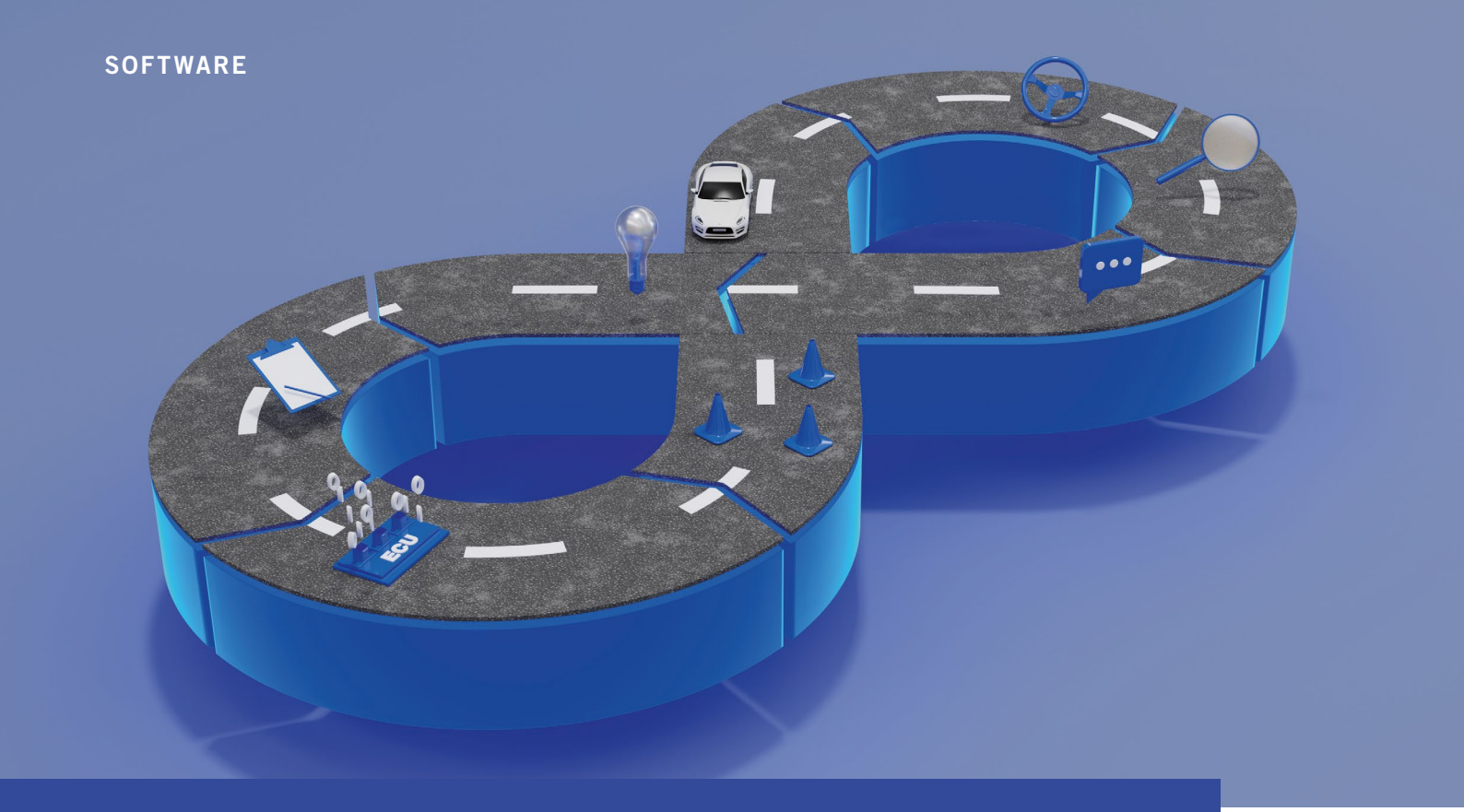

© IPG Automotive

Vollautomatisierte Werkzeugkette für die Entwicklung von Fahrerassistenzfunktionen

Ein wachsender Anteil der Wertschöpfung in der Automobilindustrie wird durch Software auf Steuergeräten generiert. Daher ist es wichtig, in der Softwareentwicklung Methoden einzusetzen, die dem aktuellen Stand der Technik entsprechen. Continental Engineering Services und IPG Automotive erläutern, wie eine moderne Umgebung in Kombination mit dem virtuellen Fahrversuch dazu beiträgt, die Fahrzeugentwicklung effizienter zu gestalten und gleichzeitig eine größtmögliche Testabdeckung sicherzustellen.

