

Neue Entwicklungsmethoden in einen konsistenten Prozess integrieren

Herausforderung Software

Zukünftige Fahrzeuge werden automatisiert, elektrisch, vernetzt und Software-definiert sein. Ein Benutzererlebnis, das weitgehend durch Software bestimmt wird, soll nicht nur die beste Technik, sondern auch umfassende Mobilitätskonzepte und digitale Dienstleistungen bereitstellen. Software wird zum künftigen Treiber von Wertschöpfung.

Gaurav Tomar

Während für Komponenten- und Systementwickler Software ein Höchstmaß an Funktionalität und Effizienz aus einem konstruierten System ermöglicht, betrachtet die Digitalabteilung Automobil-Software umfassender. Für sie verbessert Fahrzeug-Software nicht nur die Funktionen, sondern erreicht auch jene Mobilitätsziele, die neuen Mehrwert für das Unternehmen generieren. Daher ist eine klare

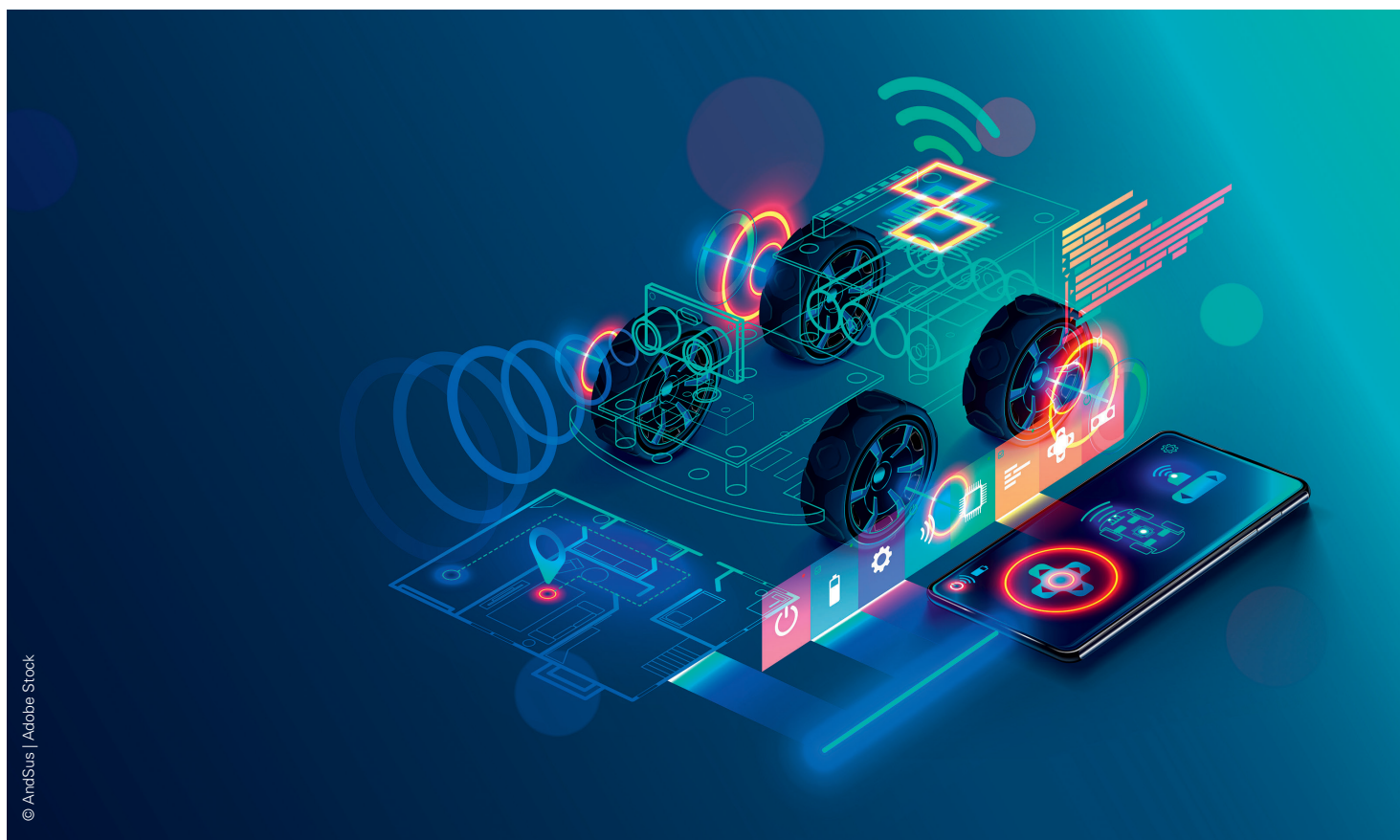
Strategie, wie das Software-definierte Fahrzeug entwickelt werden soll, unabdingbar.

