Springer Vieweg

ATZ extra

KI-gestützte Testentwicklung für zuverlässige Fahrzeugsoftware

Cognizant Mobility

© Cognizant Mobility

KI-gestützte Testautomatisierung für Embedded-Elektronik-Systeme

VERFASST VON

Tomasz Witkowski, M. Sc. Eng.
ist Technical Lead E/E System Design & MBSE Competence Lead bei Cognizant Mobility in München.

Dipl.-Inform. Jörg Hofmeyer
ist Head of E/E XiL-Test & SW bei Cognizant Mobility in München.

Dr. Daniel Isemann
ist Head of Data Science & AI bei Cognizant Mobility in München.

Cognizant Mobility gibt einen Praxisbericht über den Einsatz von KI entlang der gesamten Testwertschöpfungskette – von den Anforderungen bis zur Testfallanalyse. Der Beitrag zeigt, wie jeder Schritt dieser Kette durch KI und Automatisierungstechnologien unterstützt werden kann.

Moderne High-End-Fahrzeuge enthalten mehr Recheneinheiten als so manches Weltraumfahrzeug. Sicherzustellen, dass Bordelektronik, Infotainment, ADAS/AD-Funktionen und Konnektivität zuverlässig alle Anforderungen erfüllen, ist anspruchsvoll geworden. Klassische Qualitätssicherung wird daher zunehmend durch modellbasierte Systemtechnik und Testautomatisierung ergänzt, um höhere Testabdeckung bei geringeren Kosten zu erreichen.

MODEL-BASED SYSTEMS ENGINEERING

Model-Based Systems Engineering (MBSE) etablierte sich seit Mitte der 2000er-Jahre mit SysML 1.0 und ge-

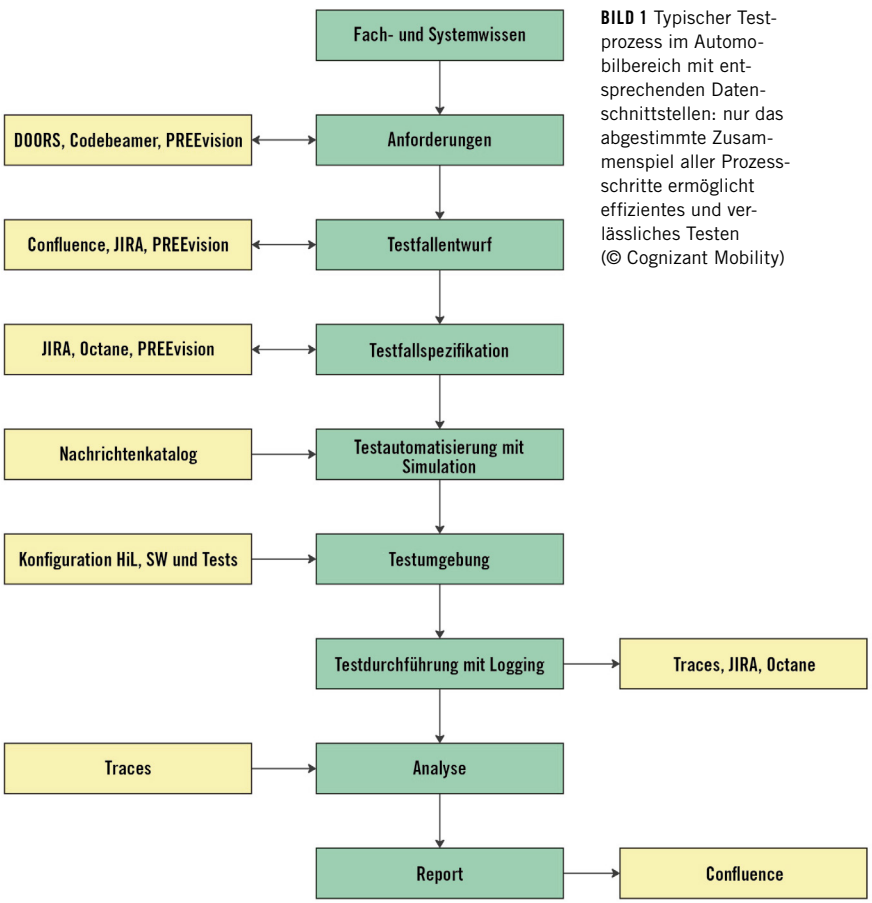


BILD 1 Typischer Testprozess im Automobilbereich mit entsprechenden Datenschnittstellen: nur das abgestimmte Zusammenspiel aller Prozessschritte ermöglicht effizientes und verlässliches Testen (© Cognizant Mobility)

