



# SIEMENS

*Ingenuity for life*



Siemens Digital Industries Software

## Koordinieren der Entwicklung von Embedded-Anwendungen für Automotive

### Anwendungsentwicklung und Qualitätssicherung

#### Kurzdarstellung

Die derzeit interessantesten und innovativsten Automotive-Funktionen basieren auf hoch entwickelten, äußerst komplexen Embedded-Softwareanwendungen. Hunderte Millionen von Codezeilen steuern in modernen Fahrzeugen Dutzende von Systemen. Ingenieure im Bereich Automotive Software sind gefordert, diese enorme Komplexität zu meistern und dem Endkunden als einfache, intuitive Benutzererfahrung zu präsentieren. Ingenieure, die Anwendungscode entwickeln und Qualitätssicherungstests durchführen, brauchen eine einheitliche Plattform, die ihnen einen umfassenden Einblick in die Designanforderungen, die wesentlichen Systemattribute und -einschränkungen und sogar die einzelnen Codezeilen bietet.

Piyush Karkare  
Siemens Digital Industries Software

# Einführung

Angesichts der vorherrschenden Trends in der Automobilindustrie gewinnt die Entwicklung von Embedded Software für die gesamte Fahrzeugentwicklung an Bedeutung. Fahrzeugautomatisierung, Konnektivität, Elektrifizierung und Shared Mobility (ACES) erhöhen den Bedarf an Software auf höchstem Niveau, die als Grundlage für Funktionen wie Fahrerassistenzsysteme (Advanced Driver Assistant System, ADAS), Batteriemangement, Verkehrsnetzwerk (Vehicle-to-everything-Kommunikation, V2X) usw. dient (Abbildung 1).

Um mit diesen Trends Schritt halten zu können, verfolgen Automobilhersteller und Führungskräfte einen Weg der vertikalen Integration mit dem Ziel, die Softwareentwicklung und weitere Schlüsseltechnologien in das eigene Unternehmen einzugliedern. Die Lieferanten stehen vor der Aufgabe, auf Fahrzeugebene überzeugende Funktionen bereitzustellen, die eine umfangreiche Softwareentwicklung erfordern, um im harten Wettbewerb bestehen zu können. OEMs und Lieferanten, die ihre Kernkompetenzen um Software erweitern möchten, sehen sich

schwierigen Herausforderungen gegenüber. Für moderne Fahrzeuge entwickelte Software muss den höchsten Entwicklungsstand aufweisen, vor allem was automatische Sicherheitsfunktionen und Fahrerassistenzfunktionen anbelangt. Zudem wird von der Software äußerste Zuverlässigkeit und Funktionalität erwartet.

Ausgeklügelte Automotive-Funktionen erfordern hochkomplexe Software und führen zu einer massiven Zunahme der Komplexität der fahrzeuginternen Software insgesamt. Heutzutage umfasst die Fahrzeugsoftware in der Regel mehr als 150 Millionen Codezeilen. Die Komplexität der Softwareanwendungen an sich wird noch durch die Interaktion zwischen den Fahrzeugsystemen verstärkt, die auf Basis dieser Softwareanwendungen kommunizieren und zusammenwirken. Ingenieure im Bereich Automotive Software sind gefordert, diese enorme Komplexität zu meistern und dem Endkunden als einfache, intuitive Benutzererfahrung zu präsentieren.