Herausforderungen bei der Entwicklung

Automobilhersteller gehen verschiedene zentrale Entwicklungshürden an – darunter die späte Erkennung von Mängeln und fehlendes domänenüber-

greifendes Wissen und Kompetenz in der Systementwicklung. Diese Herausforderungen wurden aber in jüngster Zeit durch technologische Fortschritte wie künstliche Intelligenz und Software-definierte Systeme verschärft.

Die Automobilindustrie setzt bereits Modellierung und Simulation ein, um Design-Fehler frühzeitig zu erkennen, und verwendet dazu Modellierungsrichtlinien und Modellbibliotheken. Auf-



© AndSus | Adobe Stock

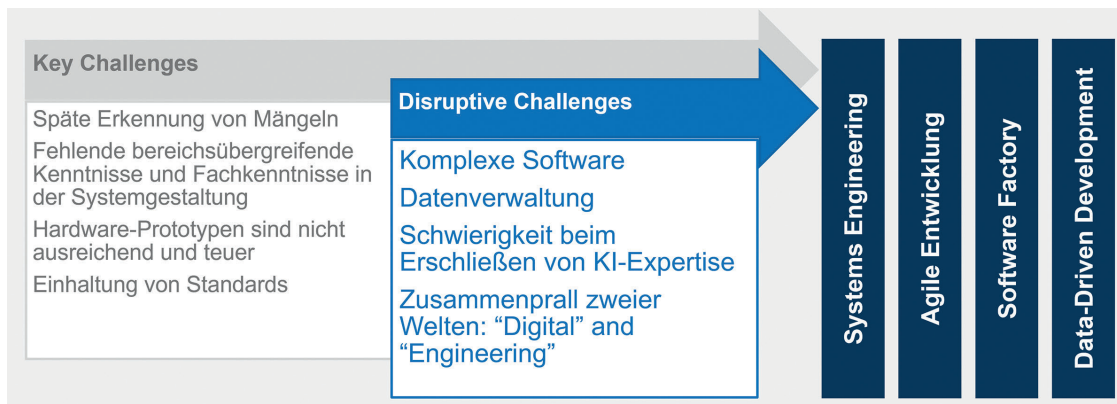


Bild 1: Zusammenfassung der wichtigsten Herausforderungen und neuen disruptiven Probleme mit Strategien, um sie anzugehen

© MathWorks

grund der komplexen Architekturen, die heutige Fahrzeuge erfordern, verlagert sich diese Herausforderung von der Funktionsebene auf die Systemebene. Hier sind Design-Fehler nicht nur spezifisch für die jeweilige Funktion, sondern es entstehen auch Wechselwirkungen über Funktionsgrenzen hinweg. Funktionen, wie Fahrerassistenzsysteme und automatisiertes Fahren erfordern zudem nicht nur die Kenntnis neuer Disziplinen wie Computer Vision, Sensorfusion und Pfadplanung, sondern auch Prozesse und Standards zur Integration dieser neuen Software-basierter Technologien in das System.

Um noch weniger auf Prototypen angewiesen zu sein sowie häufige Software-Bereitstellungen und komplexe Testfälle zu bewältigen, müssen Umfang und Fähigkeiten simulationsbasierter Tests und Validierungen für Modelle und Code ausgebaut werden. Automobilhersteller und -zulieferer verantworten heute zudem gemeinsam die Entwicklung von Software. Das erfordert neue Strategien für die Erfüllung von Nachweisbarkeit, Versions- und Varianzmanagement, Governance und Testfähigkeit. Darüber hinaus sind Herausforderungen zu beobachten, die grundlegend neu und disruptiv sind.