wann durch die MBSE 2020 Vision von Incose zusätzlich an Bedeutung [1, 2]. Für die Automobilentwicklung bietet es ein durchgängiges Systemmodell über das gesamte V-Modell hinweg, verbessert Rückverfolgbarkeit, Konsistenz und Zusammenarbeit und reduziert den Aufwand der Toolintegration. Um die wachsende Systemkomplexität zu beherrschen, nutzen OEMs und Zulieferer vermehrt MBSE. Tools wie Cameo, Enterprise Architect, Rhapsody sowie automobilspezifische Plattformen wie PREEvision, Siemens Capital oder SystemDesk unterstützen diesen Wandel. MBSE ergänzt bis heute klassische Multitool-Umgebungen. Es bietet eine gute Grundlage, den gesamten Testprozess zu automatisieren und KI-gestützt zu optimieren [3].

STATE-OF-THE-ART-TESTEN

Ausgangspunkt eines ganzheitlichen Testprozesses [4] ist ein tiefes Fach- und Systemverständnis, das über die einzelne Komponente hinaus das Gesamtsystem berücksichtigt, denn was isoliert funktioniert, kann in der Wirkkette scheitern, **BILD 1**. Die Testautomatisierung beginnt mit präzisen, testbaren Anforderungen, aus denen ein Fallentwurf mit klaren Vorbedingungen, zu erwartenden Ergebnissen und systematischen Eingabewerten entsteht. Daraus wird die Fallspezifikation mit konkreten Testfällen und -daten abgeleitet. Die Automatisierung erfolgt je nach Testart über Generatoren, Keywordansätze oder vollständige Skripte, wobei Variantenvielfalt und notwendige Restbussimulationen besondere Herausforderungen darstellen. In der Testumgebung werden Skripte, Simulationen und Konfigurationen zusammengeführt und mit den Steuergeräten verbunden; idealerweise wird dieser Schritt über CI/CD automatisiert. Die Durchführung umfasst Ausführung, Monitoring und Umgang mit instabilen Tests, gefolgt von Analyse, Fehlerdokumentation und Reporting. Eine klare Teststrategie und professionelles Releasemanagement schließen den Prozess ab.

Die Erfahrung zeigt, dass KI nahezu jeden Schritt der Testpipeline unterstützen kann. Fach- und Systemwissen lässt sich über Large Language Models (LLMs) oder Bots, die auf interne Quellen wie

PDFs, Word-Dokumente oder Confluence zugreifen, wesentlich schneller erschließen. Zusätzliche Informationen wie Nachrichtenkataloge oder Konfigurationen können effizient über RAG- oder MCP-Anbindungen (JIRA, Confluence) eingebunden werden. Wichtig ist ein klarer Erstellungs- und Reviewzyklus für nachvollziehbare Ergebnisse und einen hohen Qualitätsstandard. Der Reviewschritt harmonisiert bestehende Artefakte. Dieser Beitrag zeigt, wie eine MBSE-Basis systematisch um KI erweitert werden kann.

PREEVISION TRIFFT OLLAMA

Cognizant Mobility nutzt das Requirements-Engineering-Tool PREEvision von Vector. Da dieses keine native GenAI-Integration bietet, wurde eine WebApp entwickelt, **BILD 2**. Um Informationssicherheit der sensiblen Anforderungsdaten zu gewährleisten, werden die für die Inferenz benötigten LLMs auf einem Ollama-Server lokal in einer abgesicherten Rechnerumgebung betrieben.