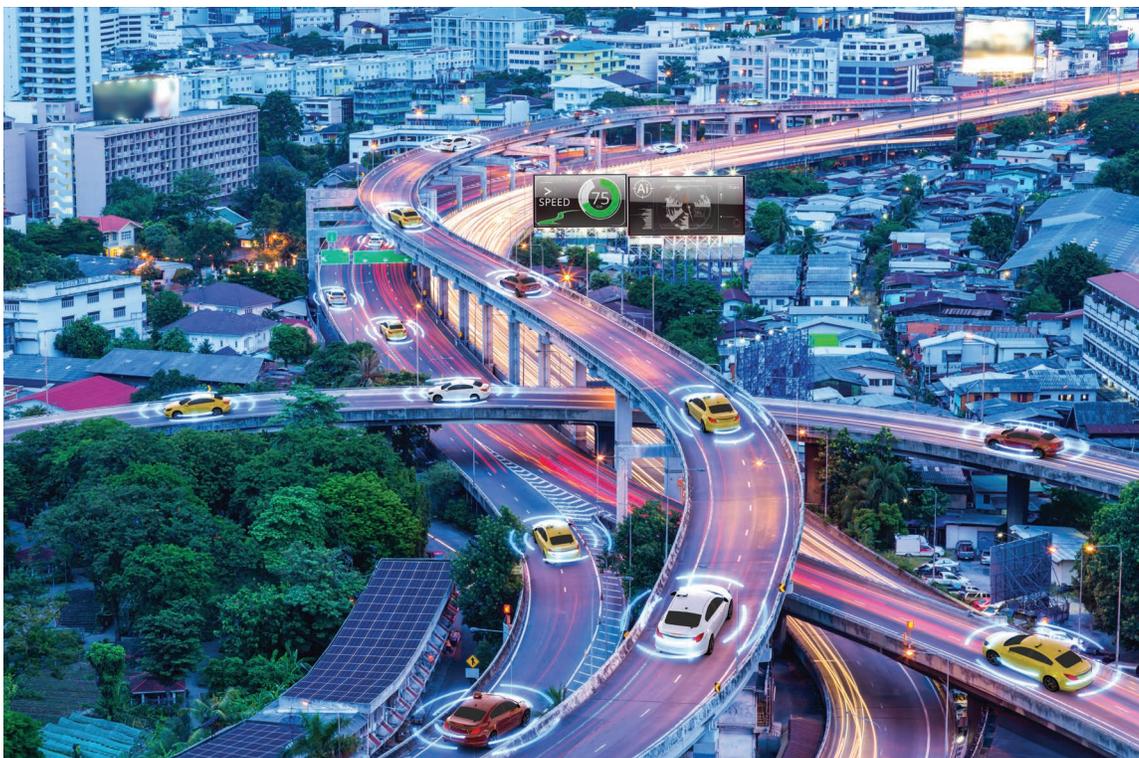


Abbildung 1: Trends wie autonomes Fahren, Konnektivität, Elektrifizierung und Shared Mobility lassen den Bedarf an Automotive Software steigen.

# Die Entwicklung von Embedded-Anwendungen wird zunehmend anspruchsvoller

Softwareingenieure sind eng in den Entwicklungsprozess eingebunden. Sie prüfen Änderungen und Systemaktualisierungen auf Basis der Softwarearchitekturen und stellen mit Softwaremodellen die Funktionalität unter Beweis, die von der Definition auf Systemebene gefordert wird. Viele OEMs und Lieferanten wenden bei der Erstellung ihrer Anwendungen einen modellbasierten Softwareentwicklungsansatz an. Mithilfe der Architektur- und Modellierungsprozesse wird die Funktionalität der Software geprüft und validiert, bevor Code erstellt oder geändert wird. Die größte Herausforderung besteht darin, die Aktivitäten aller Ingenieure entlang des Datenflusses passiv zu koordinieren, ohne dass die Konsistenz der Daten manuell sichergestellt werden muss.

Ingenieure im Bereich Embedded Software für Automotive müssen außerdem besonders strenge Sicherheitsbestimmungen einhalten, was den fertigen Software-Build und den Entwicklungsprozess angeht. Es ist wichtig, dass die Ingenieure die Möglichkeit haben, die im Zuge der Entwicklung getroffenen Designentscheidungen, erstellten Testpläne, angefallenen Testergebnisse und erstellten Daten zu prüfen. Dieser digitale rote Faden dient als Beleg dafür, dass die Software die erforderlichen Bestimmungen erfüllt. Angesichts der Komplexität und der Geschwindigkeit der Softwareentwicklung ist es leicht möglich, den Überblick über diese Informationen zu verlieren, wenn Termine eingehalten oder Probleme gelöst werden müssen.

Die Entwicklung von Embedded Software für Automotive wird immer anspruchsvoller, da Fahrzeugfunktionen zunehmend auf komplexen Softwareanwendungen basieren. Embedded-Software-Teams benötigen

eine einheitliche Entwicklungsplattform, die sowohl die Verwaltung des Lebenszyklus der Softwareentwicklung von der Planung über die Entwicklung bis hin zur Bereitstellung als auch die Interaktion mit dem Product Lifecycle Management (PLM) und anderen erforderlichen Engineering-Lösungen ermöglicht (Abbildung 2). Während der Anwendungsentwicklung und der Qualitätssicherung (Quality Assurance, QA) brauchen die Teams einen umfassenden Einblick in die Designanforderungen, die wesentlichen Systemattribute und -einschränkungen und sogar die einzelnen Codezeilen. Eine solche Plattform gewährleistet aufgrund der umfassenden Verfolgbarkeit die Einhaltung der gesetzlichen Bestimmungen und unterstützt die funktionale Wiederverwendung und die Zusammenarbeit über die beteiligten Engineering-Domänen hinweg.

