

Siemens Digital Industries Software

Orchestrieren der Entwicklung von eingebetteten Anwendungen in der Automobilbranche

Definition und Planung von Anwendungen

Kurzdarstellung

Die Automobilindustrie erlebt derzeit einen rapiden Wandel – angetrieben durch eine ständig wachsende Nachfrage nach erweiterten Features, für die ausgefeilte Elektroniksysteme und eingebettete Software erforderlich sind – wie etwa Fahrerassistenzsysteme und vernetzte Fahrzeuge. Die Entwicklung von eingebetteten Anwendungen gleicht dem Versuch, sich in einem Labyrinth zurechtzufinden, in dem viele unterschiedliche Werkzeuge und Teams unter ständig wechselnden Bedingungen koordiniert werden müssen. Eine einheitliche Plattform für die Anwendungskonstruktion kann sicherstellen, dass Randbedingungen und Anforderungen vollständig und effektiv kommuniziert werden und dass die Entwicklung die Anforderungen exakt erfüllt.

Piyush Karkare

Einführung

Die Automobilindustrie ist eine hochdynamische Branche, die sich schnell verändert. Die ständig wachsende Nachfrage nach erweiterten Features, für die ausgefeilte Elektroniksysteme und eingebettete Software erforderlich sind – wie etwa Fahrerassistenzsysteme und vernetzte Fahrzeuge –, ist die treibende Kraft hinter diesen Veränderungen. Die Nachfrage ist so groß, dass sich die Kernkompetenzen der Autobranche von mechanischen Systemen weg hin zu Software und Elektronik verlagern. Differenzierung und Innovation im hart umkämpften Automobilmarkt hängen zunehmend von Software und Elektronik ab anstatt von mechanischen Systemen.

Autobauer und Zulieferer reagieren auf die digitale Transformation moderner Fahrzeuge, indem sie Softwarefunktionen in ihrer gesamten Angebotspalette standardisieren. Volkswagen kündigte vor Kurzem Pläne für eine neue Geschäftseinheit an, in der über 5.000 Softwareexperten u. a. ein einheitliches Betriebssystem für alle VW-Marken entwickeln werden (Automotive News Europe, 2019).

Auch neue Player aus dem Technologiesektor – wie z. B. Waymo und Apple – drängen auf den Auto- und Mobilitätsmarkt. Diese Unternehmen haben viel Erfahrung mit der Entwicklung und Erstellung leistungsstarker Softwarefunktionen, Anwendungen und Benutzeroberflächen. Gleichzeitig stehen Hunderte von jungen Unternehmen in den Startlöchern, um diesen Markt zu erobern – diese verknüpfen technologische und automobilspezifische Features gleich von Anfang an, um überzeugende Fahrzeugplattformen zu bauen. In beiden Fällen schafft die wachsende Bedeutung von Fahrzeugsoftware und -elektronik gleiche Spielregeln für alle – sowohl die neuen Unternehmen am Markt als auch traditionelle Automarken.

Zusätzlich zu diesen neuen Herausforderungen ist der Wettbewerb zwischen bereits etablierten Autobauern so stark wie nie zuvor. Im Jahr 2001 hatte der Ford Escape etwa sechs direkte Konkurrenten auf dem relativ kleinen Crossover-SUV-Markt. Heute gibt es in diesem Segment über 21 Modelle sowohl der Mittelklasse als auch der Ober- und Luxusklasse. Der Wettbewerb forciert den Bedarf an neuen Inhalten für Fahrzeuge (in Form neuer Funktionen), und dies wiederum erhöht die Komplexität.

Diese Komplexität manifestiert sich überwiegend in den Bereichen Elektrik und Elektronik (E/E) und Software. Dies lässt sich an der Anzahl von Anforderungen und Spezifikationen in diesen Bereichen ablesen, die deutlich höher ist als in anderen Gebieten (Abbildung 1). Für einen Mittelklassewagen mit guter Ausstattung

beispielsweise gelten etwa 50.000 mechanische Anforderungen und Rechtsvorschriften. Die Bereiche E/E und Software für dasselbe Fahrzeug schlagen dagegen mit bis zu 450.000 Anforderungen zu Buche. Die Prüf- und Testpläne für E/E und Software multiplizieren die Anzahl von Elementen, die mehrmals nachverfolgt werden müssen.

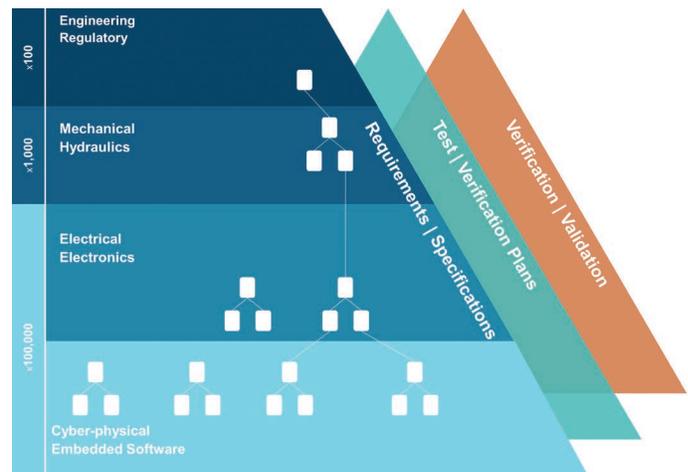


Abbildung 1: Die Anforderungen und Spezifikationen für die Bereiche Software und Elektrik/Elektronik sind deutlich komplexer und zahlreicher als in anderen Bereichen.

