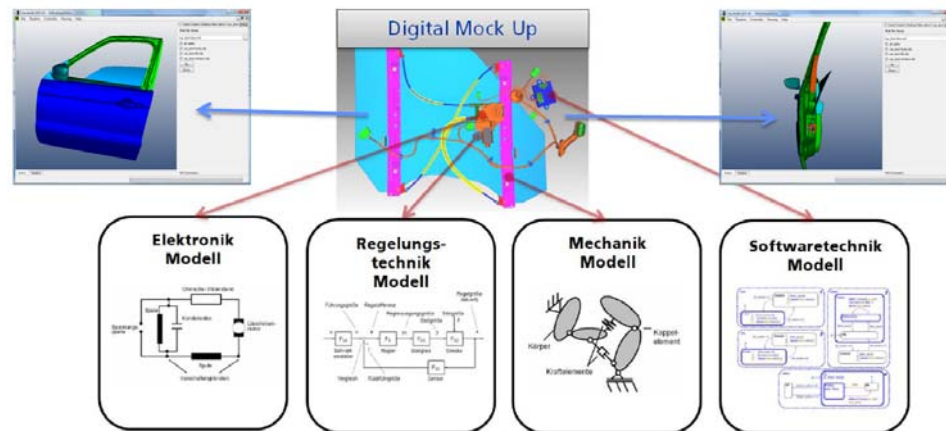


Abbildung 7: Visualisierung von Automotive Software - Zusammenspiel multipler Systemkomponenten virtuell verifizieren.

Das Funktionsverhalten einer sicherheitsrelevanten Software, wie beispielsweise die Erkennung des Einklemmfalls einer Einklemmschutzfunktion des Fensterhebers bei unterschiedlichen Reibwerten der Türdichtung, kann mit der digitalen Produktentwicklung multidisziplinär durch kooperative Simulation mittels FunctionalDMU evaluiert werden.

Das laufende Fraunhofer Forschungsprojekt FunctionalDMU ermöglicht das Zusammenspiel multipler Systemkomponenten und macht das Systemverhalten für Entwicklungsingenieure virtuell darstellbar. Der prinzipielle Lösungsansatz von FunctionalDMU liegt in einer Erweiterung von 3D-Konstruktionsprototypen (DMU) um Verhaltensmodelle, die in verschiedenen Modellierungssprachen für unterschiedliche Simulatoren existieren können.

Wechselwirkungen der funktionalen Grundelemente eines mechatronischen Systems werden in einem übergeordneten **Simulationsmodell** (SysML) beschrieben.