## Entscheidende Herausforderungen

Eine große Herausforderung bei der Anwendungsentwicklung ist die Inkonsistenz zwischen Software- und Produktentwicklungsprozessen. OEMs verwenden meist Fahrzeugmeilensteine, um Fahrzeugfunktionen, Änderungen und Aktualisierungen auf Systemebene zu verfolgen. Inzwischen erfolgt die Softwareentwicklung in Form schneller AGILE- und Hybrid AGILE-Prozesse. Der Widerspruch zwischen diesen Entwicklungsmethodiken kann zu Problemen bei Prüfpunkten führen, die den Fortschritt der Entwicklung behindern können. Softwareteams müssen möglicherweise auf Aktualisierungen auf Systemebene warten, die vielleicht erst beim nächsten Meilenstein verfügbar sind. Ebenso machen Softwareteams, die bis zum nächsten Meilenstein einen Build

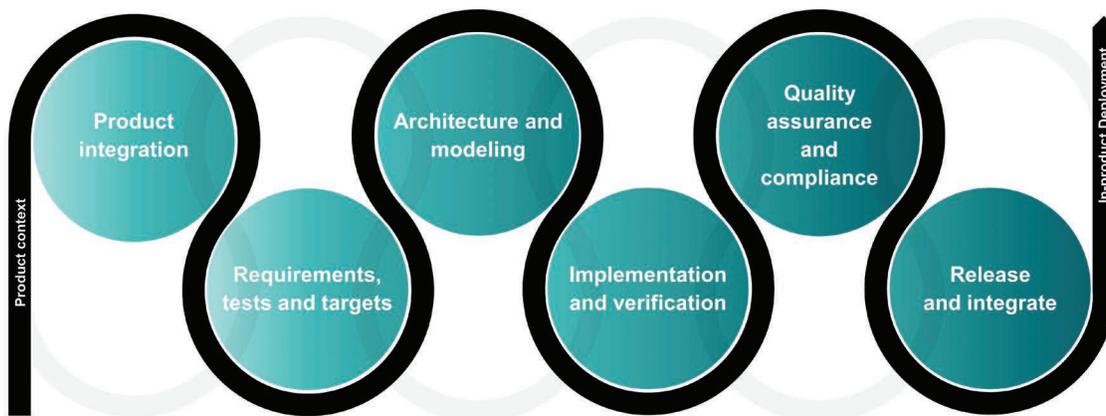


Abbildung 2: Ingenieure im Bereich Embedded Software für Automotive brauchen eine einheitliche Plattform, um die Anwendungsentwicklung verfolgen und verwalten zu können.

erstellen müssen, eventuell Abstriche bei der Qualität, um den Termin einhalten zu können.

Widersprüchliche Entwicklungsmethodiken erschweren zudem die Verfolgbarkeit und Sichtbarkeit der Informationen in den verschiedenen Teams. Bei den Tests, insbesondere bei Hardware-in-the-loop-(HiL-) oder Driver-in-the-loop-(DiL-) Tests, ist entscheidend, dass die Softwareteams den Gegenstand des Tests, vor allem aber den Grund des Tests kennen. Beispielsweise muss bekannt sein, welche Änderungen an welchen Datenartefakten vorgenommen wurden, welche Software-Builds einsatzbereit sind und welche Hardware-Abstraktionsebenen für die geprüften und validierten Software-Builds erforderlich sind. Die für die Implementierung verantwortlichen Teams wiederum brauchen Einblick in die Testergebnisse, damit sie zeitnah und effizient Aktualisierungen vornehmen und beim Test ermittelte Probleme lösen können.

Während die automatisierte Codeerstellung Teams durch die Integration von Softwaremodellierung und -codierung flexibler machen kann, führt eine schwerfällige und segmentierte Datenmanagementlösung bei der Integration und Koordination zu einem erhöhten Risiko und macht erzielte Zeiteinsparungen zunichte. Ingenieure sind darauf angewiesen, dass die Codeimplementierung mit den aktuellen Einschränkungen auf Hardware- und Systemebene konsistent ist. Dies setzt einen prüfbareren Änderungsverwaltungsprozess voraus, der den Ingenieuren die Möglichkeit gibt, Anwendungsbuils basierend auf den Testergebnissen in Echtzeit zu aktualisieren.

Ein prüfbarer Änderungsverwaltungsprozess, der Produkt- und Software-Engineering abdeckt, ist für die Einhaltung der gesetzlichen Bestimmungen von entscheidender Bedeutung. Erstens muss gewährleistet sein, dass die Softwareteams schnell auf neue oder missverständliche Bestimmungen reagieren können, die Auswirkungen auf die Funktionalität der Software haben. Zweitens muss die Änderungsverwaltung sicherstellen, dass der OEM über die erforderlichen Belege verfügt, um die Einhaltung der gesetzlichen Bestimmungen nachweisen zu können.

Für die Flexibilität des Implementierungs- und Prüfprozesses sind nicht nur die Ingenieure, die den Code erstellen, verantwortlich. Softwaretests und ihre Abdeckung müssen auf Anforderungen basieren und in einer früheren Phase des Entwicklungszyklus angestoßen werden. Es ist von Vorteil, wenn Fehler frühzeitig in einer Vielzahl von Implementierungen gefunden werden, da Fehler umso höhere Kosten verursachen, je später sie gefunden werden. Anwendungsdesigns sollten frühzeitig modelliert, simuliert und ausgeführt werden, damit das Konzept basierend auf Designanforderungen, Architektureinschränkungen und Systemverhalten optimiert und validiert werden kann. Wenn frühzeitig im Systemkontext Co-Simulationen von Hardware und Software durchgeführt werden, können die Softwareingenieure Probleme und Fehler zu einem früheren Zeitpunkt im Entwicklungsprozess finden.