Die Komplexität ist schon innerhalb dieser Fachgebiete selbst überwältigend, und das bereichsübergreifende Durchsetzen von Anforderungen kann noch wesentlich komplizierter sein. Die Bereiche E/E, Software und Mechanik sind sehr eng miteinander verflochten und erfordern sehr viel Interaktion an sehr vielen Berührungspunkten. Anforderungen weisen häufig bereichsübergreifende Beziehungen auf, die unter Umständen sehr schwer zu verstehen und noch schwerer vorherzusagen sind. Es ist jedoch wichtig, dass alle Teams jede einzelne Konstruktionsanforderung verstehen und wissen, wie diese sich auf das System insgesamt auswirkt.

Während Automobilhersteller und Zulieferer versuchen, Prozesse und Wertschöpfungsketten aufzubauen, finden gleichzeitig Konstruktionsänderungen und Produktweiterentwicklung statt. Entscheidend ist hier die Fähigkeit, die Konsistenz und Qualität miteinander zusammenhängender Daten

fachgebietsübergreifend aufrechtzuerhalten. Der Schlüssel zum Erfolg ist die Sicherstellung von Konsistenz und Kompatibilität zwischen allen Arbeitsaufgaben, die von den verschiedenen Beteiligten über eine Vielzahl von Plattformen und Organisationen hinweg ausgeführt werden. Hersteller und Zulieferer müssen ihre Aktivitäten auf jeder Ebene der Designabstraktion koordinieren können – von den Fahrzeugfunktionen auf Plattformebene über die logische Zuordnung bis hin zur Entwicklung der eigentlichen Softwareanwendung.

Autobauer setzen verstärkt auf serviceorientierte Architekturen. Dieser Übergang macht sich in der Konsolidierung der großen Menge an hardwarebasierten Steuergeräten in eine geringere Anzahl von Steuerungseinheiten für die einzelnen Bereiche bemerkbar. Für Services und Funktionalität sorgen stattdessen Softwareanwendungen, die auf Abstraktionen der Steuergeräte portiert wurden. In solchen Architekturen muss sehr sorgfältig verfolgt werden, wo jeder einzelne Service ausgeführt wird, um korrekte Untergliederungen auf Anwendungsebene zu ermöglichen. Eine feature- und funktionsbasierte Architektur mit Untergliederungen in Services und Zuweisungen zu Steuergeräteabstraktionen ist ein effektiver, organischer und skalierbarer Ansatz für das Verfolgen von Änderungen und ihren Auswirkungen.

Die Koordinierung der Aktivitäten zwischen all diesen Teams und Abstraktionen kann eine ziemliche Herausforderung darstellen, und die Organisation der Anwendungsentwicklung ist besonders schwierig. Die Entwicklung von eingebetteten Anwendungen

bei der Fahrzeugfertigung gleicht dem Versuch, sich in einem Labyrinth zurechtzufinden, in dem viele unterschiedliche Werkzeuge und Teams unter ständig wechselnden Bedingungen koordiniert werden müssen. In dieser Umgebung stellt die Zusammenarbeit eine Herausforderung dar und führt zu Problemen. Heute stellen Softwaresysteme aufgrund der Komplexität und entscheidenden Bedeutung von Softwareanwendungen eine Hauptquelle für Programmrisiken dar. Softwarefehler und häufige Änderungen der Randbedingungen verursachen Produktionsverzögerungen, überzogene Budgets und steigende Garantiekosten.

Zum Zweck dieses White Papers unterteilen wir die Entwicklung von integrierten Softwareanwendungen in der Automobilbranche in folgende drei Segmente:

- Definition und Planung von eingebetteten Anwendungen
- Entwicklung und Qualitätssicherung von eingebetteten Anwendungen
- Auslieferung und Überwachung von eingebetteten Anwendungen

Im vorliegenden Dokument konzentrieren wir uns auf das erste Segment: Definition und Planung von eingebetteten Anwendungen.

Definition und Planung von Anwendungen

Einige Automobilhersteller haben bereits erfolgreich einen featurezentrischen, architekturbasierten Ansatz für die Fahrzeugsoftware eingerichtet. Bei einem solchen Ansatz startet die Phase der Definition und Planung von Anwendungen dann, wenn die Anforderungen, Spezifikationen und Randbedingungen für Features auf Fahrzeugebene vollständig definiert und in die Abstraktion der Softwareanwendung kaskadiert wurden. Diese Anforderungen, Spezifikationen und Randbedingungen definieren die Fahrzeugplattform bzw. die Konstruktionsabsicht der Features auf Fahrzeugebene (Abbildung 2). Beispielsweise definieren Konstrukteure die Spezifikationen und Randbedingungen für die Fahrzeugfeatures oder -funktionen, die Fahrerassistenzsysteme ermöglichen: Spurhalteassistenten, Abstandsregeltempomaten, Kollisionswarnsysteme und vieles mehr.

Danach untergliedern Softwareingenieure die Definitionen auf Funktionsebene in die Anforderungen und Definitionen auf Anwendungsebene. Diese wiederum werden in der nächsten Phase in spezifische Softwarefunktionen heruntergebrochen. In der Realität allerdings ist die Untergliederung von Softwarefunktionen sowie deren Standardisierung für verschiedene Fahrzeugplattformen ein kontinuierlicher Prozess, der sich durch die gesamte Entwicklung zieht (Abbildung 3). Softwareingenieure untergliedern fortwährend Anforderungen höherer Ebenen

SWF – Kunden-/Konstruktionsfeature
 SWf – Einzelfunktion – Software
 HWf – Einzelfunktion – Hardware

LC – logische Komponente
 PC – physische Komponente

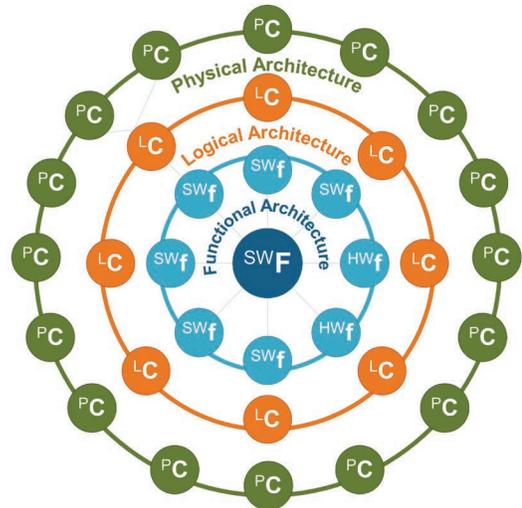


Abbildung 2: Softwarefeatures werden in Softwarefunktionen untergliedert, logischen Komponenten zugewiesen und physischen Komponenten mit Teilenummern zugeordnet.