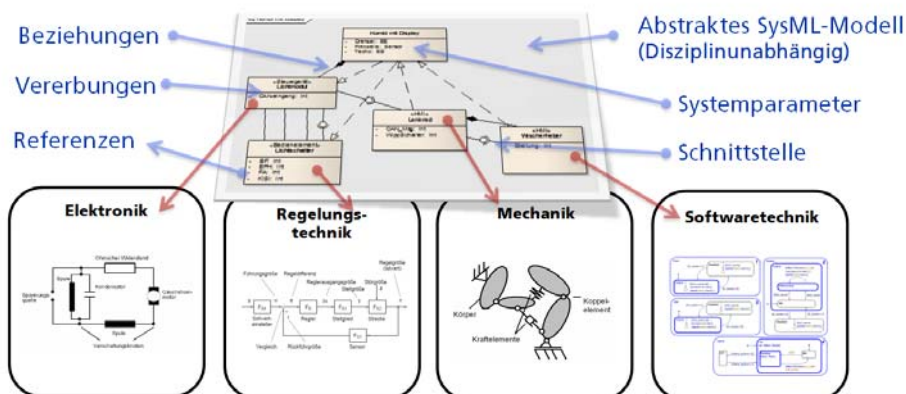
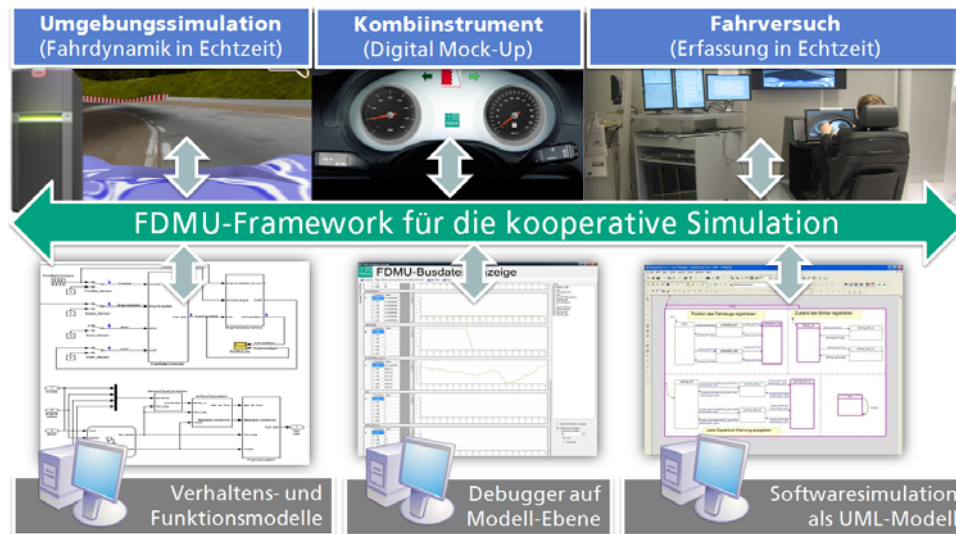


Abbildung 8: Domänenübergreifende Systementwicklung - Modellierung interdisziplinärer Zusammenhänge.

Verhaltensmodelle eines mechatronischen Systems liegen in der Systementwicklung zumeist in einer domänenspezifischen Modellierungssprache vor. Typischerweise ist diese domänenspezifische Modellierungssprache nochmals sehr stark herstellerabhängig geprägt. Ein Beispiel zeigt die herstellertypische Repräsentation für signalflussorientierte Blockdiagramme in MATLAB/Simulink®. Für eine übergreifende Beschreibung von Simulationsparametern, Schnittstellen, Beziehungen ist ein übergeordnetes Modell erforderlich, wodurch sich

unterschiedliche Modellsprachen (technisch) referenzieren lassen. Die Systems Modeling Language (SysML) ist solch eine grafische Modellierungssprache für die Systemtechnik (Abbildung 8). Die SysML wird in FunctionalDMU für die Spezifikation von funktionalen Bausteinen (FunctionalBuildingBlocks) und übergeordneten Simulationsmodelle angewandt.



**Abbildung 9: Frühzeitige Absicherung des Funktionsverhaltens -
FDMU-Software Framework für Integration und Evaluierung.**

Die Kopplung der unterschiedlichen Verhaltensmodelle und ihrer Simulatoren erfordert technologisch ein Framework (Abbildung 9), das die Verhaltensmodelle in unterschiedlichen Simulatoren, wie z. B. Adams®, SimPack®, Dymola®, MATLAB/Simulink®, AdvanceMS®, Rhapsody® ausführt und den Datenaustausch zwischen diesen Simulatoren koordiniert, synchronisiert und darstellt.

Mittels des FunctionalDMU-Frameworks und Wrappern für Simulatoren wird funktionales Design Review komplexer Mechatronik möglich und aufwandstechnisch beherrschbar. Ein frühes Erkennen von Problemen im Zusammenspiel der mechatronischen Komponenten führt zur Verkürzung der Entwicklungszeit. Die Auswertung von (Parameter-)Studien, im Hinblick auf die Wechselwirkung zwischen mechatronischen Komponenten, ermöglicht domänenübergreifend bessere Entwicklungsergebnisse.