Es kann sehr zeitraubend sein, nach Abschluss einer Testrunde die empfohlenen Änderungen zu implementieren. Dies gilt vor allem dann, wenn das Implementierungsteam anhand komplizierter Testergebnisse feststellen muss, welche Korrekturen durchzuführen sind. Es liegt im Interesse einer schnellen und genauen Lösung, wenn die relevanten Teams und Stakeholder eine genaue Dokumentation der Anforderungen und Systemeinschränkungen und die Möglichkeit zur Verfolgung erhalten.

Einer umfassenden Verfolgbarkeit von Änderungen auf Systemebene bis hinunter zu einzelnen Codezeilen stehen häufig fortlaufende Änderungen und Aktualisierungen entgegen. Während des gesamten Entwicklungsprozesses einen klaren Durchblick zu bewahren, wird mit der Zunahme der Softwareinhalte und der Abhängigkeiten über mehrere Funktionen hinweg immer komplizierter. Das Problem wird noch verschärft, wenn Ingenieure durch mehrere, nicht integrierte Werkzeuge navigieren müssen, um die Verfolgbarkeit aufrechtzuerhalten. Lösungen, die auf Tabellen, Datenbanken und selbst erstellten Verbindungen basieren, sind kompliziert und wachsen nicht mit dem Unternehmen.

# Entwicklung und Qualitätssicherung von Embedded-Anwendungen mit einer einheitlichen Plattform

Heutzutage sind Lösungen verfügbar, mit denen die komplexen und gleichzeitig ausgeführten Aufgaben des Anwendungsentwicklungs- und Qualitätssicherungsprozesses verfolgt und verwaltet werden können. Diese Lösungen bieten die umfassende Verfolgbarkeit und Sichtbarkeit, die für die Entwicklung von Embedded-Anwendungen erforderlich ist. Im Folgenden wird die Entwicklung einer Embedded-Anwendung bei vorhandener Lösung zur Koordination der Entwicklung beschrieben (Abbildung 3).

## Funktionsdesign und Modellierung – Nutzen von Modellen und getesteten Softwarekomponenten für die Codierung

Softwareteams erstellen robuste Architekturen aus Softwarekomponenten, Softwaremodellen, die echte funktionale Zwillinge des erforderlichen Verhaltens darstellen, und Modellinteraktionen zur Gewährleistung der Vollständigkeit der Kommunikation, die erforderlich ist, um vor der Codeerstellung die Elemente einer Softwareanwendung zu definieren und zu validieren. Softwarearchitekten und -ingenieure verwenden für diese Arbeiten in der Regel spezielle Werkzeuge wie MATLAB oder SIMULINK. Moderne Lösungen zur Koordinierung der Softwareentwicklung umfassen Erweiterungen für diese Werkzeuge, die eine direkte Verbindung zwischen Architektur, Modellen, Anforderungen, Testmethoden, Verlustanalyse, Einhaltung der Sicherheitsbestimmungen usw. herstellen. Mit zunehmender Ausgereiftheit der Ergebnisse verfolgt die Softwareentwicklungslösung den Fortschritt und die Konsistenz der Modelle, der Architektur und der Komponenteninteraktionen in Bezug auf die Anwendungsanforderungen und -einschränkungen. Dadurch ist gewährleistet, dass alle Anforderungen, Spezifikationen und Verhaltensweisen als einer robusten Architektur zugeordnete

Modelldarstellung abgedeckt werden, die alle Interaktionen zwischen den Komponenten beschreibt.

Die einheitliche Plattform für die Anwendungsentwicklung hilft Softwareingenieuren, Konsistenz über die funktionale Architektur hinweg zu wahren, und ermöglicht es unterschiedlichen Teams, in Bezug auf die Architektur zu fundierten Kompromissen zu gelangen. Eine solche Lösung verfolgt auch automatisch, ob die Modelle die Anwendungsanforderungen erfüllen und ermöglicht die Verfolgbarkeit zwischen den Modellen und den relevanten Anforderungen. Ingenieure können die modellbasierte Softwareentwicklung infolgedessen effektiver nutzen, um Probleme frühzeitig zu erkennen und bei nachgelagerten Prozessen die Konsistenz wahren (Abbildung 4).