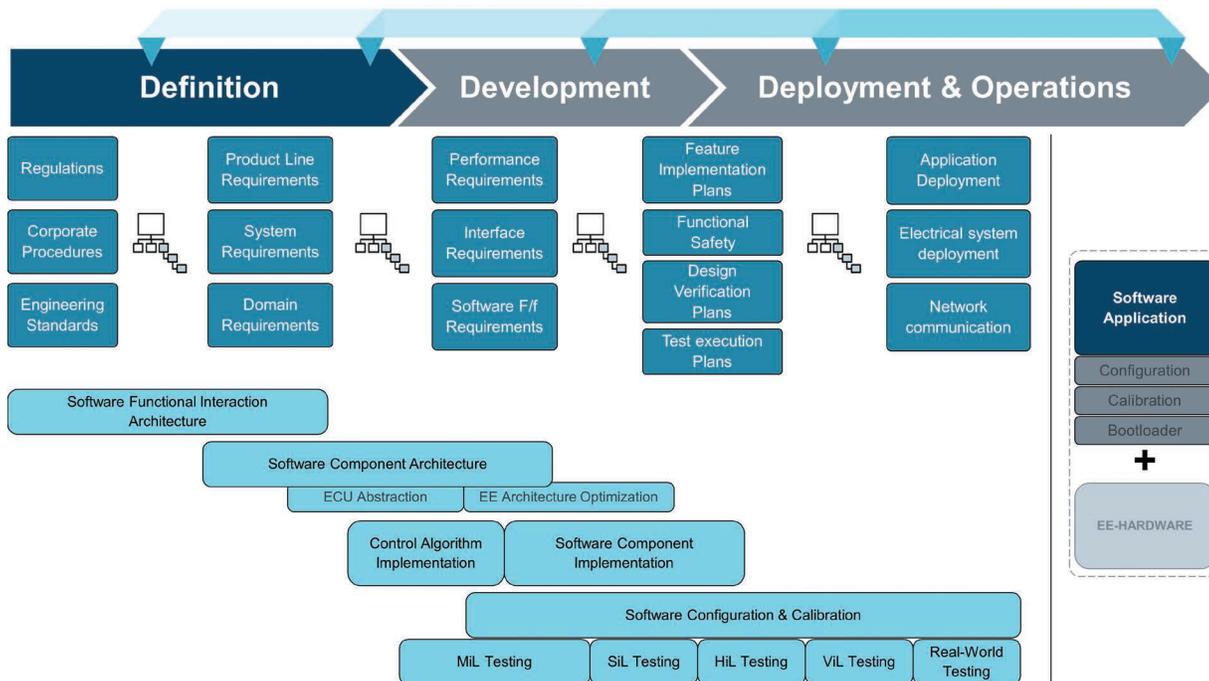


Abbildung 3: Die Untergliederung von Funktionen aus Anforderungen und Randbedingungen ist ein kontinuierlicher Prozess während der gesamten Entwicklung.

in Einzelfunktionen, um den Funktionssatz eines Fahrzeugs zu realisieren. Im Rahmen der Entwicklung optimieren und maximieren die Ingenieure die Konstanz und Interaktion zwischen den Funktionen, um sicherzustellen, dass diese für verschiedene Fahrzeugplattformen skaliert werden können. Dieser Ansatz ermöglicht es den Ingenieuren auch, Funktionen aus anderen Systemen oder Organisationen wiederzuverwenden.

In dieser Phase müssen auch Sicherheitskonzepte für die Anwendung berücksichtigt und Strategien für das Risikomanagement entwickelt werden. Hohe Sicherheitsanforderungen sowie Strategien für Verifizierung und Validierung werden auf Systemebene definiert und bis auf die Ebene der Entwicklung eingebetteter Softwareanwendungen heruntergebrochen. Softwareingenieure entwickeln Prozesse für FMEA (Failure Mode and Effects Analysis) bzw. FMEDA (Failure Modes Effects and Diagnostic Analysis) und HARA (Hazard Analysis and Risk Assessment) sowie andere Sicherheitsanforderungen und -analysen. Diese Vielzahl an Prozessen wird dokumentiert und nachverfolgt, um ISO 26262-Richtlinien zu erfüllen.

Die wichtigsten Herausforderungen

Die Komplexität der Entwicklung von eingebetteten Anwendungen macht eine manuelle Verwaltung des Prozesses zu einem extrem schwierigen und zeitaufwendigen Unterfangen (Abbildung 4). Gleichzeitig stehen die Ingenieure, die für solche Anwendungen zuständig sind, unter massivem Druck, hochwertige Software zu liefern, die alle Fahrzeugvarianten unterstützt – und das mit sehr eng getakteten Zeitplänen. Während die Ingenieure daran arbeiten, diese Erwartungen zu erfüllen, können Änderungen auf Systemebene in Bezug auf Schnittstellen, Funktionsverhalten, Hardwarekonsolidierungen und -einschränkungen sowie Probleme bei der Netzwerkbandbreite die Softwareentwicklungsaktivitäten schnell aus dem Tritt bringen. Die Verfolgung solcher Änderungen auf Systemebene ist eine gewaltige Herausforderung, die darüber mitentscheiden kann, ob Termine eingehalten werden können oder nicht. Schlimmer noch: Je nachdem, wann die Änderungen angeordnet werden, können teure Nachbesserungen notwendig werden, die wiederum weitere Verzögerungen in der Entwicklung nach sich ziehen oder sogar die gesamte Ausführung gefährden können.

