



© NicoElNino | AdobeStock

## Durchgängige Entwicklungsmethoden für Software-definierte Fahrzeuge

# Steigende Komplexität beherrschen

Durch den zunehmenden Einsatz von Elektronik und Software steigt die Systemkomplexität im Fahrzeug. Gleichzeitig nehmen aufgrund kürzer werdender Entwicklungszyklen Fehlerquellen weiter zu. IPG Automotive und TraceTronic haben das Zusammenspiel ihrer Systeme für einen durchgängigen Testprozess in jeder Entwicklungsphase von Fahrzeug-Software optimiert.

**Martin Herrmann, Henning Kemper, Michael Gebauer und Willi Meierhof**

Fahrerassistenzsysteme und autonome Fahrfunktionen gewinnen zunehmend an Bedeutung. Aufgrund der Vielzahl der zu berücksichtigenden Parameter müssen diese Systeme mithilfe von Fahrzeugprototypen in unterschiedlichen Szenarien und über viele Kilometer erprobt und validiert werden. Aufgrund des hohen Aufwands ist das allein auf Basis realer Fahrversuche nicht umsetzbar. Darüber hinaus können keine vergleichbaren Versuchsergebnisse erzielt werden, weil komplexe reale Ver-

kehrssituationen mit mehreren Verkehrsteilnehmern nicht vollständig reproduzierbar sind. Eine Verlagerung der Tests in frühe Entwicklungsstadien von Fahrfunktionen ist somit unumgänglich.

Das gilt vor allem für die Erprobung kritischer Szenarien, sogenannter Edge Cases. Diese sind für die Absicherung automatisierter Fahrfunktionen von besonderer Bedeutung. Sie treten zwar im realen Verkehr nur selten auf, bergen aber eine Gefahr für Mensch und Material. Durch den Einsatz einer Simu-

lationsumgebung ist es möglich, entsprechende Szenarien gezielt zu entwickeln und im virtuellen Fahrversuch zu testen. Der Testdurchsatz kann dabei durch den Einsatz von Testautomatisierungstools deutlich erhöht werden.

Im Zusammenspiel mit Tools zur Unterstützung agiler Software-Entwicklungsprozesse bildet die virtuelle Fahrzeugentwicklung eine optimale Basis, um Fehler frühzeitig zu erkennen und den Anforderungen an die Entwicklung hochautomatisierter Fahrfunktionen gerecht zu werden.

**Szenarienbasiertes Testen**

Das szenarienbasierte Testen stellt eine Methode des Fahrversuchs dar, die im Rahmen der Projekte der PEGASUS-Familie etabliert wurde [1]. Sie trägt wesentlich zur Reduzierung des Testaufwands in der Simulation bei, indem hauptsächlich die für die Fahrfunktion relevanten Szenarien untersucht werden, anstatt eine bestimmte Anzahl notwendiger Testkilometer zu definieren.

Dabei profitiert der Entwicklungsprozess von realistischen Einsatzbedingungen in einer virtuellen Umgebung. So lässt sich ein Großteil der Szenarien simulieren, um nur noch wenige Tests auf dem Prüfgelände beziehungsweise im Feld durchführen zu müssen. Die verbleibenden realen Fahrversuche dienen vor allem zur Validierung der Simulationsergebnisse.