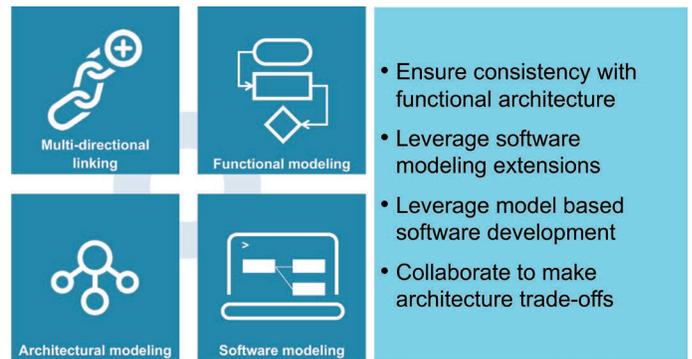


Abbildung 4: Eine einheitliche Anwendungsengineering-Plattform gewährleistet die Verfolgbarkeit von den Anforderungen zu Softwarearchitekturen, Modellen usw. und unterstützt auf diese Weise Architektur- und Modellierungsprozesse.

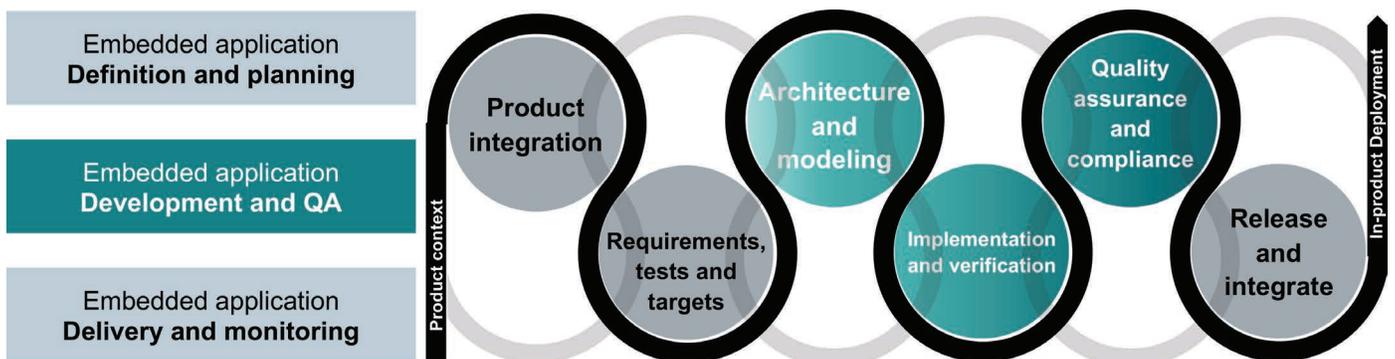


Abbildung 3: Anwendungsentwicklung und Qualitätssicherung im Gesamtablauf der Anwendungsentwicklung.

Die Softwaremodellierung gewährleistet, dass die Systemanforderungen erfüllt sind, bevor die Ingenieure viel Zeit in die eigentliche Codierung investieren. Die Anwendungsentwicklungsplattform lässt sich so konfigurieren, dass diese Codierungs-Tasks den Teammitgliedern automatisch zugewiesen werden. Dabei werden anpassbare Bedingungen zugrunde gelegt, die auf Projektbasis festgelegt werden können. Die Anwendungsentwicklungsplattform kann die Fortschritte der Ingenieure bei der Ausführung dieser Tasks überwachen. Dazu werden Erweiterungen der ausgewählten Lösung für die Software-Code-Verwaltung (Software Code Management, SCM) und der ausgewählten integrierten Entwicklungsumgebung (Integrated Development Environment) verwendet. Die Anwendungsentwicklungsplattform kann über verschiedene integrierte Werkzeuge für Codeleistung (statisch und dynamisch) und Codeabdeckung auch die Einhaltung nicht funktionaler Anforderungen verfolgen.

Eng koordinierte und konsistente Verfahren zur Softwaremodellierung beschleunigen nicht nur den Prozess, sondern können auch mit Methoden wie SOTIF (Safety of the Intended Functionality) gewährleisten, dass die Software wie vorgesehen arbeitet und dass Risiken per Design ausgeschlossen werden. Die Einbindung von SOTIF-Methoden ergänzt standardmäßige Ansätze für die funktionale Sicherheit, die Risiken durch die Anwendung von Sicherheitszielen minimieren, die vom Auftreten von Fehlern ausgehen. Diese Kombination führt zu außergewöhnlich robusten Embedded-Softwareanwendungen für Automotive.

#### Codeimplementierung und -integration

Die Anwendungsentwicklungs-Plattform gibt Ingenieuren die Möglichkeit, die Aktivitäten der Modellierung und Codeimplementierung zusammenzuführen und so zu vermeiden, dass gemeinsame Aufgaben von Modellerstellung und Codierung doppelt ausgeführt werden. Eine direkte Verbindung zwischen Anforderungen, Modellen, Code und Prüfung gewährleistet, dass Implementierungen im Laufe ihrer Weiterentwicklung geprüft werden können. Eine solche Verbindung sorgt auch dafür, dass Beschränkungen und Änderungen der Definition auf der Systemebene, der Hardwarespezifikationen usw. sofort berücksichtigt werden und die Implementierungen daher immer auf dem neuesten Stand sind. Eine Anwendungsentwicklungsplattform bietet außerdem prüfbare Prozesse für die Änderungssteuerung, die nach Modell-, Software-, Hardware- oder Vehicle-in-the-loop-Tests flexible Codeaktualisierungen ermöglichen (Abbildung 5).