Qualitätssteigerung

An das Automotive Software Engineering werden bereits schon in den frühen Entwicklungsprozessen zukünftig immer hochgradigere Qualitätsanforderungen gestellt, die zum einen aus den sicherheitskritischen Einsatzbereichen und zum anderen aus international standardisierten Richtlinien und Bestimmungen resultieren. Die Bedeutung der Qualitätssicherung nimmt derart zu, dass die reinen Nachweise bezüglich der Softwarequalität durch Funktionstests zukünftig nicht mehr ausreichen werden. Aus einem der etablierten ISO Standards (ISO/IEC 9126-1:2001; Software Engineering - Product Quality) für Software Qualität (Abbildung 10) ersieht man, dass neben der reinen Funktionsabsicherung noch weitere Qualitätsmerkmale wie Zuverlässigkeit, Benutzbarkeit, Effizienz, Änderbarkeit und Übertragbarkeit die Gesamtqualität von Automotive Software entscheidend beeinflussen. Besonders in den nicht funktionalen Eigenschaften liegt ein hohes Potential an Fehleranfälligkeit in sicherheitskritischen Anwendungen.

- Eingebettete Systemsoftware übernimmt zunehmend **sicherheitsrelevante Aufgaben** und hat damit zunehmenden Einfluss auf Menschenleben.
- **Sicherheit** bedingt daher eine **qualitativ sehr hochwertige** Softwareentwicklung.

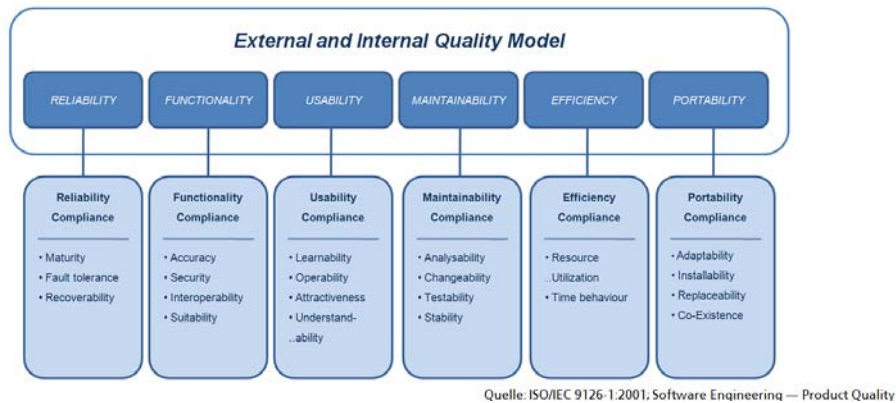


Abbildung 10: Qualität beginnt in den frühen Entwicklungsprozessen.
Sicherheit ist nur durch zuverlässige Software gewährleistet.

Für Prozesse fordern Qualitäts- und Reifegradmodelle zudem Methoden zur strategischen Fehlervermeidung, zur stetigen Effizienzsteigerung sowie der Optimierung von Entwicklungsprozessen und Arbeitserzeugnissen bei gleichzeitiger Kostenreduktion. Aus dem drastischen Zuwachs an Komplexität von Software in eingebetteten Systemen resultiert ein akuter Handlungsbedarf insbesondere für die Absicherung von Qualitätsmerkmalen, da diese Systeme häufig einem starken Kostendruck unterliegen. Standards der Qualitätssicherung zur funktionalen Sicherheit (wie z.B. ISO/IEC 26262), prozessspezifische Qualitäts- und Reifegradmodelle (wie CMMI oder Automotive SPICE) aber auch werkzeugspezifische Entwicklungsrichtlinien (wie Guidelines des MathWorks Automotive Advisory Boards) liegen meist in textueller Form als Katalog in den Entwicklungsabteilungen vor (Abbildung 11). Diese beherbergen eine Vielzahl an Entwicklungsrichtlinien, wobei sich Entwicklungsrichtlinien wechselseitig einschließen oder gar ausschließen können.

