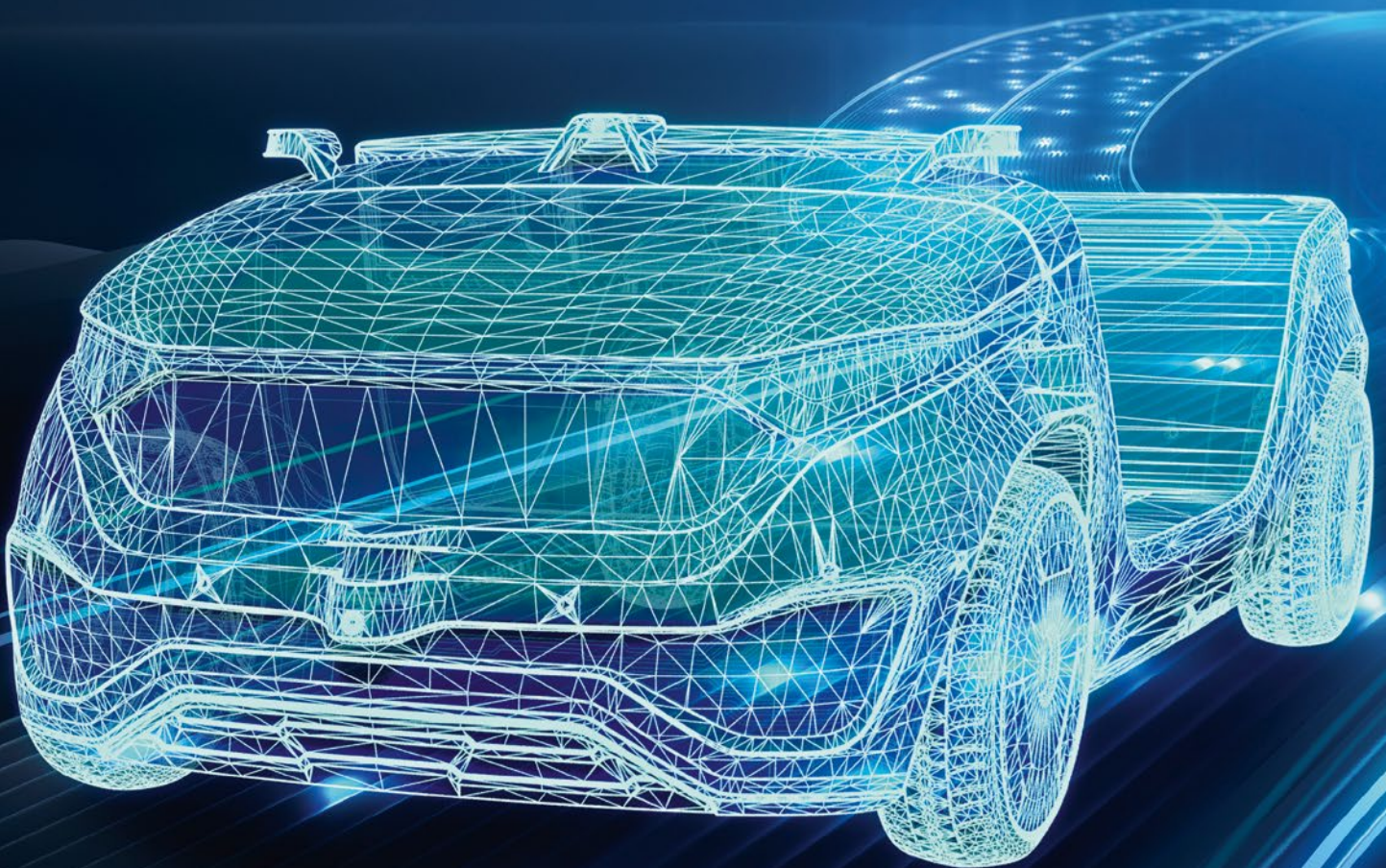


# Virtueller Test automatisierter Fahrfunktionen

Die zunehmende Komplexität und Variantenvielfalt automatisierter Fahrfunktionen lässt sich durch eine rein anforderungsbasierte Testmethodik nicht mehr zuverlässig beherrschen. Im Kontext der Plattform für hochautomatisiertes Fahren HARRI entwickelt Bertrand Methoden und Tools für den szenarienbasierten Test der zugehörigen Funktionen und Elektronik weiter.