Werkzeuge für die iterative Planung helfen Entwicklungsteams für Embedded-Anwendungen, ihre Zeit effizient einzusetzen und die Ausführung der Tasks zu verfolgen, um die Einhaltung des Zeitplans für die Entwicklung zu gewährleisten. Die Plattform kann so konfiguriert werden, dass die Ausführung bestimmter Aktionen oder Prozessschritte elektronische Signaturen erfordert. Dadurch wird gewährleistet, dass Entscheidungen durch wichtige Stakeholder überprüft und genehmigt werden. Die Entwicklungsplattform für Embedded-Anwendungen kann außerdem die Verantwortlichkeit für die Analyse und Implementierung von Änderungen verfolgen. Das heißt, dass die Teams bei geplanten Aktualisierungen prüfen können, ob ein Software-Release oder -Build gerechtfertigt ist. Möglich wird dies durch die uneingeschränkte Verfolgbarkeit aller geänderten Datenartefakte und der Begründung für jede Änderung.

Ebenfalls zum Umfang der Planungslösungen gehört die Verwaltung der Teamgeschwindigkeit, die es möglich macht, basierend auf empirischen Daten den Endzeitpunkt einer Aufgabe zu schätzen. Unterschiedliche Schätzmethoden wie Zeit- und Kostenschätzung sowie Story Points werden unterstützt. Die Bedingungen für einen abgeschlossenen Workflow können vom Teamleiter oder Anwendungseigentümer bestimmt werden. Zum Beispiel könnte festgelegt werden, dass ein Workflow nur dann als abgeschlossen gekennzeichnet wird, wenn die Dokumentation übermittelt wurde. Die Benutzer können auch Begleitdokumente, wie Spezifikationen der Komponenten- und Designanforderungen, beifügen, um die funktionale und qualitative Konsistenz zu gewährleisten.

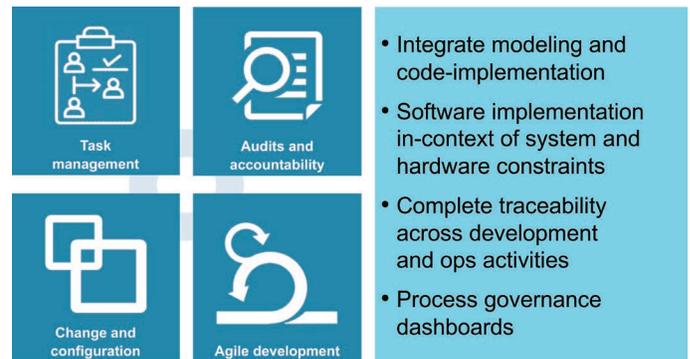


Abbildung 5: Eine einheitliche Plattform für die Koordination der Anwendungsentwicklung erweitert Task-Management, Verantwortlichkeit und Änderungsverwaltung und unterstützt agile Entwicklungsmethoden.

#### Qualitätssicherung

Die Qualitätssicherung ist ein meist kontinuierlicher Prozess, der zu einem frühen Zeitpunkt beginnt und die Entwicklung über die Umsetzung der Anforderungen, die Erstellung von Architekturen und Modellen sowie die Codeimplementierung bis zur Reife begleitet. Die Qualitätssicherung ist die Summe der Tests auf mehreren Abstraktionsebenen, zum Beispiel Model-in-the-loop (MiL), Software-in-the-loop (SiL), Hardware-in-the-loop (HiL) und Vehicle-in-the-loop (ViL), wobei sowohl virtuelle HiL-Tests als auch physische Tests auf dem Testgelände durchgeführt werden. Der iterative Testablauf lässt sich so koordinieren, dass die Qualität auf jeder Entwicklungsstufe und bei der Implementierung von Designänderungen bewertet wird. Ein solcher Testablauf kommt auch den Abläufen des modellbasierten Softwaredesigns entgegen, da die Ingenieure die Softwaremodelle nach Bedarf schon frühzeitig im Entwicklungsprozess ändern und prüfen können.

Beim Entwickeln und Testen von Softwarearchitekturen, Modellen und Code erzeugen die Ingenieure jedoch ständig neue Daten, die verfolgt und dem jeweiligen Software-Build bzw. der entsprechenden Konfiguration zugeordnet werden müssen. In jeder Testrunde müssen die Softwareingenieure die Testpläne mit Details zum Testfall, zur Strategie für die Testausführung, zu Testvektoren und zu sonstigen relevanten Datenartefakten erstellen bzw. aktualisieren. Nach Abschluss der Tests müssen die Probleme zur Bearbeitung zugewiesen und behoben werden, während die Testergebnisse den jeweiligen Testläufen

zugeordnet werden. Diese Datenzuordnungen sind für den Nachweis der Testabdeckung von entscheidender Bedeutung und unterstützen eine umfassende Verfolgbarkeit von den Anforderungen bis zur Implementierung.