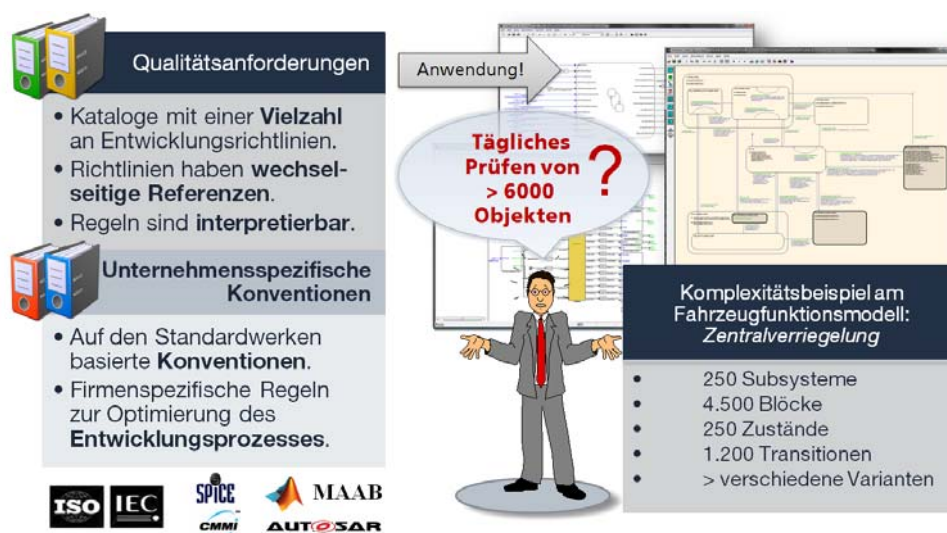


Abbildung 11: Qualitätssicherung wird zunehmend komplexer.

Manuelle Überprüfungen sind fehleranfällig und kostenintensiv.

Zudem sind Regeln in textueller Form interpretierbar oder oftmals nicht ausreichend formuliert, wie beispielsweise: „Es soll eine sinnvolle Namenskonvention gewählt werden“. Aufbauend auf diesen Standardkatalogen liegen den Entwicklern meist weitere firmenspezifische Regelwerke mit Konventionen sowie Best-Practices zur Optimierung der Prozesse und Artefakte vor.

Ein hohes Qualitätsniveau zur Einhaltung der Standards und stetigen Verbesserung der Entwicklungsprozesse erfordert jedoch, dass die Richtlinien auch gelebt werden und in den Prozessen und Artefakten Anwendung finden. Problematisch bezüglich der Umsetzung erweist sich jedoch, dass die zur Verfügung stehenden Werkzeugketten im Automotive Software Engineering keine Konsistenz der jeweils erzeugten Artefakte ausreichend sicherstellen können. Als Folge dieses Sachverhalts ist häufig mit Problemen wie Schnittstelleninkonsistenz oder fehlerhaft bzw. nicht nachvollziehbar umgesetzter Anforderungen zu rechnen. Bisher sind Systementwickler, die sich um Einhaltung der Konsistenz von Entwicklungsartefakten (z.B. in Dateien, Modellen oder Dokumenten) bemühen, auf sich alleine gestellt.

Das Erfüllen von Konsistenzkriterien kann von ihnen oftmals nur durch mühsames Vergleichen (Review) einer großen Menge von Daten erreicht werden. Diese Tätigkeiten geschehen manuell, da sie typischerweise einen Übergang zwischen Artefakten oder Prozessen betreffen, die oft mit verschiedenen Werkzeugen erstellt wurden. Bei immer komplexeren Entwicklungsartefakten, wie ein Komplexitätsbeispiel am Fahrzeugfunktionsmodell Zentralverriegelung verdeutlicht, sind Entwickler mit manuellen Prüfungen überfordert und brauchen einen zu hohen Zeitaufwand, was hohe Kosten verursacht. Auch ist es menschlich, dass Defekte übersehen werden und so die Fehlerrate nicht weiter gesenkt werden kann. Hierzu bedarf es an innovativen Technologien zur automatischen Prüfung von Entwicklungsartefakten.