Die Implementierung von Änderungen und Aktualisierungen auf Systemebene ist eine Aktivität, die mehrere Fachgebiete und mehrere Organisationen betrifft. Eine neue Spezifikation des für das Chassis zuständigen Teams, die vielleicht durch ein Problem im Kabelbaum oder durch veränderte Maßgaben zum Leergewicht verursacht wurde, kann neue Anforderungen für die Funktionen des Antriebsstrangs bedeuten, die sich wiederum auf eine Anwendung der für das Fahrwerk zuständigen Gruppe auswirken. Während die verschiedenen Teams den Entwicklungslebenszyklus durchlaufen, ist es nahezu unmöglich, den Ursprung und sämtliche Auswirkungen von Änderungen zu verfolgen und nachzuvollziehen. Wer auch immer eine Änderung an einer Softwareanwendung vornimmt, hat also nicht unbedingt vollständigen Einblick dahingehend, welche Folgen diese Änderung haben wird.

Steuergeräte, Sensoren und andere Hardwarekomponenten, die zur Implementierung von Fahrzeugfeatures erforderlich sind, können konsolidiert oder modifiziert werden, um andere Implementierungen zu unterstützen. Wenn Varianten entwickelt oder Hardwarekomponenten konsolidiert werden, muss die eingebettete Software ihre Funktionsweise für all diese Varianten optimieren. Zu diesem Zweck müssen die Ingenieure Spezifikationen und Anforderungen von jedem Hardware- oder Betriebssystem erfassen, um die Anwendung für eine optimale Funktionalität für alle unterstützten Hardwarevarianten und Betriebssysteme zu konfigurieren.

Klarheit und Qualität von Konstruktionsanforderungen verschlechtern sich häufig auf allen Ebenen, wenn Mitarbeiter in den Ruhestand treten oder zu einem anderen Unternehmen wechseln und daher zur Klärung fehlender oder uneindeutiger Informationen nicht mehr erreichbar sind. Der Verlust dieses internen Wissens kann die Entwicklung spürbar behindern. Auch die Integrität der Software und anderer Modelle kann beeinträchtigt sein, wenn einige Beteiligte mit falschen oder veralteten Modellen arbeiten. Manche Teams arbeiten möglicherweise vollständig unabhängig von den Modellen, sodass sich der Ursprung einer bestimmten Komponente im Fahrzeug unter Umständen überhaupt nicht mehr nachvollziehen lässt. Scheinbar kleine Inkonsistenzen können in nachgelagerten Prozessen zu erheblichen Verzögerungen und teuren Nacharbeiten führen.

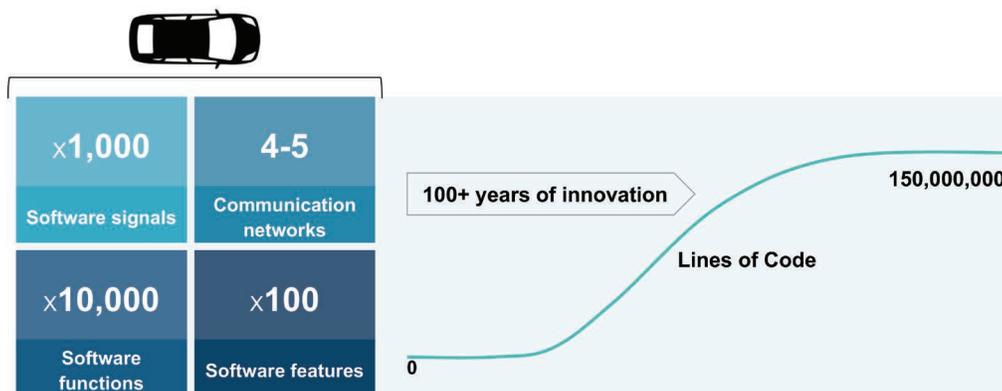


Abbildung 4: Durch die steigende Komplexität der Software wird die Verwaltung der Anwendungsentwicklung ausgesprochen schwierig.

Definition und Planung von eingebetteten Anwendungen mit einer einheitlichen Plattform

Projekte zur Definition und Planung von eingebetteten Anwendungen müssen Features auf Systemebene oder Produktinformation erfassen und nachverfolgen und Verknüpfungen zwischen Anwendung und Systemebene herstellen. Diese Verknüpfungen ermöglichen eine aktive und organische Ausrichtung an Änderungen auf Systemebene und stellen sicher, dass das Projekt während der Entwicklung im Kontext des Gesamtsystems bleibt. Die Definition oder Änderung von Softwareanforderungen führt zu Änderungen an Softwarearchitektur und -modell, die an den neuen Anforderungen ausgerichtet sind, sowie zu Aktualisierungen in Übereinstimmung mit Steuerungsalgorithmen. Diese Modellierungsaktivitäten lassen sich jetzt auf Anwendungs- und Systemebene im Hinblick auf das gewünschte Ergebnis überprüfen und validieren, bevor Änderungen auf Codeebene ausgelöst werden.