## AUTOREN



**Dr.-Ing. Torsten Butz**  
ist Abteilungsleiter Elektronik-  
Entwicklung Fahrwerk, Antrieb und  
Automatisiertes Fahren bei  
Bertrandt in Ingolstadt.



**Dipl.-Wirt.-Ing. (FH)  
Simon Paleduhn**  
ist Teamleiter Systementwicklung  
und Testmanagement bei Bertrandt  
in Ingolstadt.



**Dipl.-Ing. (FH) Alexander Merkel**  
ist Abteilungsleiter Elektronik-  
Entwicklung Diagnose, Komfort,  
ADAS und Powertrain bei Bertrandt  
in Ehningen.



**Dipl.-Ing. (FH)  
Christian Bohner, MBA**  
ist Teamleiter Virtueller Test  
bei Bertrandt in Ingolstadt.

## MOTIVATION

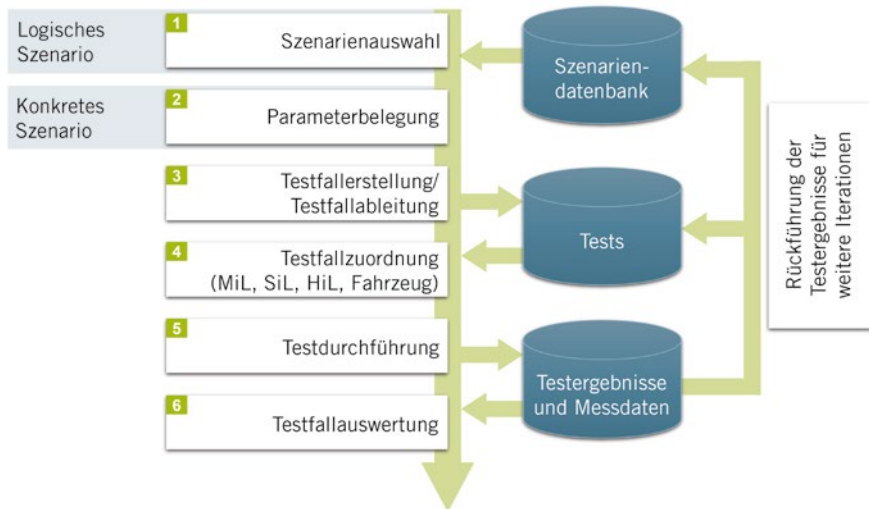
Zur Verbesserung der Verkehrssicherheit und stetigen Automation des Straßenverkehrs finden vermehrt Fahrerassistenzsysteme und automatisierte Fahrfunktionen Einsatz. Die steigende Komplexität der zugrundeliegenden Elektronik- und Softwarekomponenten, agile Entwicklungsmethoden sowie die Reduzierung von Erprobungsfahrzeugen erhöhen auch die Anforderungen an geeignete Absicherungsmethoden [1, 2]. Herkömmliche Testverfahren nach anforderungsbasiertem Ansatz allein sind nicht mehr ausreichend, um in immer kürzer werden-

den Entwicklungszyklen die Abdeckung aller notwendigen Absicherungsaspekte hinsichtlich Funktion, Robustheit und Sicherheit zu erreichen. Eine szenarienbasierte Testmethodik und eine konsequente Virtualisierung ermöglichen die funktionsunabhängige Darstellung der Testbedingungen und eine Verschiebbarkeit von konkreten Testaufwänden in Richtung virtueller Testorte.