Nach einer Evaluationsphase mit verschiedenen Open-Weight-Modellen kam ein Modell von Mistral zum Einsatz. Die Architektur ermöglicht einen nahtlosen Wechsel zwischen verschiedenen Open-Weight-Modellen und Anbietern, darunter OpenAI, Anthropic und Google. Die Entwicklung konzentrierte sich auf wirkungsvolle, geringe Komplexitätsanwendungen wie automatisierte Qualitätsprüfung von Anforderungen und KI-unterstützte Testfallgenerierung.

ANFORDERUNGSÜBERPRÜFUNG

Für die Anforderungsüberprüfung wurde ein Few-Shot-Prompting-System implementiert, um eine qualitative Überprüfung der als Eingabe übermittelten Anforderungen durchzuführen. Dem LLM wurden hierbei Rolle, Zweck und Kontext, Qualitätskriterien, Plan zur Durchführung der Überprüfung, Beispiele für korrekt und inkorrekt bewertete Anforderungen sowie das gewünschte Ausgabeformat (JSON) über Prompt-Mechanismen zur Verfügung gestellt. Mit einer Moving-Window-Methode werden alle Anforderungen durchlaufen, wobei je zwei benachbarte davor und danach als Kontext mitgeführt werden.

Das Ergebnis der Überprüfung besteht aus:

- *Overall Quality* – mathematischer Durchschnitt über alle zehn Kriterien
- *Review Statistics* – Einblick in verwendeten Token und der für die Überprüfung eines Ergebnisses benötigten Zeit, um potenzielle Kosten und die Zeit zur Überprüfung von Anforderungssätzen zu verstehen
- *Quality* – Wert zwischen 0 und 5 in jedem der zehn Kriterien für gut formulierte Anforderungen: Klarheit, Testbarkeit, Eindeutigkeit, Atomizität, Konsistenz, Machbarkeit, Überprüfbarkeit und Messbarkeit mit einer kurzen Erklärung des gegebenen Scores, **BILD 3**
- *Issues* – bei Kriterien mit einer Punktzahl von 3 oder weniger werden dem Anwender Verbesserungsvorschläge präsentiert
- *Open Questions* – Rückfragen des Systems, die beim Schreiben der Anforderungen berücksichtigt werden sollten
- *Suggestions* – Vorschläge seitens der KI: einfache Korrekturen von Grammatik, Zeichensetzungen und Tippfehlern oder Neuformulierung der Anforderung unter Beibehaltung der Bedeutung oder Aufteilung der Anforderung in kleinere, atomare Varianten, wenn eine niedrige Bewertung der Atomizität vorliegt, **BILD 4**
- *Tests* – Die KI basierte Erstellung der Testfallspezifikation bildet den Startpunkt der Testautomatisierung, **BILD 5**. Über Few-Shot-Prompts nach einem Wenn-Dann-Muster entsteht eine Spezifikationsliste in natürlicher Sprache.