Der in der Softwareentwicklung gebräuchliche Prozess der Continuous Integration (CI) beschreibt das fortlaufende Hinzufügen der einzelnen in der Entwicklung stehenden Softwarekomponenten in die Gesamtsoftware. Mit diesem Vorgehen soll sichergestellt werden, dass alle Komponenten fehlerfrei integrierbar sind. Der Prozess des Continuous Testing (CT) hingegen bezeichnet das regelmäßige Testen der Softwarestände in kurzen Iterationszyklen. Beide Prozesse sind eng miteinander verzahnt und bilden die Basis für eine vollautomatisierte Werkzeugkette, die einen effizienten Integrations- und Testprozess gewährleistet. Auf diese Weise wird die Softwareentwicklung durch aktuelle und umfassende Testergebnisse bestmöglich unterstützt.

Notwendig ist diese Unterstützung aufgrund des zunehmenden Automatisierungsgrads von Fahrzeugen. Durch ihn steigt der Testaufwand exponentiell an, da Funktionen immer breitere Anwendungsfelder abdecken müssen und nicht mehr nur unterstützend, sondern in alleiniger Verantwortung fahrzeugführend sind. Da skalierbare Testprozesse während der Entwicklung notwendig sind, werden diese zunehmend digitalisiert und integrierte Softwaremodule mit virtuellen Prototypen getestet. Speziell für Fahrerassistenzsysteme und automatisierte Fahrfunktionen ist dabei nicht nur die realitätsgetreue Abbildung des Fahrzeugprototyps, sondern auch die der statischen und dynamischen Umgebungen relevant.

VERFASST VON

Axel Schneider
ist Manager Simulation & Advanced Tooling bei der Continental Engineering Services GmbH in Frankfurt.

Vanessa Klauer
ist Engineer Virtual Validation bei der Continental Engineering Services GmbH in Frankfurt.

Martin Herrmann
ist Business Development Manager ADAS und automatisiertes Fahren bei der IPG Automotive GmbH in Frankfurt.

Henning Kemper
ist Specialist Editor bei der IPG Automotive GmbH in Karlsruhe.

Der Grund dafür ist, dass die Sicherheit in allen denkbaren Fahrszenarien gewährleistet sein muss. Diese Szenarien werden über die Sensoren des Prototyps erfasst und in die zu prüfende Software eingespeist.

CI/CT-WERKZEUGKETTE

Eine beispielhafte Werkzeugkette für CI/CT-Prozesse ist in **BILD 1** dargestellt. Zu Beginn dieser Kette erstellt ein Mitglied des Entwicklungsteams einen neuen Softwarestand und lädt diesen in ein Source-Code-Management(SCM)-System hoch. Dieses Vorgehen aktiviert die Kette, die den neuen Softwarestand vollautomatisiert analysiert, testet und das Ergebnis übermittelt [1].

Im ersten Schritt werden der derzeitige Entwicklungsstand des Produkts, die Unit Tests und eine Simulationsumgebung generiert. Darauf folgt eine statische Codeanalyse, die die handwerkliche Qualität der Software bewertet, die Einhaltung von Standards überprüft und kritische Stellen im Code aufzeigt. Während des Funktionstests der Software werden sowohl Unit Tests durchgeführt als auch das Gesamtsystem in der Simulationsumgebung geprüft.

Funktionsbestandteile wie die Umgebungserfassung (Perzeption) des Fahrzeugs, die alleine keinen geschlossenen Regelkreis bilden, können im Open-Loop getestet werden. So ist es beispielsweise möglich, in der Realität aufgezeichnete oder synthetisch erzeugte Kamerabilder abzuspielen, die detektierten Objekte mit den dazugehörigen perfekten Annotationen (Ground Truth) abzugleichen und die Performanz zu beurteilen.

Die Basis für Closed-Loop-Tests von Regelfunktionen bilden vorab definierte Szenarienkataloge sowie auf den jeweiligen Funktionstest spezialisierte Bewertungskriterien. Nachdem sich der virtuelle Prototyp gemäß seiner Fahraufgabe frei durch die Verkehrsszenarien bewegen konnte, kann etwa überprüft werden, ob Sicherheitsabstände und Verkehrsregeln eingehalten wurden oder ob es zu Kollisionen kam.

Nach Aufzeichnung und Evaluierung der Tests wird ein Report generiert, der die Ergebnisse zielgerecht darstellt. Schließlich erfolgt die Rückmeldung der Informationen an das Teammitglied, das so ein direktes objektives Feedback zum Softwarestand erhält. Damit ist CI/CT kein linearer Prozess mit Start- und Endpunkt, sondern ein geschlossener Kreislauf, der wiederholt durchlaufen wird und die Ergebnisse vorheriger Durchläufe zur Weiterentwicklung der Software nutzt.