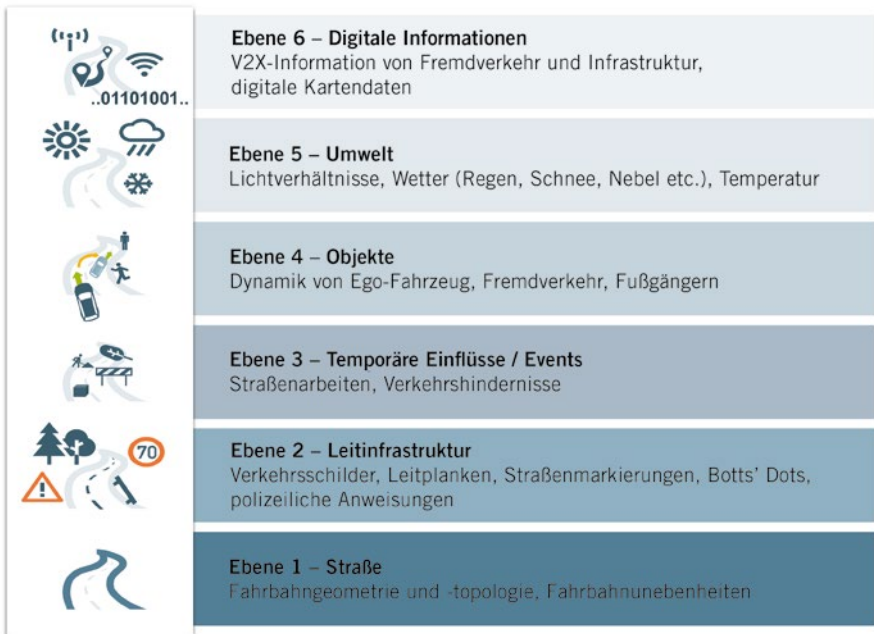
**TESTMETHODIK**

Die weiterentwickelte szenarienbasierte Testmethodik, **BILD 1**, ist eine Ableitung aus den Forschungsergebnissen des BMWi-geförderten Projekts Pegasus [3] und den normativen Standards der Safety of the Intended Functionality [4]. Wesentlicher Bestandteil sind funktionale Test-szenarien, die über eine abstrakte sprachliche Beschreibung definiert sind. Ihre Parametrierung erfolgt über statische Informationen, wie Straßengeometrien und Wetterbedingungen, sowie dynamische Informationen, wie Manöverbeschreibungen von Ego-Fahrzeug und anderen Verkehrsteilnehmern. Durch die Übersetzung in das universelle Format OpenScenario [5] und die Festlegung der infrage kommenden Parameterräume entstehen logische Szenarien, die eine Tool-unabhängige Verwendbarkeit sowie eine Portierbarkeit zwischen den Testorten gewährleisten sollen. Die Ablage erfolgt in einer Szenariendatenbank, die einen funktionsunabhängigen Szenarienkatalog darstellt.

Basierend auf einer Kritikalitätsbewertung wählt eine automatisierte Teststeuerung die für den jeweiligen Testfokus relevanten logischen Szenarien aus und erzeugt konkrete Szenarien durch Belegung der Parameterräume mit eindeutigen Werten. Nach Ableitung der konkreten Testfälle wird eine Zuordnung auf virtuelle Testinstanzen, wie Model-, Software- und Hardware-in-the-Loop-Umgebungen (MiL, SiL, HiL) oder physische Tests im Fahrzeug vorgenommen. Im Anschluss an die Testdurchführung werden die abgedeckten Testfall-beziehungsweise Parameterräume bewertet und ihre Kritikalität verifiziert. Die Testergebnisse werden in der Ergebnisdatenbank abgelegt und über eine Testnachsteuerung werden gegebenenfalls weitere konkrete Szenarien erzeugt, um iterativ die Abdeckung aller wichtigen Bereiche des Testfallraums sicherzustellen.



**BILD 1** Szenarienbasierte Testmethodik (© Bertrandt)



**BILD 2** Sechs-Ebenen-Modell gemäß Pegasus (© Bertrandt)

**SCENARIENMODELLIERUNG UND -BEWERTUNG**

Die Qualität der Testergebnisse wird maßgeblich dadurch bestimmt, dass die Szenarien den relevanten Testfallraum abdecken und auf validen Daten basieren, die der Realität auf der Straße gerecht werden. Grundlage dafür sind generische Manöver aus dem alltäglichen Straßenverkehr, aber auch kritische Situationen, die durch Auswertung von Felddaten oder Unfalldatenbanken identifiziert werden. Zum strukturellen Aufbau der Test-szenarien wird ein Sechs-Ebenen-

Modell gemäß Pegasus verwendet [6], das sich aus den Ebenen Straße, Leitinfrastruktur, temporäre Einflüsse, zum Beispiel Baustellen, Dynamik von Ego-Fahrzeug und weiteren Verkehrsteilnehmern, Umweltbedingungen sowie digitale Informationen (Car2X) zusammensetzt, **BILD 2**. Die Informationen aus den unterschiedlichen Ebenen können im Wesentlichen beliebig miteinander kombiniert werden, um eine breite Variation von Testszenarien zu erhalten.