Integrierte **Regel-Checker** zur automatisierten Prüfung minimieren Kosten & Fehler.

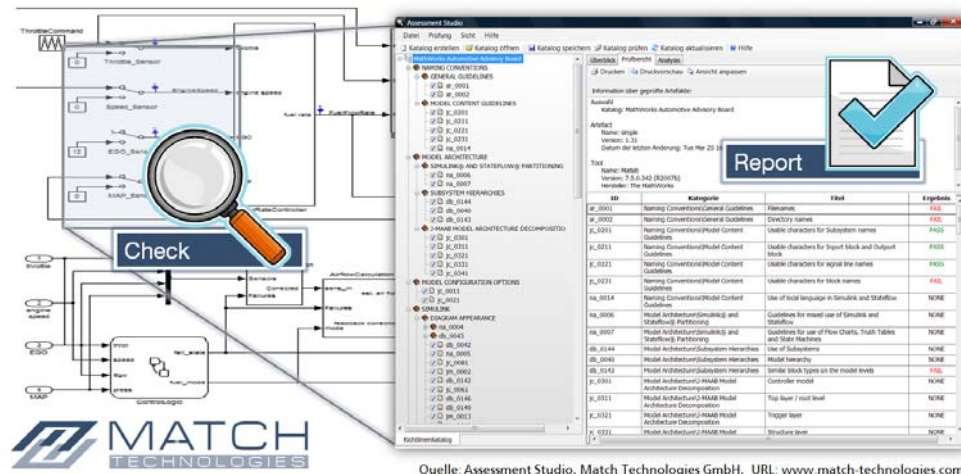


Abbildung 12: Automatische Erhöhung der Qualität bei gleichzeitiger Kostenreduktion.

Regel-Checker für die automatisierte Prüfung (Abbildung 12) können zur strategischen Fehlervermeidung sowie Optimierung von Entwicklungsprozessen und Arbeitserzeugnissen im Automotive Software Engineering eingesetzt werden. Sie schaffen eine sofortige Kostenreduktion, da aufwendige manuelle Reviews entfallen. Hierbei werden die textuellen Richtlinien aus den Qualitätsnormen bzw. Richtlinienkatalogen als computerauswertbare Ausdrücke (Constraints) mithilfe einer werkzeugunabhängigen Regelsprache umgesetzt. Die Software analysiert Dokumente, Modelle sowie andere Datenquellen und erstellt einen eindeutigen Prüfbericht sowie eine Performanceanalyse, sodass bei späteren Assessments auch eine Verbesserung nachgewiesen werden kann.

Qualitätsanforderungen sind angewendet auf Prozesse oder Entwicklungswerkzeuge und Artefakte oftmals übergreifend. Fragestellungen wie: „Wurde unsere einheitliche Namenskonvention in den Anforderungen sowie in den Modellen und Softwarefunktionen umgesetzt?“, „Wurden alle Systemkomponenten aus den Modellen getestet?“ bedingen eine umfassendere Analyse.

Durch **werkzeugübergreifende Prüfungen** können Artefakte aus unterschiedlichen Werkzeugen in wechselseitiger Beziehung geprüft werden.

Automatisierte Qualitätsprüfung:

- Anforderungen
- Office-Dokumente & Projektpläne
- Modellbasierte Entwicklung
- Simulationsdaten
- Softwaremodellierung
- Softwareentwicklung
- Diagnose- und Testwerkzeuge

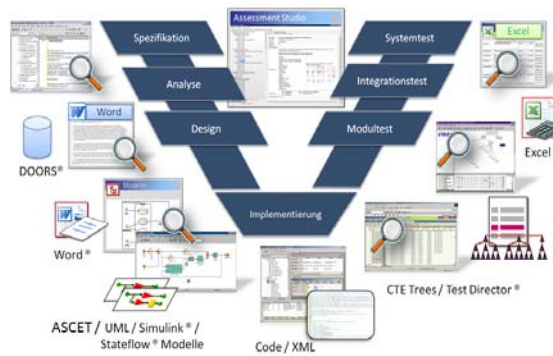


Abbildung 13: Qualitätsanforderungen sind oftmals werkzeugübergreifend.