Produktintegration

Eine unverstellte Sicht auf alle Aspekte von der Implementierung bis zum Kontext auf Systemebene ist eine Notwendigkeit im komplexen Fachgebiet der eingebetteten Software für die Automobilherstellung (Abbildung 5). Softwareingenieure müssen wissen, inwiefern ihre Elemente der Anwendungsdefinition mit den Featuredefinitionen auf Systemebene und Randbedingungen für die Hardware in Zusammenhang stehen.

Die Pflege des Featurekontexts auf Systemebene ist wichtig, weil damit sichergestellt wird, dass alle Projektergebnisse mit den Anforderungen, Randbedingungen und Abhängigkeiten auf Systemebene konsistent sind. Dies gilt insbesondere dann, wenn Änderungen auf Systemebene vorgenommen werden. Möglicherweise wird eine neue Anforderung definiert, die Features von Bremsassistent und Abstandsregeltempomat kombiniert und in der Tempomatsteuerung oder im Hardwaremodul für das Chassis implementiert werden muss. Durch den Kontext auf Systemebene können Softwareingenieure verstehen, wie sich diese Änderung auf die Anwendung auswirkt, die sie entwickeln, und welche Änderungen sie vornehmen müssen. Es ist unabdingbar, dass Ingenieure diese Beziehungen bei ihrer eigenen Arbeit sehen können, damit sie schnell eine hochwertige Software liefern können, die mehrere Fahrzeugvarianten unterstützt.

Eine einheitliche, integrierte und erweiterbare Plattform für die Entwicklung eingebetteter Anwendungen ist von entscheidender Bedeutung, um die zahlreichen und komplexen Beziehungen zwischen Aktivitäten der Softwareentwicklung und der Definition auf Systemebene fachgebiets- und organisationsübergreifend verwalten zu können. Auf einer solchen Plattform lassen sich Anforderungen mit Arbeitselementen koordinieren, und sie hilft beim Kaskadieren und Implementieren von Änderungen in der gesamten Softwarearchitektur.

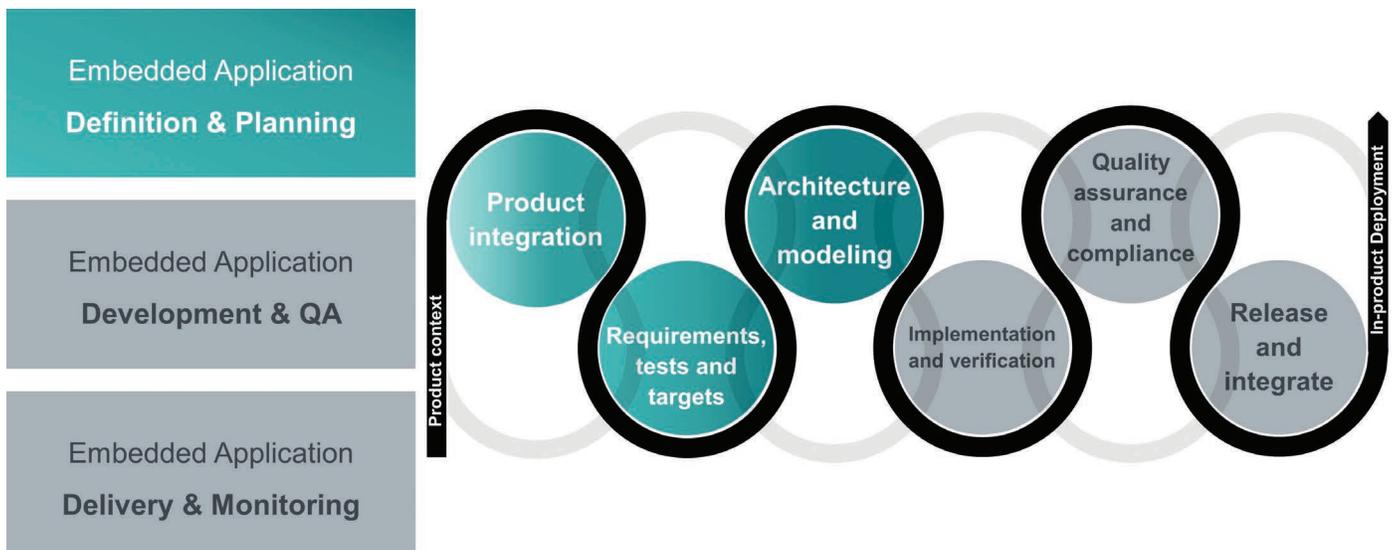


Abbildung 5: Eine unverstellte Sicht auf Featuredefinition und Hardware-Randbedingungen auf Fahrzeugebene ist während des gesamten Entwicklungsprozesses für eingebettete Anwendungen erforderlich.

Erweiterte Werkzeuge zur Koordination der Softwareentwicklung – wie z. B. Polarion – können Featuredefinitionen auf Systemebene nutzen und nachverfolgen, um eine direkte Verknüpfung zwischen Änderungen auf Systemebene und Anwendungsentwicklung zu erstellen. So lässt sich sicherstellen, dass Anwendung und allgemeine Systementwicklung jederzeit synchron laufen (Abbildung 6). Möglicherweise fordert die Systemdefinition, dass eine Anwendung für das Steuermodul der Servolenkung sämtliche Features und Funktionen der Lenkung verwaltet, wie z. B. Driftkompensation oder Spurhalteassistent. Das Koordinationswerkzeug kann die relevanten Anforderungen, Spezifikationen, Modelle auf Fahrzeugebene, Definitionen von Konfigurations- und Kalibrierungsparametern und vieles mehr von der Systemebene abrufen.