Komplexe Software und Datenverwaltung

Die Architektur von Fahrzeug-Software ändert sich von vielen funktions-spezifischen Steuergeräten (ECUs) hin zu einem gemischten Ansatz mit funktions-spezifischen Steuergeräten, Domain-ECUs, zonalen Steuergeräten und Fahrzeugsteuergeräten in einer Service-orientierten Architektur (Service Oriented

Architecture, SOA). Diese Architektur muss auch die zukünftige Nutzung von Cloud-Infrastrukturen berücksichtigen und stützt sich auf Technologien, die für Automobilanwendungen noch relativ neu sind, wie Linux (oder andere Fahrzeug-Betriebssysteme), Edge Enabler, Automotive Ethernet und Service-orientierte Software-Interaktion. Statt gespeicherter Testdaten treffen jetzt Petabytes live ein und müssen nach Geschäftsfällen wie Wartung, Weiterentwicklung oder Dienstleistungen getrennt und aggregiert werden.

Erschließen von KI-Expertise sowie Einbindung neuer Experten

Technikteams müssen KI-Expertise finden und wirksam in der Produktentwicklung nutzen. Komplexe KI-Systeme lernen derzeit zwar aus realen Situationen, ihr Einsatz als Primärsysteme wird aber noch eine ganze Weile dauern. Zum Aufbau neuer Fähigkeiten werden zudem neben Software-Ingenieuren auch IT-Experten eingestellt, die eigene Prozesse, Methoden und Tools mitbringen. Anfangs fehlt ihnen u.U. die Stringenz, die für die Entwicklung automobilreifer Software und zur Einhaltung von Standards wie ISO 26262 und ASPICE erforderlich ist. Methodik-, Prozess- und Kulturkonflikte sind die Folge.

Als Reaktion auf diese breiteren und neuen disruptiven Herausforderungen erweitern Automobilunternehmen ihre Entwicklungsprozesse durch (**Bild 1**):

- Einführung neuer Vorgehensweisen zur frühzeitigen Erkennung von Mängeln und zum Aufbau von domänenübergreifendem Wissen
- Neuorganisation der Technikteams

mit Ausrichtung auf Software-Methodiken

- Einrichtung neuer Infrastruktur und automatisierter Prozesse zur Vorverlegung durch Virtualisierung
- Zugriff der Technikteams auf Flottendaten für datenbasierte KI und Softwareentwicklung und um Einblicke in die Verwendung von Funktionsmerkmalen durch Endbenutzer zu gewinnen

Etliche Methoden werden allmählich von Automobilherstellern und -zulieferern übernommen. Um mit ihrer Hilfe jedoch die physikalischen Aspekte des Fahrzeugs zu berücksichtigen sowie Sicherheits- und Auditanforderungen zu erfüllen, ist ein frischer Blick darauf erforderlich.

Systems Engineering

Systems-Engineering-Ansätze wurden für komplexe, domänen-übergreifende Systeme und zur Einhaltung hoher Sicherheitsstandards eingeführt. Modelle als lebende Darstellungen von Spezifikationen, Entwürfen und Implementierung werden in Simulationen getestet und liefern frühzeitiges Feedback. Wie kann Systems Engineering die rasche Bereitstellung von Software ermöglichen, ohne auf die viel langsamere Entwicklung von Elektronik und Mechanik warten zu müssen?

Agile Entwicklung

Agile Entwicklung schafft Mehrwert und schützt davor, mit hohem Aufwand falsche, unvollständige oder geänderte Anforderungen zu dokumentieren. Die testbasierte kontinuierliche Bereitstellung führt zur schnelleren Marktreife

Integration von Systems Engineering in Software Entwicklungsmethoden		Auswahl einer cloudfähigen Toolchain	Integration modell- und datenbasierter Entwicklungsabläufe
	Anpassung der agilen Entwicklung an die Automobiltechnik		
Standardisierung der Toolchains in der Software Factory		Auswahl von Automatisierungselementen und -zeitpunkten in der Software Factory	Finden, Pflegen und Erschließen interner KI- und Softwareentwicklungsfähigkeiten

Bild 2: Empfehlungen für Entscheidungsträger © MathWorks

von Artefakten. Das sind nicht notwendigerweise Software-Komponenten oder Code, sondern auch schnelle Tests auf der Fahrzeugebene zur Prüfung gegenseitiger Abhängigkeiten. Wie kann eine schnelle und inkrementelle Software-Bereitstellung mit weniger Nacharbeit bei Einhaltung von Standards und Vorschriften für alle Produktvarianten gelingen?