Bezüglich der umfassenden Fehlervermeidung im Produktentstehungsprozess (Abbildung 13) erweist sich als zusätzlich schwierig, dass die dort zur Verfügung stehenden Werkzeugketten keine übergreifende Konsistenz der erzeugten Dokumente und Artefakte sicherstellen können. Konsistenzprüfungen werden gegebenenfalls von Entwicklern lokalisiert und semi-automatisiert in den einzelnen Applikationen durch Batch-Skripte in der jeweils proprietären Programmiersprache des Werkzeugherstellers durchgeführt. Aufgrund dieses Sachverhalts sind auch hier in der Praxis häufig Probleme durch Schnittstelleninkonsistenz zwischen Abteilungen, fehlerhaft bzw. nicht nachvollziehbar umgesetzte Anforderungen, keine Standardisierung der Richtlinien und fehlende Durchgängigkeit zu beobachten.

Aus den mehrjährigen Forschungsarbeiten am Fraunhofer-Institut FOKUS entwickelt, sind heute für eine mehrere Phasen umfassende Qualitätssicherung technische Lösungen (wie das Werkzeug *Assessment Studio* aus der Abbildung 12) bereits am Markt verfügbar. Diese basieren auf offenen technischen Standards (wie XML) und können sofort auf verschiedenste Datenformate im Automotive Software Engineering angewendet werden: z.B. auf das Requirements-Interchange-Format (RIF), auf AUTOSAR XML-Beschreibungsdateien, Beschreibung von Diagnosedaten in ODX (Open Diagnostic Data Exchange, auf UML/SysML basierte XML-Metadata-Interchange (XMI) Formate und weitere.

Die beim Fraunhofer-Institut FOKUS vorliegende Expertise und innovative Technologien zur modellbasierten Softwareentwicklung und Softwaretestens basieren auf der Adaption von neuen Technologien und etablierten Standards aus der Informations- und Kommunikationstechnik, speziell angepasst für den Bereich des Automotive Software Engineerings. Modellbasierte Entwicklung und Testmethoden werden unter dem Aspekt der qualitätsorientierten Entwicklung an Prototypen im FOKUS Automotive-Lab aufgebaut und in Zusammenarbeit mit der Industrie evaluiert (Abbildung 14). Die maßgebliche Produktivitätssteigerung im Automotive Software Engineering und die Erhöhung von Zuverlässigkeit und Robustheit der Systeme können so praxisnah erprobt und zu fertigen Produktlösungen mit Industriepartnern ausgebaut werden.

FOKUS Automotive Lab – Innovation aus Forschung- und Entwicklung erleben

- Optimierung modellbasierter Methoden sowie Entwicklungs- und Testwerkzeuge.
- Durchgängigkeit von der Anforderungsanalyse bis hin zur Endabnahme.
- Zeitgewinn mittels Automatisierung durch Kopplung von Software zur höheren Performance von Entwicklungsschritten.
- Präventive Qualitätssicherung durch automatisierte Richtlinienüberprüfung.
- Innovatives Engineering: Konfigurations- und Variantenmanagement mit Modellen.

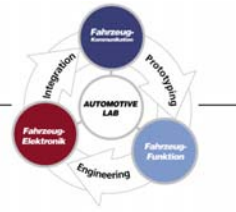


Abbildung 14: Automotive-Lab - Labor für Entwicklung und Demonstration innovativer Forschungsergebnisse.