Aufgrund der Vielzahl möglicher Szenarien und Parametervariationen würde eine Absicherung des vollständigen Test-

fallraums exponentiell hohe Aufwände verursachen, die vor dem Hintergrund immer höherer Releasefrequenzen von Softwarelieferungen ausgeschlossen sind. Um die Testdurchführung auf relevante konkrete Szenarien zu beschränken, ist eine Kritikalitätsbewertung erforderlich, die die Auswahl und Wertebelegung der verfügbaren Parameter auf den für die zu testende Funktion interessanten Parameterbereich fokussiert. Dabei kommen sowohl prädiktive Verfahren zum Einsatz, die keine Testdurchführung bedingen, wie auch Analysen, die im Rahmen der Testnachsteuerung den Parameterbereich justieren. Kritikalitätsmetriken wie Time-to-Collision, Time-to-Brake oder Distance-of-Closest-Encounter [7] können für jedes Szenario gesondert definiert und in der Szenariendatenbank abgelegt werden.

Zur Konkretisierung der Testszenarien wurde ein Softwaremodul entwickelt, **BILD 3**. Für die initiale Szenarienerstellung wird die Manöverbeschreibung für die dynamischen Verkehrsteilnehmer mit statischen Informationen aus den anderen Leveln des Sechs-Ebenen-Modells überlagert. Im zweiten Schritt werden die Szenarienparameter miteinander kombiniert und über eine Wahrscheinlichkeitsverteilung die Abstraten für die jeweiligen Parameterbereiche festgelegt. Auf Basis der prädiktiven Kritikalitätsbewertung werden die relevanten, kritischen Szenarien identifiziert und damit der Test der Systemgrenzen fokussiert. Abschließend sorgt eine stochastische Variation der Szenarienparameter dafür, dass im Test auch zufällige Streuungen und Abweichungen berücksichtigt werden, die im realen Straßenverkehr unvermeidlich sind. Im Rahmen der Testnachsteuerung führt eine Kritikalitätsbewertung bereits getesteter Szenarien zu einer Optimierung der dynamischen und statischen Szenarienparameter abhängig davon, wie zuverlässig sich die Prädiktion der Szenarienkritikalität erweist.

## VIRTUELLE TESTTOOLKETTE

Trotz Begrenzung der Absicherung auf die als relevant identifizierten Szenarien, ist eine Teststrategie unerlässlich, die die Testumfänge im Fahrzeug sowie – aufgrund des Rechenzeitbedarfs für die Echtzeitsimulation – in SiL- und HiL-Umgebungen auf das notwendige Minimum reduziert. Vielmehr ist es das Ziel,

die verwendete Toolkette, **BILD 4**, vollständig zu virtualisieren und durch Transfer der Testdurchführung in die Cloud eine Parallelisierung sowie Simulationsgeschwindigkeiten schneller als Echtzeit zu erreichen.

Während die zuvor skizzierten Tools für Szenariendatenbank, Szenarienkonkretisierung, Ergebnisdatenbank und Testnachsteuerung auf dem Client-PC laufen, lassen sich die rechenzeitintensi-

ven Bestandteile der virtuellen Toolkette vollständig in die Cloud integrieren. Im Wesentlichen handelt es sich dabei um die Umgebungssimulation, die Simulation des virtuellen Steuergeräts (vECU) sowie die übergeordnete Testautomatisierung. Dabei deckt die Umgebungssimulation neben der Dynamik des Ego-Fahrzeugs auch die virtuelle Verkehrs-umgebung sowie Modelle für die Umfeldsensorik ab, wie Kamera, Lidar