Software Factory

Software Factories automatisieren Entwicklung und Wartung der Varianten eines Grundprodukts durch Anpassung, Montage und Konfiguration Framework-basierter Komponenten. Ein erheblicher Anteil der Architektur- und Funktionsentwicklung findet jedoch während der Erkundungsphase statt, wenn Testabläufe noch nicht vollständig definiert sind. Was ist die richtige Methode und der richtige Zeitrahmen, um von der Software-Fabrik und ihrer Automatisierungsfähigkeit zu profitieren?

Datengesteuerte Entwicklung

Flottendaten sind die Basis der Entwicklung von Machine- und Deep-Learning-Funktionen (KI-basierte Funktionen) sowie deren Verbesserung im laufenden Betrieb. Diese Daten sollten gut zugänglich und das Abrufen relevanter Daten wohldefiniert sein. Wie integrieren

Unternehmen datenbasierte Funktionen in die Simulationsinfrastruktur und den Fahrzeugentwicklungsprozess? Wie validieren sie diese Algorithmen unabhängig vom Anwendungskontext?

Empfehlungen für Entscheidungsträger

Anknüpfend an die bis hierher vorgestellten Herausforderungen empfehlen wir Entscheidungsträgern, im Rahmen ihrer Risikominderung die folgenden sieben Strategien zu berücksichtigen (Bild 2):

- Integration von Systems Engineering in Softwareentwicklungsmethoden
- Anpassung der agilen Entwicklung an die Automobiltechnik
- Auswahl einer Cloud-fähigen Werkzeugkette
- Standardisierung der Toolchains in der Software Factory
- Auswahl von Automatisierungselementen und -zeitpunkten in der Software Factory
- Finden, Pflegen und Erschließen interner KI- und Software-Entwicklungsfähigkeiten
- Integration modell- und datenbasierter Entwicklungsabläufe

Systems Engineering in Software-Entwicklungsmethoden integrieren

Zur schnellen Bereitstellung von Fahrzeugsoftware sollte Systems Enginee-

ring weiterentwickelt und in die etablierte Software-Entwicklung integriert werden. Ausführbare Spezifikationen in der Systemtechnik können systembezogenes Know-how zum Tragen kommen lassen, Verifikation und Validierung von Anforderungen nach vorne verlagern und Hardware-Einschränkungen für die Software frühzeitig aufzeigen.

Agile Entwicklung an Automobiltechnik anpassen

Die agile Entwicklung lässt sich an die Entwicklung automobilreifer Fahrzeugsoftware anpassen, indem man:

- Kundenanforderungen in kürzeren Sprints klärt und bewertet, ob sich eine Technologie für die anstehende Aufgabe eignet. Die Reife der Vision sowie des Ziels entscheidet über den Zweck des jeweiligen Sprints, also Analyse, Entwurf oder Test. Das Ergebnis ist eine Systemsimulations-Demo, keine Software-Komponente.
- Mit längeren Sprints ein neues Produkt entwickelt, für das bereits Grundrisiken verringert wurden, oder ein in Verwendung befindliches Produkt weiterentwickelt oder verbessert. Ergebnis ist eine Systembereitstellungs-Demo mit der Software-Komponente in einer Hardware-in-the-Loop (HiL) oder virtuellen HiL. Im Sprint entsteht so kontinuierliches Feedback, das inkrementelle Änderungen berücksichtigt. Fahrzeugtests werden in größerem Maßstab virtualisiert und simuliert, um sie in Sprints einzubeziehen.

Cloud-fähige Toolchain auswählen

Software-Erstellung und -tests sowie Simulation verlagern sich in die Cloud, und das umso mehr, da in der Cloud interaktiv entwickelt wird und sich Anwendungsfälle skalieren lassen. Zudem verspricht der Wechsel in die Cloud erleichterte Teamarbeit und das Aufbrechen von Silos. Unternehmen sollten ihre Cloud-Strategie sorgfältig entwickeln und sicherstellen, dass erwartete Risiken nicht nur erarbeitet, sondern auch abgewehrt und bewältigt wurden, wobei Cloud-fähige Werkzeugketten helfen. Eine Cloud-fähige Toolchain muss also DevOps für primär drei Anwendungsfälle unterstützen:

- Entwicklung und Flottentests,
- kontinuierliche Systementwicklung und Feldaktualisierungen durch OTA sowie
- Mobilitätsprodukte und -dienste.