Das Organisieren und Verfolgen der Testwertschöpfungskette (Testpläne, Testfälle, Testergebnisse, zugehörige Datenartefakte und der Status der ermittelten Probleme und Risiken) verleiht der Herausforderung und der Bedeutung einer umfassenden Koordination der Softwareentwicklung eine neue Dimension. Die große Menge vorhandener Daten weist komplexe Wechselbeziehungen auf, die verfolgt und verwaltet werden müssen. Werden jedoch die Ingenieure mit dem Verfolgen und Verwalten der Daten beauftragt, hindert sie dies daran, sich auf das Software-Engineering und die Softwareentwicklung zu konzentrieren. Die Softwareteams sind auch dafür verantwortlich, die Bereitstellung der Anwendungen mit allen zugehörigen Daten für die fertigen Builds zu koordinieren. Die heutigen Plattformen für Anwendungsentwicklung und Koordination weisen Funktionen auf, mit denen sich diese Komplexität meistern und die Qualität der bereitgestellten binären und zugehörigen Daten gewährleisten lässt (Abbildung 6).

Plattformen für Anwendungsentwicklung und Koordination wie Polarion verfügen über integrierte Werkzeuge für das Testmanagement. Mit diesen Werkzeugen können die Ingenieure Testfälle erstellen und mit dem entsprechenden Work Item verknüpfen. Beispiele für Work Items sind Anwendungsanforderungen, Änderungsanforderungen, andere Testfälle usw. Mithilfe der Testmanagement-Werkzeuge lassen sich die Schritte jedes Testfalls ändern und die Testspezifikation von der Testelementkonfiguration trennen, um mit einer einzelnen Prozedur eine größere Testabdeckung zu erreichen. Diese Lösungen können auch mit externen Werkzeugen zur Testautomatisierung integriert werden. Dadurch wird es möglich, Tests automatisch durchzuführen und die Ergebnisse automatisch zu importieren.

Dieser Workflow vereinfacht die Gewährleistung der Prozesskonformität. Unternehmensprozesse und Informationen zu Best Practices sind direkt in der Projektansicht verfügbar, damit sich alle Teammitglieder an die festgelegten Prozesse halten, auch wenn sie nicht alle Prozessdetails kennen. In den Polarion-Workflow sind auch Anforderungen, Tasks, Änderungsanforderungen, Prozessmanagement, Projektplanung, Zeitmanagement, Build-Management, Quellcodes, Audits und Metriken sowie die Verfolgung der Konformität mit dem Reifegradmodell integriert. Dadurch wird gewährleistet, dass die Anwendungsentwicklung im Zeitplan liegt und eine hohe Qualität aufweist.

Erweiterte Plattformen für Anwendungsentwicklung und Koordination verfügen zusätzlich über Fehler- und Risikomanagementfunktionen. Mit diesen Funktionen können automatisch Problem- und Risikoberichte zu fehlgeschlagenen Tests erstellt und den Softwareingenieuren bestimmte Tasks zugewiesen werden, um die Zeit bis zur Lösungsimplementierung zu verkürzen. Die Entwicklungsplattform für Embedded-Anwendungen verfolgt automatisch Probleme und zeichnet alle nachfolgenden, zugehörigen Aktivitäten auf. Dies

schließt auch implementierte Korrekturen und zusätzliche Tests ein. Wenn zusätzliche Tests durchgeführt werden, ist es wichtig, eine Co-Simulation der Hardware und Software durchzuführen, um die Gesamtqualität zu verbessern und die Softwarebereitstellung für Fahrzeugprogramme zu ermöglichen. Mit einer einheitlichen Plattform zur Koordinierung der Entwicklung von Embedded-Anwendungen kann gewährleistet werden, dass jeder Software-Build in allen relevanten Hardwarekonfigurationen mit emulierter und physischer Hardware getestet wird. Alle Testinformationen werden verfolgt. Dabei spielt es keine Rolle, ob der Test manuell oder automatisch, mit einem Werkzeug eines anderen Anbieters oder in der Plattfromgebung für die Anwendungsentwicklung durchgeführt wird. So ist gewährleistet, dass während der Entwicklung alle aktiven und behobenen Probleme in einem Bericht erfasst werden.