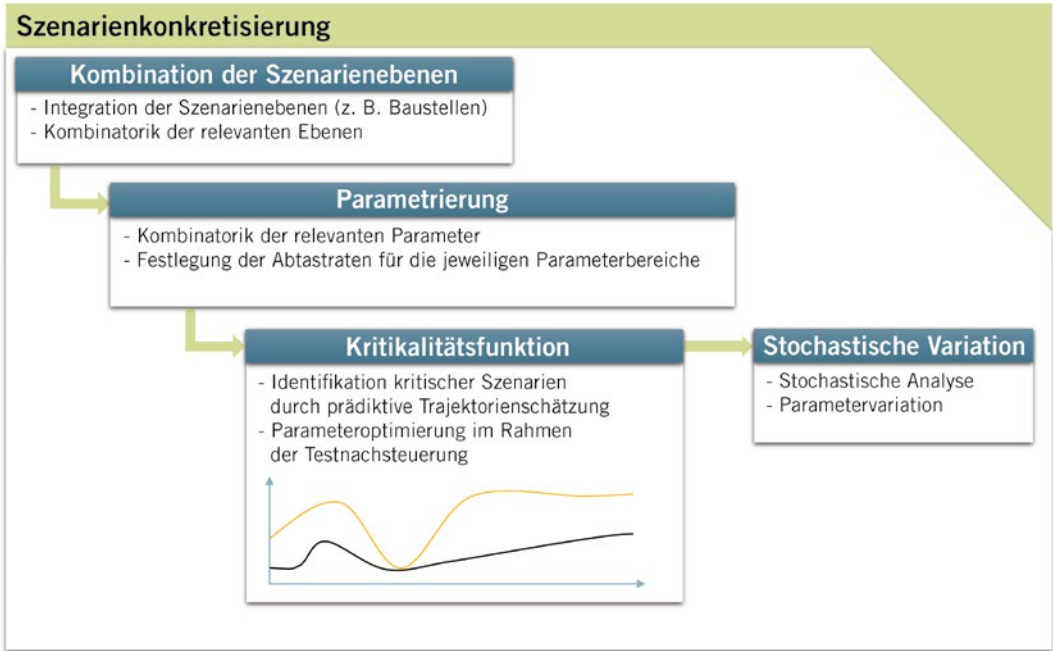


BILD 3 Modul zur Szenarienkonkretisierung (© Bertrandt)

oder Radar, die Informationen über den Fremdverkehr in Form von Punktwolken, Target- oder Objektlisten zur Verfügung stellen. Die zu testende Funktion wird über ein vECU-Tool integriert, das die Erzeugung und Einbindung von einzelnen Regelalgorithmen oder Softwarekomponenten bis hin zu vollständigen Steuergerätearchitekturen mit zugehöriger Bussimulation gestattet. Die programmtechnische Anbindung des vECU-Codes erfolgt häufig über ein Functional Mock-up Interface (FMI). Die Testautomatisierung steuert die Simula-

tion der Testszenarien an und nimmt nach Ende der Testfalldurchführung die szenarienspezifische Auswertung vor.

Im Hinblick auf Leistungsumfang und Wartbarkeit der eingesetzten Tools wurden für die maßgeblichen Komponenten der virtuellen Testtoolkette ausschließlich kommerzielle Produkte eingesetzt. Exemplarisch wurden zwei Varianten implementiert, die sowohl die Pegasus-Methodik wie auch perspektivisch OpenScenario bestmöglich unterstützen. Zum einen findet eine homogene Toolkette bestehend aus den Simulations- und vECU-Tools

ASM, VEOS und System Desk von dSpace Anwendung, zum anderen eine heterogene Kombination von IPG CarMaker und QTronic Silver. Als Testautomatisierung kommt in beiden Fällen ECU-Test von TraceTronic zum Einsatz.

Für die Bereitstellung in die Cloud gibt es mehrere Optionen. Sofern die verwendeten Tools die Nutzung von Docker-Containern unterstützen, ist die Portierung der PC- in die Cloud-Umgebung relativ einfach zu realisieren. Alternativ ist der Betrieb der Toolkette in einer virtuellen Maschine möglich, bei der zusätzlich das Betriebs-

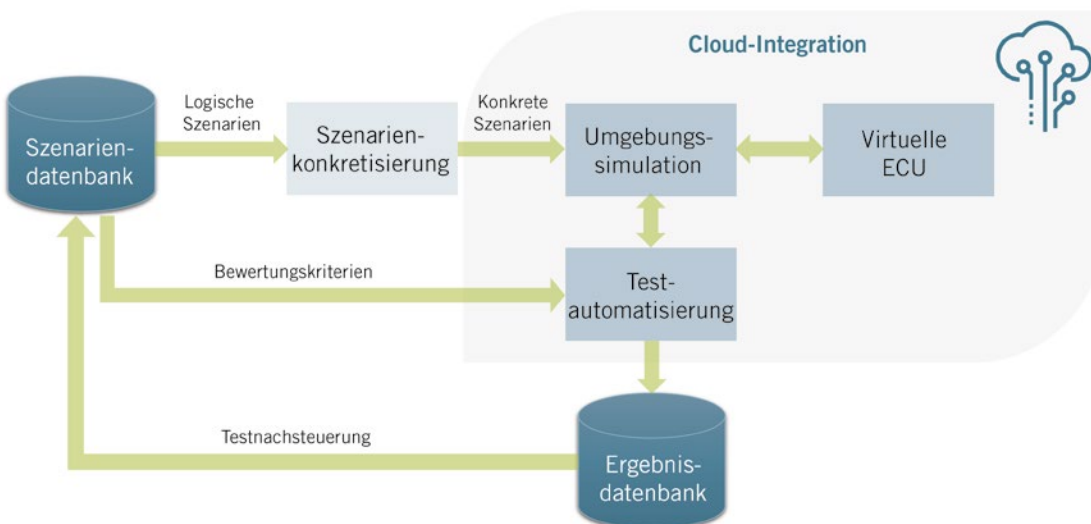


BILD 4 Virtuelle Testtoolkette (© Bertrandt)

system sowie CPU-, GPU- und Speicherressourcen emuliert werden müssen. In beiden Fällen lassen sich die virtuellen Toolketten nahezu beliebig vervielfachen und die Testdurchführung aufgrund der Skalierbarkeit der Rechenressourcen in der Cloud parallelisieren.