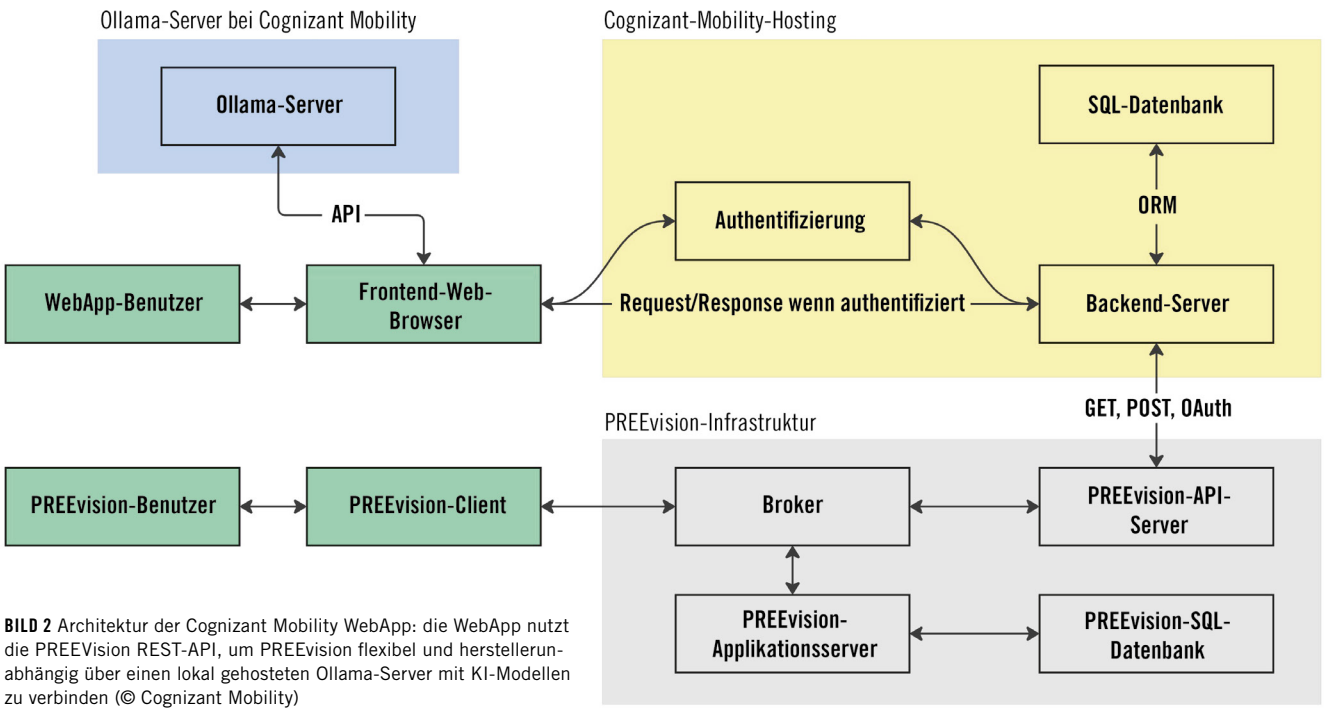


BILD 2 Architektur der Cognizant Mobility WebApp: die WebApp nutzt die PREEVision REST-API, um PREEvision flexibel und herstellerunabhängig über einen lokal gehosteten Ollama-Server mit KI-Modellen zu verbinden (© Cognizant Mobility)