Für die konkrete Anwendung im Entwicklungsprozess wird CarMaker als offene Integrations- und Testplattform eingesetzt. Mit ihr werden der Assistenzfunktion valide fahrzeugdynamische Signale sowie relevante Informationen über das Fahrzeugumfeld bereitgestellt und Testszenarien generiert. Auf diese Weise wird eine große Anzahl von Closed-Loop-Tests des Fahrerassistenzsystems und damit ein durchgängiger Test der Software ermöglicht.

WAHL DER PASSENDEN TESTSTRATEGIE

Bei der Generierung der Testszenarien entsteht ein Zielkonflikt: Die Notwendigkeit, viele verschiedene Testfälle zu generieren, um die Robustheit der Funktion und die Testabdeckung zu erhöhen, geht mit einem hohen Wartungsaufwand einher – die Testszenarien müssen kontinuierlich an veränderte Rahmenbedingungen angepasst werden. Beispielsweise können durch verändertes Verhalten der Funktion aufgrund der laufenden Weiterentwicklung einzelne Manöver zu früh oder zu spät ausgeführt werden, wodurch der Testzweck möglicherweise nicht mehr erfüllt wird.

Eine mögliche Lösung für diesen Konflikt ist die Erstellung von Basisszenarien („logisches Szenario“), die an den relevanten Stellen Parameter vorhalten. So kann mit der Variation verschiedener Parameter eine große Anzahl an Parametersätzen generiert werden, **BILD 2**. Mit wenigen Dutzend detailliert erstellten Basisszenarien, die sich noch sehr gut an veränderte Rahmenbedingungen anpassen lassen, können so automatisiert zehntausende Testfälle erstellt werden. Werden diese Testfälle nun den Software-

BILD 1 Die CI/CT-Werkzeugkette für die Entwicklung von Fahrerassistenzsystemen (© Continental Engineering Services, IPG Automotive)

BILD 2 Mehrere Variationen eines Basisszenarios (© IPG Automotive)

anforderungen zugeordnet, können sie gezielt getestet werden – ein Überblick über den Reifegrad der Software ist so jederzeit möglich.

Bei einer automatisierten Testfallgenerierung und -evaluation besteht zudem stets das Risiko eines falsch positiven Tests und einer späten Fehleridentifikation. Dieses Risiko kann minimiert werden, etwa durch robustere Basisszenarien oder durch das bewusste Testen einer nicht funktionsfähigen Softwarekomponente. Wurde beispielsweise ein Testfall erfüllt, obwohl das Fahrerassistenzsystem nicht funktioniert, besteht Handlungsbedarf. Dieses Risiko lässt sich nicht eliminieren, jedoch durch eine gute Teststrategie minimieren. Ein gewisses Risiko ist allerdings unumgänglich, da durch die Anzahl an Testfällen – vor allem in Hinblick auf Fahrfunktionen des SAE-Level 3 – die manuelle Erstellung und Wartung aller Testfälle zu aufwendig ist.

VORAUSSETZUNGEN ZUR AKZEPTANZ

Ob die Einführung einer solchen Werkzeugkette für den Entwicklungsprozess einen Mehrwert bietet, hängt von mehreren Faktoren ab. Als Hauptkriterien gelten die bewusste Wahrnehmung der CI/CT-Werkzeugkette als Hilfsmittel sowie die Einführung darauf basierender Prozesse. Damit die Kette als ein solches Hilfsmittel verstanden wird, sind mehrere Aspekte relevant.

Zunächst muss, nachdem ein neuer Softwarestand hochgeladen wurde, eine zeitnahe Rückmeldung der Ergebnisse erfolgen. Auf diese Weise kann das Teammitglied fokussiert bleiben und nach einer kurzen Unterbrechung die Entwicklung aufbauend auf den Ergebnissen fortsetzen. Darüber hinaus müssen die verschiedenen Schritte transparent gestaltet sein, sodass nachvollziehbar ist, wie die Ergebnisse zustande kommen. Im Optimalfall sollten alle Schritte auf dem Rechner des Teammitglieds reproduziert werden können.

Dies betrifft hauptsächlich Anforderungen an die Infrastruktur und die Teststrategie, da diese vor allem mit den zur Verfügung stehenden Ressourcen und dem Testumfang zusammenhängen.