Standardisierung der Toolchains in der Software Factory

Kontinuierliche Bereitstellung (Continuous Delivery, CD) ist eine Schlüsselfähigkeit für ständig etwa mittels OTA-/FOTA-Updates aktualisierbare Fahrzeugplattformen. CD hilft bei der Umsetzung von DevOps durch Verzahnung von Entwicklung (Dev) und Betrieb (Ops). Interoperabilitätsprobleme zwischen vorhandenen und neuen Tools oder Abhängigkeiten zwischen alten und neuen Standards sind oft schwer zu beheben. Standardisierte Werkzeugketten helfen, sich auf hochwertige Ergebnisse und Wertschöpfung zu konzentrieren, da sie die Einhaltung von Prozess- und Sicherheitsstandards sowie ASPICE, ISO 26262 und ISO 21434 erleichtern. Es empfiehlt sich, eng mit wichtigen Anbietern zusammenzuarbeiten und ihnen Anforderungen mitzuteilen, um frühzeitig Zugang zu Technologien zu erhalten, indem Technologie-Roadmaps beeinflusst werden. Zudem lässt sich das Anbieter-Ökosystem nutzen, um unternehmensintern auf die benötigte Technologie vorzubereiten, insbesondere, wenn sich die Software-Entwicklungsmethoden ändern.

Automatisierungselemente und -zeitpunkte in der Software Factory

Um kontinuierlich Mehrwert – in Form von Modellen, Software oder Artefakten – bereitzustellen, sollten routinemäßige und häufige Aufgaben automatisiert werden. Zuerst empfiehlt es sich, eine Systemsimulation einzurichten, um Lücken in den Spezifikationen zu finden und zu schließen sowie Nacharbeit zu minimieren, indem zu frühe Automatisierung vermieden wird. Nach Implementierung der Spezifikationen als Modelle und Validierung durch Simulationen besteht größeres Vertrauen in die beabsichtigte Funktionalität – das ist der richtige Zeitpunkt für die Automatisierung.

Entscheidend ist die Wahl von Werkzeugen und Methoden, die ein frühes Feedback geben. Empfehlenswert sind Model-in-the-Loop (MiL) und frühzeitige

Software-in-the-Loop (SiL) mit Code- und Modellintegration und Simulation. Dieses Schlüsselkonzept trägt auch dazu bei, agile Methoden für die Automobiltechnik und die Entwicklung von Software für die Automobilindustrie anzupassen.

Interne KI- und Software-Entwicklungsfähigkeiten

Vorhandene Erfahrung mit KI und Software-Entwicklung im Unternehmen sollte man nutzen. Es ist notwendig, den aktuellen Reifegrad der Teams in Hinsicht auf Modell- und Code-basierte Ansätze zu beurteilen sowie führende und rückständige Metriken zu ermitteln, die dann verfolgt werden. Für die führenden Metriken ist es hilfreich, Fachwissen zu identifizieren und für kontinuierliche Verbesserung einzusetzen. Kompetenzzentren tragen zur Verbesserung hinterherhinkender Kennzahlen bei.

Integration modell- und datenbasierter Entwicklungsabläufe

Obwohl modellbasierte und datengesteuerte Ansätze oft als inkompatibel gelten, haben Automobilteams große Erfahrung mit der Integration der beiden. In Fahrzyklustests werden etwa reale Fahrdaten erfasst, die wiederum in eine Verbesserung der Designmodelle und damit eine erhöhte Batterieleistung fließen, worauf der aktualisierte Entwurf gegen den Validierungsdatensatz erneut validiert wird.

Arbeiten die Lebenszyklen für Daten und Modelle Hand in Hand, unterstützen sie sich gegenseitig und bestätigen nicht nur das Software-Design, sondern auch die elektromechanische und sensorische Auslegung sowie beliebige Kombinationen davon. Diese Kombination aus datengesteuerten und modellbasierten Ansätzen sollte sich nicht auf Fahrassistenzsysteme beschränken, sondern auch in die Entwicklung von Hybrid- oder Elektroantrieben, Motoren und Batterien fließen.

■ (eck)

www.mathworks.com



Gaurav Tomar ist Industry Manager EMEA Automotive bei MathWorks © MathWorks

eMove360° Europe 2022

6th International trade fair for Mobility 4.0

electric – connected – autonomous

5 – 7 October 2022, Messe Berlin

eMove360° trade fair goes Berlin



www.emove360.com