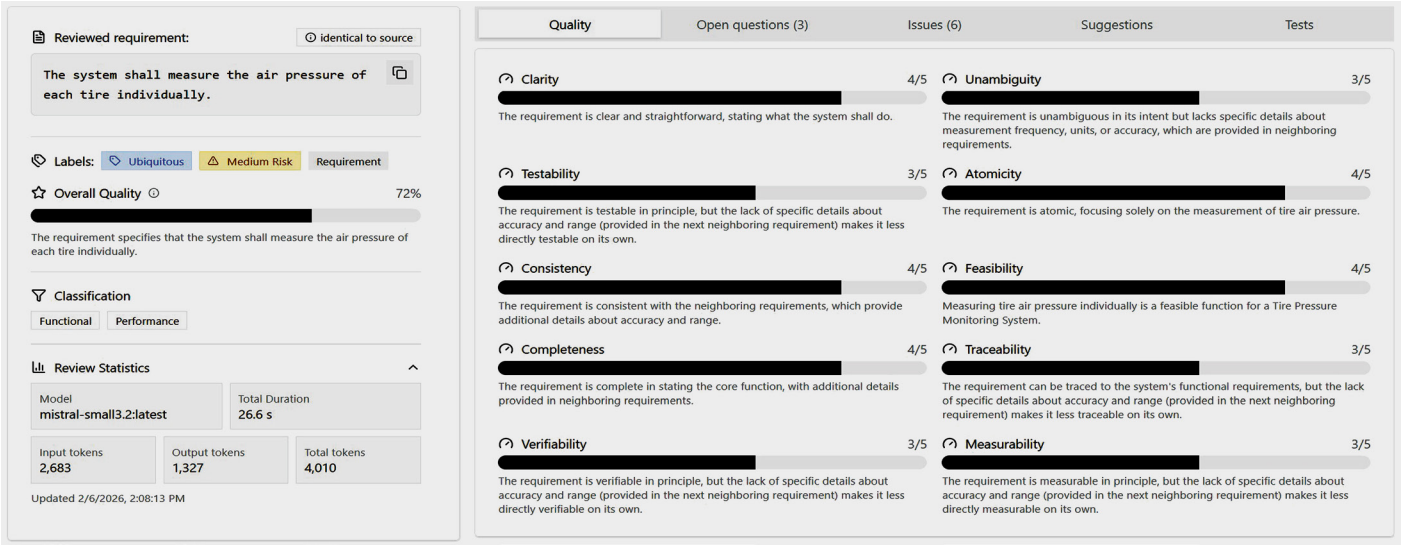


BILD 3 Anforderungsüberprüfung mit KI in der Mobility WebApp – Ergebnisse einer Qualitätsüberprüfung: die Anforderung werden automatisch aus PREEvision übernommen und auf zehn Qualitätskriterien hin überprüft (© Cognizant Mobility)

Daraus generiert die KI zahlreiche positive, negative und oft unerwartete Corner-Case-Tests. Die Ergebnisse müssen sorgfältig plausibilisiert und auf Vollständigkeit geprüft werden.

Notwendige Informationen wie Nachrichtenkataloge, Modelle oder existierende Testskripts werden dem LLM zur Verfügung gestellt. Für die Erstellung der Testautomatisierungsskripte muss der KI der Kontext über Testframework, Bibliotheken, Signale, CAN-IDs, Beispiele oder Coding-Guidelines mitgegeben werden, da sonst nicht-lauffähiger Code erzeugt wird. Hier zeigt sich auch wie gut die Qualität der Vorarbeiten waren. Je höher die Eingangsqualität, umso bessere Skriptergebnisse werden erzielt.

ANALYSE UND REPORTING

Um Systeme wie Hardware-in-the-Loop (HiL) optimal auszulasten, werden automatisierte Testläufe regelmäßig auch außerhalb normaler Arbeitszeiten durchgeführt. In einem früheren ATZelektronik-Beitrag [5] wurde gezeigt, wie solche Dauerläufe robust betrieben und erste KI-Potenziale in der Auswertung identifiziert werden können, damals vor allem mit klassischen, teils unüberwachten Lernverfahren. Auch wenn sich die Methoden weiterentwickelt haben, sind LLMs in diesem Schritt nicht das Hauptwerkzeug.