## ERGEBNISSE

Im Kontext der von Bertrandt entwickelten Plattform für hochautomatisiertes Fahren HARRI [8] wurde die beschriebene virtuelle Testtoolkette eingesetzt, um autonome Park- und Rangierfunktionen zu testen. Daneben wurde ein Proof-of-Concept für den Test einer Fahrerassistenzfunktion zur Fahrzeuglängsführung auf Autobahnen durchgeführt. Die erstellten generischen Szenarienkataloge sind ebenso für Funktionen höherer Autonomiestufen nutzbar. Im Rahmen von Kundenprojekten finden die Ergebnisse unter anderem Eingang in den Test vorausschauender Sicherheitsfunktionen. Die Testort-unabhängige Szenarienbeschrei-

bung kann dabei auch zur Generierung von Steuergrößen für Fahrroboter aus dem Fahrzeugtest verwendet werden.

Die bisherigen Ergebnisse zeigen, dass die szenarienbasierte Testmethodik gegenüber dem herkömmlichen anforderungsbasierten Test eine höhere Robustheit hinsichtlich Änderungen in der Anforderungslage und dem Entwicklungsprozess aufweist. Trotz der Wiederverwendbarkeit der generischen Testszenerien und der effizienten Auswertung der Testergebnisse sind Aufwand und Testumfang zum Nachweis von korrekter Funktionalität und Sicherheit automatisierter Fahrfunktionen jedoch erheblich. Mit zunehmender Komplexität der Funktion ist dies nur durch Reduktion der Testszenerien auf Basis einer Kritikalitätsbewertung sowie durch konsequente Virtualisierung machbar. Gegenstand aktueller Untersuchungen sind Messbarkeit und Nachweis der Abdeckung aller infrage kommenden Testszenerien sowie die Erweiterung der Testfallerstellung und -auswertung um zusätzliche Intelligenz.

## LITERATURHINWEISE

- [1] Linz, T.: Testen in Scrum-Projekten. Leitfaden für Softwarequalität in der agilen Welt: Aus- und Weiterbildung zum ISTQB Certified Agile Tester – Foundation Extension. Heidelberg: dpunkt.verlag, 2017
- [2] Maurer, M.; Gerdes, J. C.; Lenz, B.; Winner, H. (Hrsg.): Autonomes Fahren: Technische, rechtliche und gesellschaftliche Aspekte. Berlin/Heidelberg: Springer, 2015
- [3] N. N.: Forschungsprojekt Pegasus. Online: <https://www.pegasusprojekt.de/de/about-PEGASUS>, aufgerufen am 31.01.2020
- [4] International Organization for Standardization: ISO/PAS 21448:2019 Road vehicles – Safety of the intended functionality, Genf, 2019
- [5] ASAM e. V.: OpenScenario. Online: <https://www.asam.net/standards/detail/openscenario>, aufgerufen am 31.01.2020
- [6] N. N.: Pegasus Scenario Database. Online: [https://www.pegasusprojekt.de/files/tmpl/Pegasus-Abschlussveranstaltung/15\\_Scenario-Database.pdf](https://www.pegasusprojekt.de/files/tmpl/Pegasus-Abschlussveranstaltung/15_Scenario-Database.pdf), aufgerufen am 31.01.2020
- [7] Junietz, P.; Schneider, J.; Winner, H.: Metrik zur Bewertung der Kritikalität von Verkehrssituationen und -szenarien. 11. Uni-DAS e. V. Workshop Fahrerassistenz und automatisiertes Fahren, Walting, 29.-31.03.2017
- [8] Schiekhofer, P.; Erdogan, Y.; Schindler, S.; Wendl, M.: Maschinelles Lernen für das automatisierte Fahren. In: ATZ 121 (2019), Nr. 12, S. 48-51



### READ THE ENGLISH E-MAGAZINE

Test now for 30 days free of charge:  
[www.atz-worldwide.com](http://www.atz-worldwide.com)