Zusammenfassung

Die Systemkomplexität in der Fahrzeugentwicklung wächst durch den Einsatz von immer mehr softwareintensiver Automobilelektronik auch in den kommenden Baureihen weiter an. Fahrzeuge werden mit Ihrer Umgebung und dem Internet vernetzt, was einen deutlichen Anstieg an Informationsverarbeitung innerhalb der E/E-Systeme mit sich bringt. Ein starker Treiber für die Zunahme von Onboard-Software sind Infotainment-Anwendungen im Auto. Anwendungen für Sicherheit und Komfort werden sich immer mehr auf die Steuergerätkomplexität auswirken. Die Beherrschung der zukünftigen Systemkomplexität und hohe Qualitätsanforderungen im Automotive Software Engineering, bei hohen Sicherheits- und Zuverlässigkeitsanforderungen einerseits und der Kostendruck durch den existierenden Innovationswettbewerb andererseits, sind daher die zukünftigen Herausforderungen für Automobilhersteller und deren Zulieferer. Neue Standards wie AUTOSAR oder ISO/IEC 26262 werden erforderlich und betreffen das Automotive Software Engineering insbesondere. Diese Standards müssen jedoch erst behutsam in bestehende Entwicklungsprozesse von Seriensteuergerätesoftware eingeführt bzw. existierende Entwicklungen auf diese Standards migriert werden können.

Modellbasierte Entwicklungsmethoden werden zunehmend wichtiger für die optimierte Entwicklung softwareintensiver Systeme im Fahrzeug. Modelle sind hierfür ein adäquates und formales Mittel, da sie computerauswertbar und somit auch ausführbar sind. Der Einsatz erlaubt die Präzisierung einer textuellen Spezifikation (Anforderungen) und aufgrund der Standardisierung eine einheitliche Repräsentation.

Komplexitätsbeherrschung durch alternative Sichten auf Architektur, Struktur und Verhalten sind ebenso möglich, wie eine frühzeitige Funktionsprüfung durch Simulation und modellbasierten Test. Aus den Modellen lässt sich Steuergerätecode durch automatische Codegenerierung mittels zertifizierter Compiler auch für sicherheitskritische Anwendungen einsetzen. Eine systematische Wiederverwendung auf der Basis von Modellbibliotheken und Patterns schafft den notwendigen Zeitgewinn. Interdisziplinäre Simulationen mit virtuellen Prototypen erlauben die visuelle Darstellung des Verhaltens von Automotive Software an 3D-Modellen. Die modellbasierte Funktionsentwicklung ist heute bereits technisch ausgereift, aber im Produktenstehungsprozess der Steuergeräte noch deutlich unterrepräsentiert.

Die Entwicklung von sicherheitsrelevanter Automotive Software unterliegt einer kontinuierlichen Sicherstellung der Produktqualität und Systemzuverlässigkeit. Bei der Modellierung funktions sicherer Anwendungen kommen daher eine Vielzahl an unterschiedlichen Richtlinien, Standards und Normen bezüglich Qualitäts-, Sicherheits- und

Zuverlässigkeitsabsicherung sowie rechtlicher Bestimmungen der Gesetzgebung innerhalb der Automobilindustrie in Betracht. Integrierte Automatismen zur Qualitätsprüfung und insbesondere formale sowie werkzeugunabhängige Notationen für Richtlinien zur Normenprüfung, sind derzeit noch unzureichend umgesetzt. Hierbei können aktuelle Forschungsergebnisse, wie computerausführbare Richtlinien und Prüfwerkzeuge, den Entwicklungsprozess aktiv unterstützen. Durch werkzeugübergreifende Prüfung können Artefakte aus den unterschiedlichen Werkzeugen einzeln oder in wechselseitiger Beziehung analysiert werden. Prüfprotokolle ermöglichen dabei die rasche Identifikation von Fehlerquellen und ein Nachweis der eigenen Prozessperformance.

Im Automotive-Lab des Fraunhofer-Instituts FOKUS werden innovative Technologien entwickelt, aktuelle Forschungsergebnisse aus dem Automotive Software Engineering ausgestellt sowie Szenarien und Prototypen aufgebaut, um in Zusammenarbeit mit der Automobilindustrie gemeinsam die Innovationen in den Fahrzeugen von morgen zu entwickeln.