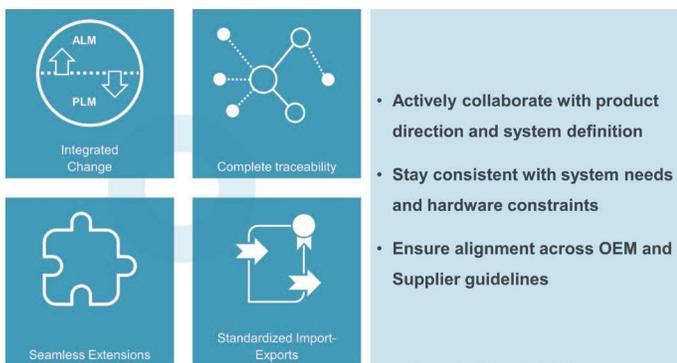


Abbildung 6: Fortschrittliche Verwaltungslösungen für die Anwendungskonstruktion können Systemanforderungen nutzen, um eine direkte Verbindung zwischen System- und Anwendungsebene herzustellen.

Dies lässt sich als Phase der Produktintegration bezeichnen, da hier die Produkt- und Softwareentwicklungszyklen integriert werden. Die Produktintegration ermöglicht eine aktive Zusammenarbeit zwischen Softwaredefinition und Planung sowie die Definition der Produktausrichtung und Systeme insgesamt. Diese Zusammenarbeit stellt sicher, dass eingebettete Softwareanwendungen mit Randbedingungen der Hardware und den allgemeinen Systemanforderungen konsistent sind. Sie hilft Herstellern und Zulieferer auch dabei, ihre jeweiligen Richtlinien und Ziele aufeinander auszurichten.

Anforderungen, Tests und Ziele

Die Definitionen und Hardware-Randbedingungen auf Systemebene werden in Anforderungen auf Systemebene festgeschrieben. Anhand dieser Anforderungen können Softwareingenieure die Anforderungen, Störfaktoren, FMEA-Prozesse, Testfälle und Funktionsziele auf die Ebene der eingebetteten Anwendung herunterbrechen. Diese Anforderungen gelten spezifisch für die jeweilige eingebettete Softwareanwendung, sind jedoch im Kontext der Systemebene konsistent.

Zur Unterstützung bei der Einhaltung gesetzlicher und prozessbezogener Vorschriften können Softwareingenieure das Anwendungsrisiko mithilfe von ISO 26262-Richtlinien und Automotive SPICE verwalten. Sie können auch die Polarion-Plattform nutzen, um Sicherheitsziele, Konzepte für die funktionale Sicherheit und damit zusammenhängende Sicherheitsanforderungen zu verwalten – mit vollständiger Verfolgbarkeit bis in den Systemkontext einerseits und bis in einzelne Codezeilen und Modelle andererseits. So können die Ingenieurteams Abweichungen, Nichtkonformität und Rückrufe reduzieren.

Mit einer einheitlichen Plattform für die Anwendungsentwicklung können Benutzer nahtlos mit allen wesentlichen Beteiligten zusammenarbeiten und konsistente Ergebnisse erzielen – unabhängig von dem Werkzeug, das für die einzelnen Ergebnisse verwendet wurde (Abbildung 7). Durch eine solche Zusammenarbeit erfolgt die Entwicklung von Anwendungsergebnissen – wie etwa Modellen oder Testplänen – trotz kontinuierlicher Änderungen an Softwareanforderungen oder Hardwareimplementierungen konsistent. Mit den Möglichkeiten der Zusammenarbeit, die diese Plattform für die Anwendungsentwicklung bietet, lassen sich auch Spezifikationen erfassen, damit die Teams jede Anwendung für die unterschiedlichsten Hardwarevarianten und Betriebssysteme von Steuergeräten optimieren können.

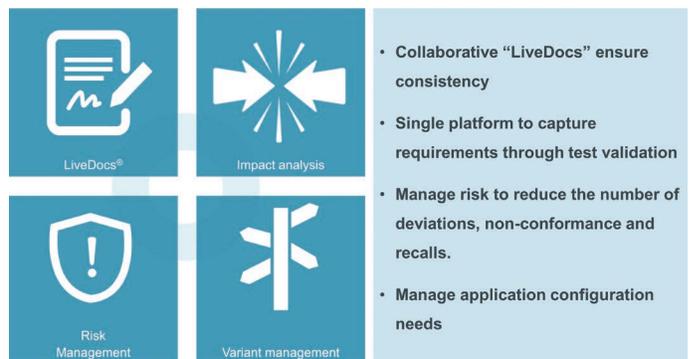


Abbildung 7: Eine einheitliche Plattform für die Anwendungsentwicklung ermöglicht Zusammenarbeit und Konsistenz zwischen Teams, die Anwendungsmodelle, Testpläne und vieles mehr entwickeln.

Architektur und Modellierung

In der Phase „Architektur und Modellierung“ beziehen Softwarearchitekten sich auf die anwendungsspezifischen Anforderungen, Spezifikationen und Verhaltensweisen, um die Architektur zu optimieren, indem sie Softwaremodelle erstellen und Interaktionen zwischen Softwarekomponenten erfassen (Abbildung 8).

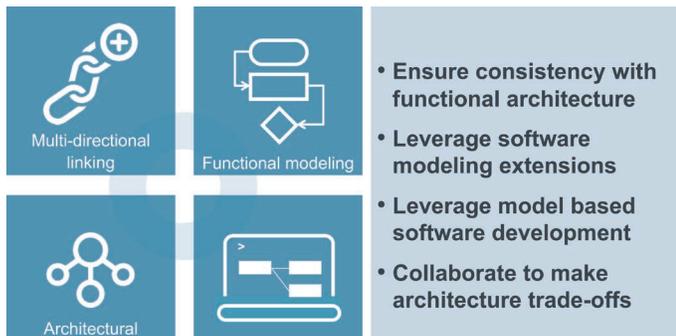


Abbildung 8: Eine einheitliche Plattform für die Anwendungskonstruktion hilft Ingenieuren beim Koordinieren von Architektur- und Modellierungsaufgaben, Durchführen von Trade-Off-Analysen und Nutzen der modellbasierten Softwareentwicklung.