Ein Kernproblem hochautomatisierter Testfahrten ist die Flut an Fehler-

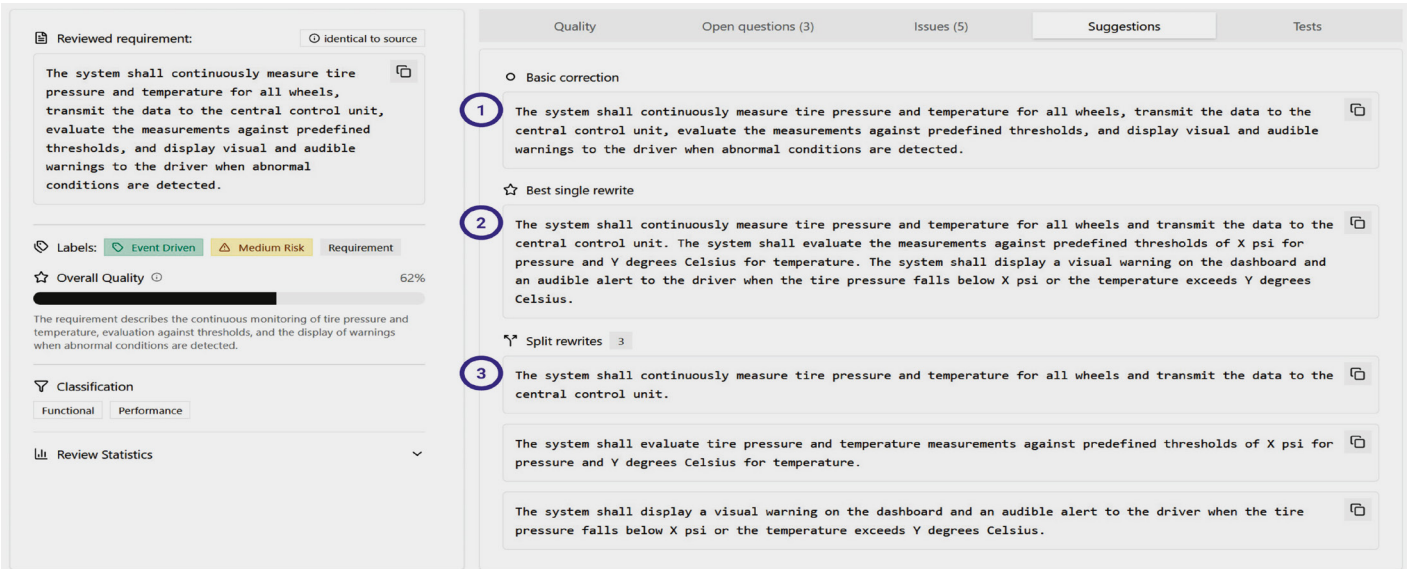


BILD 4 Anforderungspräzisierung mit KI in der Mobility WebApp – Verbesserungsvorschläge; eine nicht-atomare Anforderung wird in drei atomare aufgeteilt (© Cognizant Mobility)

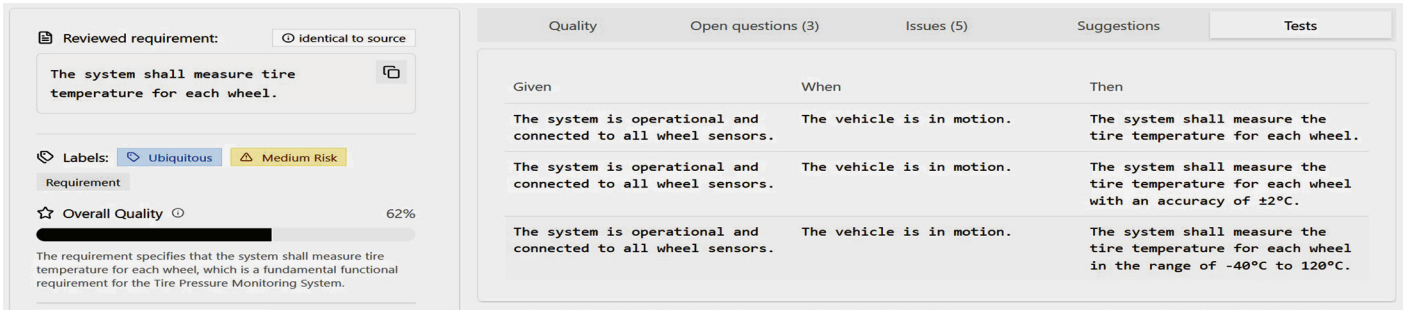


BILD 5 Testfallspezifikation mit KI in der Mobility WebApp – Auf Basis der geprüften und verbesserten Anforderungen werden Spezifikationen für Testfälle automatisch generiert (© Cognizant Mobility)

meldungen, die oft auf wenige, teils bekannte Ursachen zurückgehen. Die entstehenden Ergebnisstrukturen sind komplex strukturierte Listen oder Bäume mit informationsreichen Knoten, auf denen LLMs bisher nur begrenzt effektiv arbeiten. Deshalb setzt Cognizant Mobility auf Graphähnlichkeitsverfahren, um strukturelle Zusammenhänge zu erkennen, und kombiniert diese erst im zweiten Schritt mit der semantischen Analyse der Knotentexte – ein Bereich, in dem LLMs wieder ihre Stärke ausspielen können. Dies hat sich im Praxistest bewährt, **BILD 6**.