**Simulationsplattform**

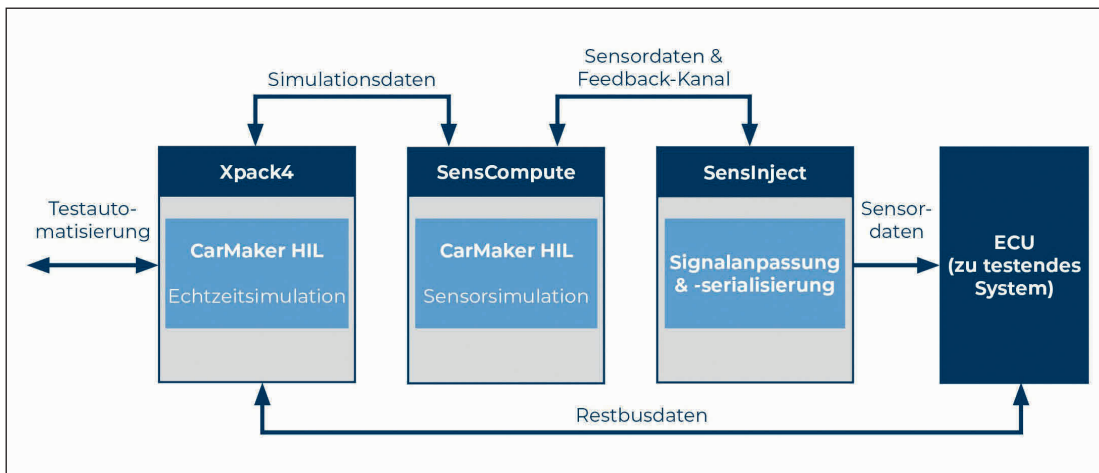
Die Basis dafür bildet eine Simulationsplattform, die einen durchgängigen Entwicklungsprozess mithilfe von Model-in-the-Loop-, Software-in-the-Loop-, Hardware-in-the-Loop- und gegebenenfalls Vehicle-in-the-Loop-Methoden (MiL, SiL, HiL, ViL) ermöglicht. Mit der Simulationsplattform CarMaker lässt sich der Software-basierte Teil des Fahrzeugs in die Simulation integrieren und durchgängig testen.

Das System-under-Test (SuT) besteht aus Modulen zur Umgebungsperezeption und Eigenlokalisierung sowie zur Bewegungsplanung und -regelung. Abhängig vom Anwendungsfall kann es über verschiedene Methoden in die Simulation integriert werden. Je nach Entwicklungsstand erfolgt die Anbindung signalbasiert oder auf Datenproto-

kollebene. Dabei werden alle gängigen Industriestandards unterstützt.

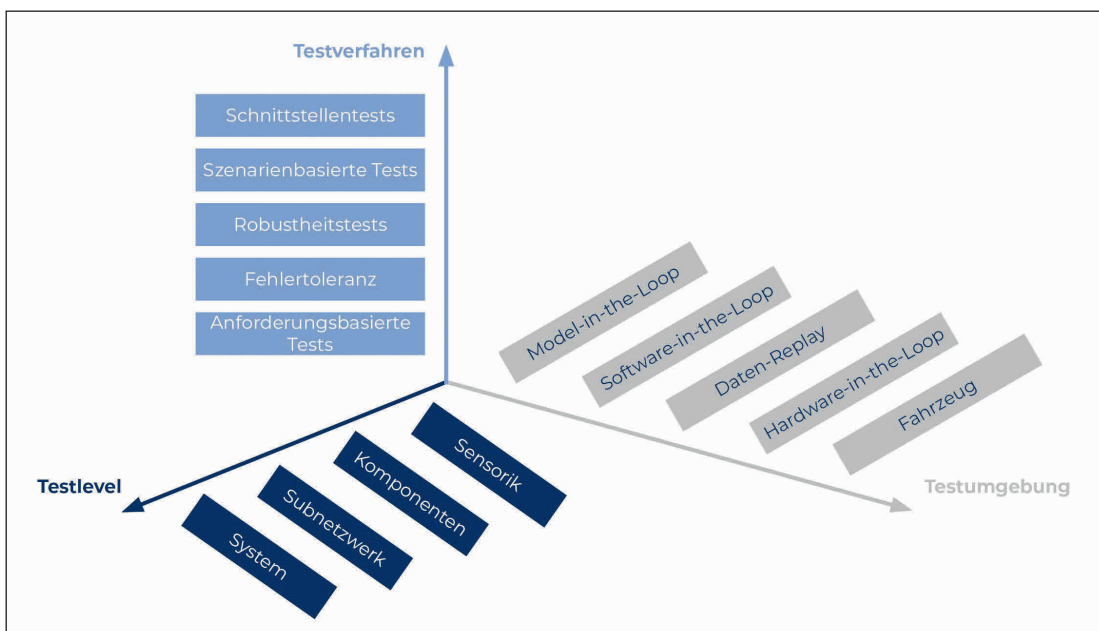
Neben der Ausprägung der Datenschnittstelle ist auch die Wahl der Integrationsstufe der Software-Module entscheidend für die Implementierung der Schnittstellen. Je nach zu untersuchendem Systembestandteil ist eine bestimmte Anzahl geeigneter Ersatzmodelle erforderlich, um das Gesamtsystem in der Simulationsumgebung realitätsnah abzubilden. Abhängig davon, ob die Objektdetektion Teil des SuT ist, muss der virtuelle Prototyp die benötigte Schnittstelle flexibel bedienen können, beispielsweise mit unverarbeiteten Bilddaten oder Objektlisten.