Aus diesen Funktionen resultieren mehrere wichtige Vorteile für Ingenieure im Bereich Embedded Software für Automotive. Sie können einen Prototyp der Designs erstellen und Designs simulieren und ausführen, um zu überprüfen, ob sie die Anforderungen erfüllen. Des Weiteren können sie Architekturen optimieren und Prüfungen auf richtiges Verhalten durchführen. Ingenieure haben auch die Möglichkeit, die Designs mit Co-Simulationen von Hardware und Software im Kontext der Systemhardware und der Peripheriekomponenten zu prüfen. Dabei lassen sich Probleme oder Risiken in der Software oder der Hardwarezuordnung aufdecken, die in einem Systemkontext sichtbar werden. Die Anwendungsentwicklungsplattform verfolgt im Hintergrund Anforderungen, Modelle, Architekturen, Testpläne, Testergebnisse usw., um die Verfolgbarkeit zu gewährleisten und Softwareingenieuren zum richtigen Zeitpunkt die richtigen Informationen bereitzustellen. Dadurch wird die Problemlösung optimiert, weil der jeweiligen Partei ohne größeren Aufwand eine genaue und vollständige Dokumentation zur Verfügung gestellt werden kann. Die umfassende Verfolgbarkeit trägt außerdem dazu bei, dass kritische Fehler nicht extern freigegeben werden.

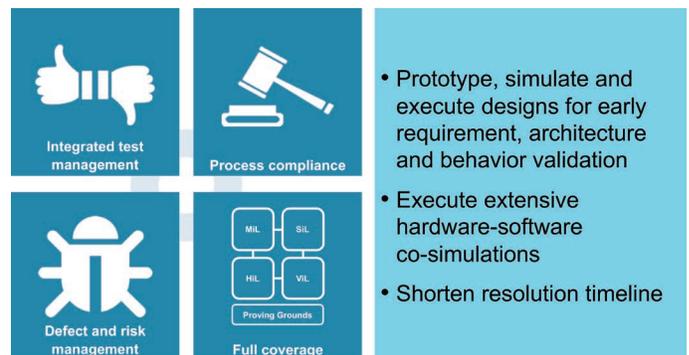


Abbildung 6: Werkzeuge für Testmanagement, Prozesskonformität und Risikomanagement helfen, eine volle Testabdeckung zu erreichen.

# Fazit

Die Entwicklung und die Qualitätssicherung von Embedded Software für Automotive ist ein komplexer Prozess, da die Implementierungshardware (einschließlich Peripheriekomponenten wie Sensoren und Aktoren) sowie Systemaspekte und -einschränkungen (einschließlich Fahrzeugintegration und geometrische Abhängigkeiten) kontinuierlich berücksichtigt werden müssen. Mit der einheitlichen Software-Engineering-Plattform steht eine Engineering-Umgebung zur Verfügung, die trotz fortlaufender Änderungen die Konsistenz der Daten gewährleistet. Alle am Entwicklungsprozess beteiligten Parteien arbeiten kontinuierlich daran, Softwareanwendungen hoher Qualität bereitzustellen, die mit sämtlichen Fahrzeugvarianten kompatibel sind. Die Plattform erreicht diese Datenkohärenz durch eine robuste Integration mit den verschiedenen, bei der Anwendungsentwicklung eingesetzten Werkzeugen und leistungsfähige Funktionen zur Änderungsverwaltung.

Diese Funktionen tragen wesentlich zur Realisierung der immer komplexeren Software-Features und -Funktionen bei, die erforderlich sind, um angesichts der Digitalisierung des Automotive-Markts

wettbewerbsfähig zu bleiben. Solange ADAS, Infotainment und eine wachsende Palette von elektrischen, elektronischen und softwarebasierten Funktionen den Mehrwert für den Kunden weiter erhöhen, wird die durchschnittliche Anzahl der Codezeilen in Fahrzeugen weiter zunehmen. Bei einem Fahrzeug im Luxussegment beträgt die Anzahl der Codezeilen heute in der Regel mehr als 150 Millionen.

Die Entwicklungen im Bereich automatisiertes Fahren, Fahrzeug-konnektivität, elektrische Antriebsstränge und Shared Mobility (ACES) werden die Komplexität zukünftig weiter steigern und die Software an die Spitze der Automotive-Wertschöpfungskette setzen. Prognosen zufolge erfordert der Betrieb selbstfahrender Fahrzeuge mehrere Milliarden Codezeilen. Angesichts dieser hohen Komplexität wächst die Bedeutung der Entwicklung von Embedded-Anwendungen, da die Entwicklung innovativer und leistungsfähiger Software zum wichtigsten Differenzierungsmerkmal für Automotive- und Mobilitätsunternehmen wird.

Weitere Informationen über die Unterstützung, die Lösungen von Siemens Ihnen bieten können, finden Sie unter [siemens.com/aes](https://www.siemens.com/aes). Hier werden zusätzliche Inhalte wie Blogs, White Papers, Podcasts, Produktvideos, Webinare, Lösungsfunktionen und Infografiken bereitgestellt.

## 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 Warenzeichen von Siemens findet sich [hier](#). Alle anderen Marken sind Eigentum der jeweiligen Inhaber.

81256-82104-C1-DE 6/20 LOC