ZUSAMMENFASSUNG UND AUSBLICK

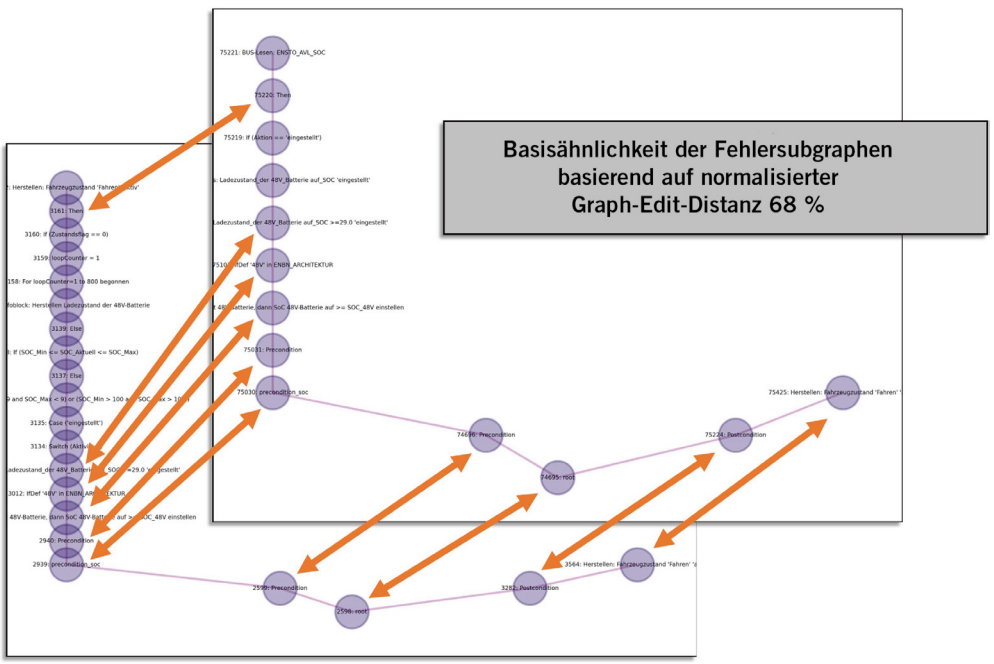
Es wurde gezeigt, welches Potenzial KI heute schon im Rahmen einer klassischen Testpipeline entfalten kann. Besonders stark sind ihre Fähigkeiten bei Formulierungsverbesserungen, Fehlerkorrekturen und der Generierung technischer Artefakte. Gleichzeitig bleiben Qualitätsbewertungen und komplexe Analysen noch unzuverlässig und erfordern menschliche Überprüfung.

Nichtsdestotrotz unterstreicht das Nutzerfeedback das KI-Potenzial. Ihr Einsatz verschiebt dabei die Rolle der Ingenieurinnen und Ingenieure: Statt große Mengen an Testfällen oder Spezifikationen zu erstellen, orchestrieren sie zunehmend digitale Werkzeuge. Die Kernkompetenzen liegen künftig darin, präzise Spezifikationen zu formulieren und KI-Ergebnisse kritisch zu prüfen. KI liefert Geschwindigkeit und Volumen, während Menschen Vision, Kontextverständnis und Qualitätsverantwortung einbringen – insbesondere in sicherheitskritischen Bereichen wie ISO 26262, in denen klare Grenzen zwischen automatisierter Unterstützung und menschlicher Verantwortung unverzichtbar bleiben.

Um KI nachhaltig und sicher einzubinden, müssen Datenschutz, Toolandschaft, Prozessintegration und ROI laufend bewertet werden. Lösungen wie RAG, lokale Modelle oder abgeschottete Firmen-Clouds können dabei helfen, Informationen kontrolliert einzubringen und nachvollziehbare Ergebnisse zu erzeugen. KI wird dabei assistierend zum festen Werkzeug in der Systemtechnik. Cognizant Mobility baut den Einsatz von KI im MBSE und Testumfeld kontinuierlich weiter aus, ohne dabei die Systemqualität zu gefährden. Im Fokus stehen aktuell unter anderem Weiterentwicklungen wie regulatorische Analysen via RAG und Vektordatenbanken oder Spec-Driven Development sowie MBSE-Chatbot-Agenten, die Informationen nach Bedarf abrufen [6] und Systemanalysen, um Widersprüche zwischen Teilsystemen zu identifizieren.