Für den Test mit Perzeptionsmodulen werden GPU-basierte Sensormodelle benötigt, um Sensorrohdaten zu generieren. Für das Rendering von Kamerabildern sind beispielsweise verschie-



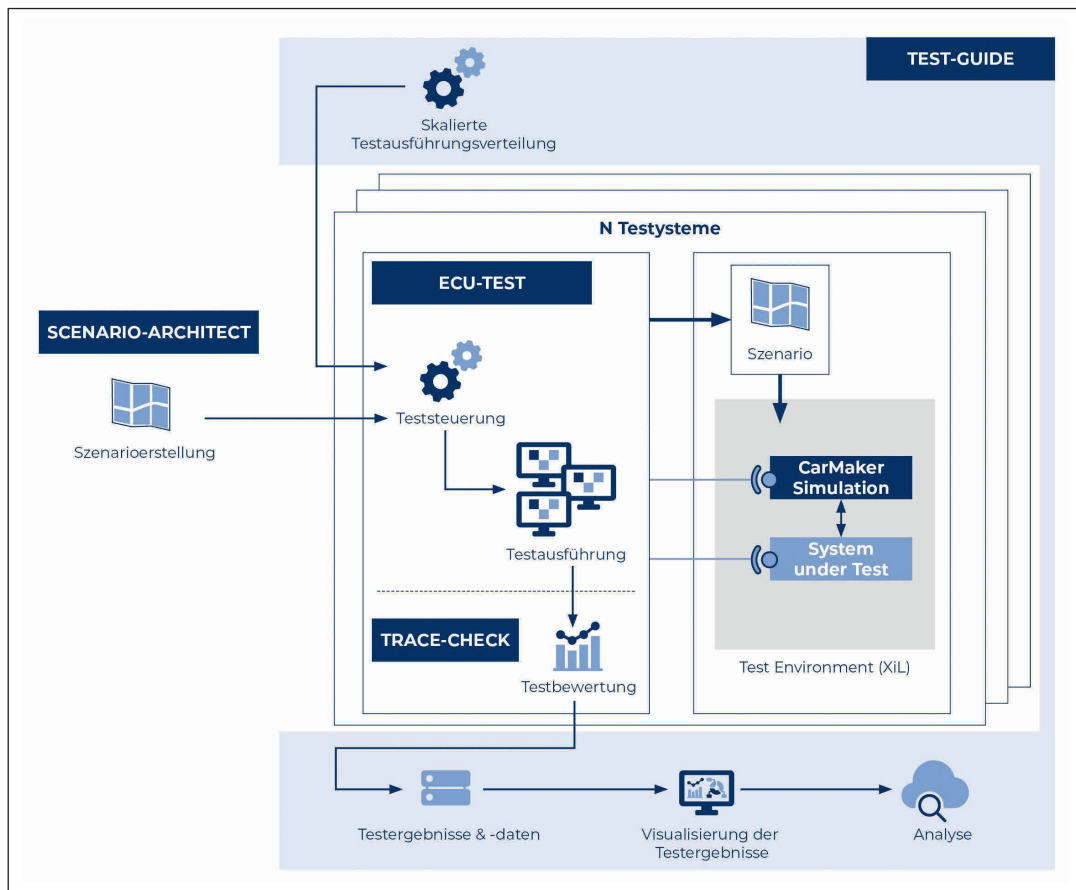
**Bild 1: ADAS/AV HiL mit Sensordateninjektion**

© TraceTronic



**Bild 2: Kombinationsmöglichkeiten von Testverfahren, -umgebung und -level**

© TraceTronic



**Bild 3: Integrierte Toolkette für CI/CT im Entwicklungsprozess**  
© TraceTronic

dene Modellbestandteile erforderlich: die Umgebungssimulation einschließlich Beleuchtungs- und Wettereffekten, ein Linsen- sowie ein Imager-Modell.