Ein modellbasierter Ansatz der Softwareentwicklung unterstützt Softwareingenieure am besten bei ihrer Arbeit, die sie meist unter hohem Druck ausführen müssen. Basierend auf zuverlässigen Modellen können die Ingenieure eine umfassende Architektur entwickeln, die bereits das potenzielle Verhalten der finalen Software enthält. Ein modellbasierter Ansatz ermöglicht auch das Bearbeiten, Umgestalten, Anreichern und Analysieren der Architektur in dem Tempo, das angesichts des Drucks an den Tag gelegt werden muss.

Analyse, Tests und Verifizierung der Architektur können zu einem früheren Zeitpunkt im Entwicklungszyklus erfolgen. Nach den Tests können die Ergebnisse direkt in die Softwaremodelle eingearbeitet werden. Durch Interoperabilität mit anderen Simulationslösungen – z. B. zur Mechatroniksimulation – über FMIs (Functional Mock-up Interfaces) oder durch Verwendung von Simulink entsteht ein geschlossener Kreislauf mit Tests und Verifizierung in den Softwaremodellen.

Polarion orchestriert die Erstellung und Modifikation dieser Ergebnisse, indem es sich direkt mit Tools des Branchenstandards verbinden lässt, wie Embedded Software Designer (ESD), MATLAB/Simulink, Enterprise Architect und anderen. Während die Modelle und Komponenteninteraktionen entwickelt werden und reifen, kann Polarion eine Verknüpfung zurück in die Anwendungsdefinition erstellen. So lässt sich sicherstellen, dass alle erforderlichen Anforderungen, Spezifikationen und Verhaltensweisen über eine Darstellung im Modell verfügen und eine robuste Architektur entsteht, die alle notwendigen Interaktionen zwischen Softwarekomponenten miteinander verknüpft. Dieser Schritt validiert das Softwaremodell, noch bevor Code implementiert wird.

Die Phase „Architektur und Modellierung“ erleichtert den Entwurf und die Validierung von Softwareanwendungen in Fahrzeugen, indem sie eine gemeinsame Umgebung für System- und Softwareingenieure bietet, in der Anforderungen in mehrere Richtungen verfolgt werden können. Das Resultat ist ein digitaler Zwilling der Softwarearchitektur und einzelner Anwendungen, der wichtige Informationen für nachfolgende Entwurfsentscheidungen bietet und diese reflektiert. Diese Phase dient auch dazu, Konsistenz und Zusammenarbeit über Funktionsarchitektur, Anwendungsverhaltensmodelle und Anforderungen hinweg sicherzustellen. Verschiedene Teams können über Trade-Offs in der Architektur entscheiden und Softwaremodelle nutzen, um Probleme zu identifizieren, Variabilitäten im Verhalten zu beobachten und die Semantik zu untersuchen.

Durch Vereinheitlichen der Definition und Planung von Anwendungen auf einer einzelnen Plattform können Softwareteams Aufgaben für die Modellierung, Programmierung, Testausführung, Builderstellung und vieles mehr in den erforderlichen Werkzeugen zuweisen. Softwareingenieure haben jederzeit Einblick in die Definitionen und Randbedingungen auf Systemebene und können mit anderen Systembenutzern zusammenarbeiten.

Darüber hinaus führen die Erstellung und spätere Änderungen von detaillierten Softwareanforderungen zu Änderungen an Architektur und Modellierung der Software. Mit den weiterentwickelten Anforderungen können Softwareingenieure diese Modelle aktualisieren, sodass sie in Einklang mit Steuerungsalgorithmen stehen. Softwareingenieure können auch MiL-Tests (Model-in-the-Loop) durchführen, um zu überprüfen und zu verifizieren, dass auf Anwendungs- und Systemebene die gewünschten Ergebnisse erzielt werden, bevor Änderungen auf Codeebene ausgelöst werden.

Fazit

In modernen Fahrzeugen steigt der Anteil an softwarebasierten Funktionen, um die Anforderungen der im Wandel begriffenen Automobilindustrie zu erfüllen, mit alarmierender Geschwindigkeit. Verbraucher erwarten in ihren Fahrzeugen immer mehr Funktionen – von modernsten Sicherheitsfeatures über Verbindungsfähigkeit bis hin zu anpassbaren Inhalten. Bei anderen, eher grundlegenden Fahrzeugsystemen ist Software schon seit Jahren nicht mehr wegzudenken, z. B. Servolenkung, Motormanagement und Bremsysteme, um nur einige zu nennen. Diese einzelnen Funktionen sind inzwischen gut bekannt, aber die immer häufigeren Interaktionen zwischen diesen und anderen Fahrzeugsystemen führen zu neuen Herausforderungen.

Die wachsende Komplexität durch immer mehr Softwareinhalte, immer ausgefeiltere Funktionen und die Wechselbeziehungen zwischen virtuellen und physischen Features erfordert neue Ebenen der Orchestrierung für den Produkt- und Anwendungslebenszyklus. Dafür sind leistungsstarke Lösungen mit einem einheitlichen digitalen roten Faden zwischen Produkt und Software notwendig, um alle Aspekte nachzuverfolgen – von Systemdefinitionen auf oberster Ebene bis hin zu einzelnen Implementierungen (Abbildung 9).