Die nächsten Jahre werden das Embedded-Software-Testing grundlegend verändern. In der IT geht der Trend in Richtung Spec-driven Development und vieles spricht dafür, dass die Embedded-

BILD 6 Automatisierte Analyse von Testfehlern; um den Ähnlichkeitsgrad zwischen Testfehlern zu bestimmen, werden Teilgraphen fehlgeschlagener Testschritte primär anhand eines normalisierten Graphähnlichkeitsmaßes miteinander verglichen (© Cognizant Mobility)



Entwicklung diesem Weg folgen wird. MBSE hat für viele Aspekte seine ersatzlose Berechtigung. Es ist durch funktionale oder nicht-funktionale Anforderungen zu ergänzen, die in eine hochpräzise, konsistente Spezifikation gegossen werden, die dann auch in der Embedded-Entwicklung zum zentralen Artefakt wird, aus dem Software, Tests und Analysen automatisiert abgeleitet werden. KI unterstützt und beschleunigt diese Entwicklung, indem sie Spezifikationen schärft und daraus direkt technische Artefakte erzeugt. Die Zukunft wird zeigen, ob sich dieser Prozess über Jahrzehnte oder nur über wenige Jahre hinziehen wird.

LITERATURHINWEISE

[1] OMG Standards Development Organization (Hg.): About the OMG Systems Modeling Language Specification Version 1.0. Online: www.omg.org/spec/SysML/1.0/, aufgerufen: 3. März 2026

[2] Incose (Hg.): Systems Engineering Vision 2023. Online: https://sdincose.org/wp-content/uploads/2011/12/SEVision2020_20071003_v2_03.pdf, aufgerufen: 2. März 2026

[3] Kriebel, S. et al.: Improving Model-based Testing in Automotive Software Engineering. In ACM/IEEE 40th International Conference on Software Engineering: Software Engineering in Practice, 2018. Online: www.researchgate.net/publication/325728663_Improving_model-based_testing_in_automotive_software_engineering, aufgerufen: 2. März 2026

[4] Austrian Testing Board, German Testing Board e. V., Swiss Testing Board (Hg.): ISTQB Certified Tester Lehrplan Foundation Level 4.0.1a, dt. Ausgabe. Online: www.gtb.de/wp-content/uploads/2024/02/ISTQB_CTFL_Lehrplan-2023_V4.0.1a_DE.pdf, aufgerufen: 2. März 2026

[5] Brandlmeier, A. et al.: KI-basiertes Testing und Robot Process Automation. In: ATZelektronik 16 (2021), Nr. 1, S. 54-56

[6] Zhang, W. et al.: MBSE Co-Pilot: A Research Roadmap. In: Systems Engineering 29 (2025), Nr. 1. Online: https://incose.onlinelibrary.wiley.com/doi/10.1002/sys.70011, aufgerufen: 2. März 2026

IMPRESSUM:
Sonderausgabe 2026 in Kooperation mit Cognizant Mobility GmbH, Ingolstädter Str. 45, 80807 München; Springer Fachmedien Wiesbaden GmbH, Postfach 1546, 65173 Wiesbaden, Amtsgericht Wiesbaden, HRB 9754, USt-IdNr. DE81148419

GESCHÄFTSFÜHRER:
Stefanie Burgmaier | Andreas Funk | Alexandra Dambeck

PROJEKTMANAGEMENT: Anja Trabusch

TITELBILD: © Cognizant Mobility

Cognizant Mobility

Michael Pollner
Dipl.-Inf., MBA
Head of Business Development

M +49 151 44 15 2348
E michael.pollner@cognizant.com