Die Integrationsstufe ist auch für die Cloud-basierte Simulation relevant. In der Regel wird ein Großteil der Versuche in der Cloud auf CPU-basierten Instanzen durchgeführt. Dieser Ansatz kommt hauptsächlich bei Tests der Bewegungsplanung und -regelung zum Einsatz. Weil GPU-basierte Instanzen im Vergleich zu CPU-basierten Instanzen deutlich kostenintensiver sind, wird der Rechenaufwand für Closed-Loop-Simulationen damit soweit wie möglich reduziert. Nur wenige ausgewählte Szenarien werden in einer SiL-Umgebung in der Cloud getestet. Alternativ können diese auch vollständig in nachgelagerten Stufen berücksichtigt werden, etwa bei HiL mit Einspeisung synthetischer Sensordaten.

Sollen reale Steuergeräte in die Simulation eingebunden werden, kann die HiL-Funktionalität von CarMaker genutzt werden. Um die Fahrfunktion in die Ziel-Hardware zu integrieren und deren Lauffähigkeit zu gewährleisten, ist es erforderlich, den erwarteten Ein-

gangs- und Ausgangsschnittstellen der Fahrfunktion die notwendigen Daten bereitzustellen. Relevant sind hier unter anderem die Protokolle CAN, FlexRay und Automotive-Ethernet.

Werden Tests mit Perzeptionsanteil durchgeführt, muss zusätzlich eine Einspeisung synthetischer Sensordaten in das Steuergerät mittels Sensorschnittstellen wie GMSL oder FPD-Link erfolgen. Entscheidend ist dabei, dass die Simulation in Echtzeit ausführbar ist und dass alle benötigten Daten rechtzeitig zur Verfügung stehen. Dies stellt insbesondere für die physikalische Sensorsimulation eine große Herausforderung dar. Mit GPU-basierten Modulen zur Generierung der Sensorrohdaten (SensCompute) sowie Hardware zur Serialisierung und Injektion der Daten (SensInject) existieren speziell entwickelte Lösungen, die dieser Herausforderung begegnen (**Bild 1**).

### CI/CT im Entwicklungsprozess

Moderne Fahrzeugfunktionen werden durch verteilte Systeme mit unterschiedlichen Sensoren und Aktoren abgebildet. Diese Komponenten verfügen

jeweils über eigene Software, müssen aber dennoch reibungslos zusammenarbeiten. Um diese Interaktionen funktional sicherstellen zu können, muss sowohl die Erstellung der Software-Komponenten als auch deren übergreifender Software-Test kontinuierlich erfolgen. Dieser Prozess wird als Continuous Integration/Continuous Testing (CI/CT) bezeichnet [2]. Entwicklungsteams erhalten auf diese Weise schnell Feedback auf Änderungen im Softwarecode und können Fehler frühzeitig identifizieren und beheben.

Die Basis für CI/CT bilden eine einheitliche Toolkette und durchgängige Workflows. Zudem ist es erforderlich, verschiedene Tests auf unterschiedlichen Ebenen und Reifegraden des Gesamtfahrzeugs durchzuführen – vom Sensor- und Komponententest über den Verbundtest bis hin zum Systemtest. Die Prüfung der Funktionen sollte dabei für jede einzelne Integrationsstufe frühzeitig erfolgen. Je eher Fehler identifiziert und behoben werden, desto geringer lassen sich die Entwicklungskosten halten. **Bild 2** zeigt, dass die einzelnen Tests in drei Dimensionen angeordnet werden können und sich die Vorgehensweise

nicht auf eine bestimmte Form am virtuellen Prüfstand beschränkt, sondern im Gegenteil sehr vielfältig ist. Neben den bereits erwähnten szenarienbasierten Tests werden auch die Robustheit, die Fehlertoleranz und die Schnittstellen geprüft.