Mit einer solchen Lösung kennen die Softwareingenieure beim Entwickeln von Anwendungen genau den jeweiligen Systemkontext für jedes Ergebnis. Teams können zusammenarbeiten, um mit mehr Agilität über Trade-Offs in der Architektur zu entscheiden und Änderungen zu verwalten. Eine solche Lösung ermöglicht auch einen geschlossenen Kreislauf von der Verifizierung und Validierung

zurück zur Definition, Planung und Entwicklung von Anwendungen, um die Software kontinuierlich zu optimieren und die Qualität zu verbessern.

Die Phase der Definition und Planung von Anwendungen ist jedoch nur der erste Schritt bei der Entwicklung von modernen Softwareanwendungen, die für die heutigen, immer stärker computergesteuerten Fahrzeuge erforderlich sind. Sobald die Planungs- und Definitionsphase abgeschlossen ist, müssen Softwareingenieure die Modelle in den eigentlichen Softwarecode umsetzen und testen, um eine hohe Qualität sicherzustellen. Die Anwendungen müssen auch so konfiguriert werden, dass sie für alle Hardware- und Betriebssystemvarianten in den Fahrzeugen funktionieren, in denen sie implementiert werden. Auch diese nachfolgenden Unterprozesse können von einer einheitlichen Plattform für die Entwicklung eingebetteter Anwendungen profitieren.

Letztendlich sind diejenigen Unternehmen, welche die Entwicklung von Softwareanwendungen über Organisationen, Fachgebiete und funktionale Abstraktionen hinweg effektiv verwalten können, in der besten Position, um in der zunehmend digitalisierten Automobilindustrie zu bestehen und zu wachsen. Dies gilt insbesondere angesichts der Tatsache, dass Softwarefunktionalität immer weniger von der Hardware der Steuergeräte abhängig ist, weil in Fahrzeugen immer mehr Betriebssysteme und Firmware zum Einsatz kommen. Eine zuverlässige Methodik zum Nachverfolgen jeder Anwendung sowie ihrer funktionalen Inhalte, Systemrandbedingungen und potenziellen Hardwarerandbedingungen ist der Schlüssel zum Erfolg. Eine einheitliche Plattform für die Entwicklung von

eingebetteten Anwendungen in der Automobilbranche ist die Grundlage dieser Fähigkeit.

Um zu erfahren, wie Lösungen von Siemens Sie unterstützen können, besuchen Sie uns unter [siemens.com/aes](https://www.siemens.com/aes). Hier finden Sie weitere Inhalte wie Blogs, White Papers, Podcasts, Produktvideos, Webinare, Lösungsfunktionen und Infografiken.

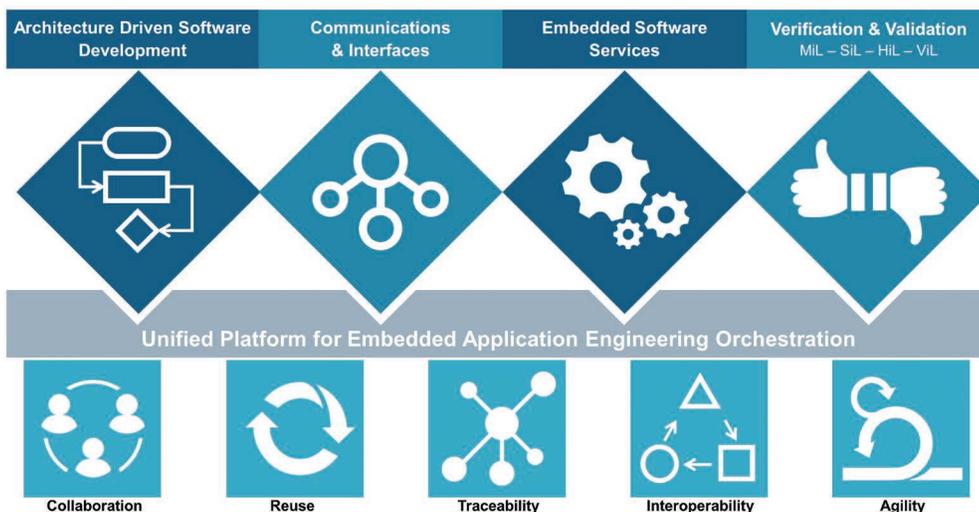


Abbildung 9: Eine einheitliche Plattform für die Orchestrierung der Konstruktion eingebetteter Anwendungen sorgt für Verfolgbarkeit und ermöglicht die Zusammenarbeit in allen Prozessen der Anwendungsentwicklung.

Siemens Digital Industries Software

Hauptsitz

Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

Nord-, Mittel- und Südamerika

Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

Europa

Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

Asien-Pazifik

Unit 901-902, 9/F
Tower B, Manulife Financial Centre
223-231 Wai Yip Street, Kwun Tong
Kowloon, Hongkong
+852 2230 3333

Über Siemens Digital Industries Software

Siemens Digital Industries Software fördert die Transformation von Unternehmen auf ihrem Weg in Richtung „Digital Enterprise“, in dem Engineering, Fertigung und Elektronikdesign bereits heute den Anforderungen der Zukunft entsprechen. Unsere Lösungen unterstützen Unternehmen jeder Größe bei der Entwicklung digitaler Zwillinge, die ihnen neue Einblicke, Möglichkeiten und Automatisierungsgrade bieten, um Innovationen voranzutreiben. Weitere Informationen über die Produkte und Leistungen von Siemens Digital Industries Software finden Sie unter [siemens.com/software](https://www.siemens.com/software), oder folgen Sie uns über [LinkedIn](#), [Twitter](#), [Facebook](#) und [Instagram](#).
Siemens Digital Industries Software – Where today meets tomorrow.

[siemens.com/software](https://www.siemens.com/software)

© 2019 Siemens. Eine Liste wichtiger Marken von Siemens findet sich [hier](#). Alle anderen Marken sind Eigentum der jeweiligen Inhaber.

81120-82103-C4-DE 6/20 LOC