Die enorme Menge notwendiger Testfälle in unterschiedlichen Entwicklungsumgebungen (MiL, SiL, HiL) ist nur durch eine Vielzahl von Schnittstellen zu unterschiedlichen Werkzeugen beherrschbar. Mit der TraceTronic-Toolkette lassen sich Software-Tests in allen genannten Umgebungen automatisiert durchführen, um in Verbindung mit geeigneten Methoden eine ganzheitlich hohe Software-Qualität sicherzustellen. Die Automotive DevOps Plattform gewährleistet, dass alle Beteiligten über den gesamten Entwicklungsprozess hinweg eine einheitliche Tool-Kette verwenden und durchgängige Workflows etablieren können.

Beim szenarienbasierten Testen steht – neben einem geeigneten Szenario – auch die Spezifikation der Testfälle auf Basis definierter Anforderungen im Vordergrund. Über eine grafische Benutzeroberfläche lassen sich mit dem Tool Scenario-Architect Fahr-szenarien intuitiv erstellen und dynamisch zu verschiedenen Zeitpunkten aus unterschiedlichen Blickwinkeln betrachten. Der dabei zugrundeliegende OpenSCENARIO-Standard gestattet es, diese Szenarien anschließend in der offenen Integrations- und Testplattform

CarMaker auszuführen. Das Szenario kann direkt in einen Testfall überführt werden, um die jeweilige Anforderung zu überprüfen. Mithilfe des Automatisierungstools ECU-Test ist es möglich, die zur Verfügung stehenden Schnittstellen der gesamten Testumgebung zur Simulation und Analyse zu nutzen. Durch die Abstraktion der Testfälle mittels generischer Tool-Zugriffe wird die Wiederverwendbarkeit der Versuche sichergestellt. Darüber hinaus können die Testfälle in sämtlichen Entwicklungsstufen genutzt werden. Durch unterschiedliche Konfigurationen lassen sich die Tests dabei konkret an das jeweilige Umfeld anpassen.

Eine effiziente und hochskalierte Verteilung von Testfalldurchführungen auf verschiedenen Testsystemen wird durch die Ausführungsverteilung des zentralen Tools Test-Guide im vollautomatisierten CI/CT-Workflow erreicht. In sogenannten Playbooks werden alle dafür notwendigen Schritte definiert. Auf dieser Basis lassen sich die relevanten Daten auf den Testsystemen verteilen. Im hier beschriebenen Konzept werden die Szenarien und die Software für das SuT von der Simulationsplattform CarMaker aufgenommen und in der gewählten Testumgebung ausgeführt (Bild 3).

Um die Testabdeckung textuell und grafisch zu ermitteln, wird während der Testdurchführung eine Vielzahl an Versuchsreports erstellt. Die Analyse-möglichkeiten dieser Reports sind dabei so gestaltet, dass unterschiedliche Stake-

holder die für ihre Rolle passenden Daten angezeigt bekommen. Beispielsweise erhält ein Test-Manager einen Report mit dem Überblick über den gesamten Testprozess und ein Funktionsentwickler eine detaillierte Auswertung zu einem spezifischen Testfall. ■ (eck) [www.ipg-automotive.com](http://www.ipg-automotive.com) [www.tracetronic.de](http://www.tracetronic.de)

### Quellenverzeichnis

[1] PEGASUS-Projekt, „PEGASUS METHOD – An Overview“, 2019. [Online]. Verfügbar: <https://www.pegasus-projekt.de/files/tmpl/Pegasus-Abschlussveranstaltung/PEGASUS-Gesamtmethode.pdf>. [Zugriff am 03.08.2023]

[2] Schneider, A.: CI/CT/CD in ADAS Development – Application of a CI/CT/CD environment to develop ADAS functions, Apply & Innovate, Karlsruhe, 2022



**Martin Herrmann** ist Business Development Manager ADAS & Automated Driving bei IPG Automotive. © IPG Automotive



**Henning Kemper** ist Specialist Editor bei IPG Automotive. © IPG Automotive



**Michael Gebauer** ist Product Owner im Bereich ADAS & Automated Driving bei TraceTronic. © TraceTronic



**Willi Meierhof** ist System Engineer im Bereich ADAS & Automated Driving bei TraceTronic. © TraceTronic