Hier bietet sich die Nutzung einer cloudbasierten Infrastruktur an [2]. Sie lässt sich sehr gut an die aktuelle Arbeitslast anpassen und deckt Lastspitzen durch das Hinzubuchen zusätzlicher Rechenkapazität ab. Die Teststrategie sollte berücksichtigen, dass ein Test des neuen Softwarestands nicht immer auf Basis des gesamten Testkatalogs erfolgen muss. Gegebenenfalls genügen ausgewählte, relevante Tests.

Eine weitere Grundvoraussetzung ist, dass einzelne Softwarebestandteile unabhängig voneinander oder gemeinsam (integriert) getestet werden können. CarMaker verfügt dafür über vielfältige Schnittstellen zur Integration einzelner Module oder größerer Systeme. Nicht integrierte Softwarekomponenten kön-

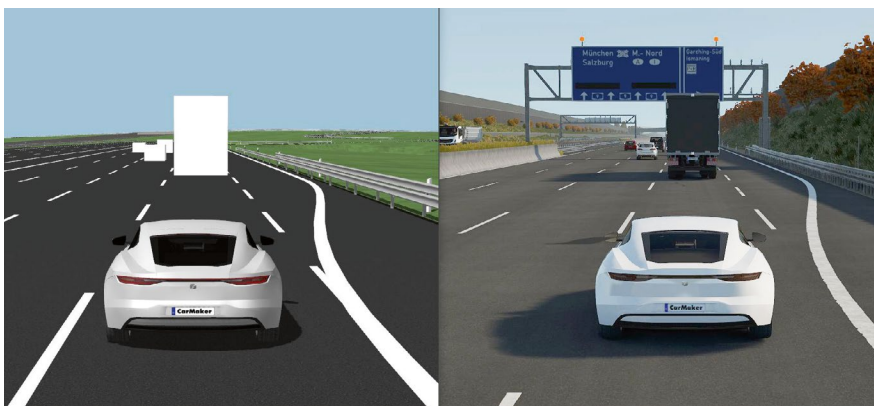

BILD 3 Detailgrad eines Szenarios für Anwendung ohne Perzeptionsmodul (links) und mit (rechts) (© IPG Automotive)

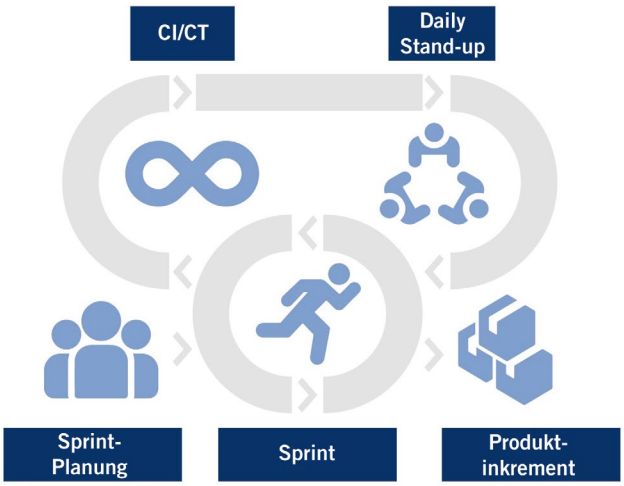

BILD 4 Ablauf des Scrum-Prozesses unter Einbeziehung der CI/CT-Werkzeugkette (© Continental Engineering Services, IPG Automotive)

nen in der Regel durch Ersatzmodelle in CarMaker abgebildet werden, die durch realistische Abbildung einzelner Phänomene dennoch Realitätsnähe ermöglichen. Alternativ können Idealisierung und Reduktion der Komplexität das Auffinden von Fehlerquellen oder auch die Generierung von Testszenarien deutlich vereinfachen.

Ein Beispiel dafür bilden Perzeptionstests: Sobald reale Perzeptionsmodule im virtuellen Gesamtfahrzeug eingebunden werden, **BILD 3**, steigen die Anforderungen an das Umgebungsmodell enorm. Für ein realistisches Sensormodellverhalten auf Basis von Rendering- oder Raytracing-Techniken sind entsprechend detaillierte 3-D-Modelle der Umgebung inklusive der für die Sensortechnik relevanten Materialeigenschaften nötig. Neben einem höheren Aufwand für die Szenarienerstellung steigen damit auch die Ressourcenanforderungen an die verwendete Cloudplattform: Während Central-Processing-Unit(CPU)-basierte Instanzen für viele Testarten ausreichen, sind für physikalische Sensormodelle auch Graphics-Processing-Unit(GPU)-basierte Instanzen nötig. Im Sinne der Effizienz ist also sicherzustellen, dass die Anteile ohne Perzeption nach Möglichkeit in aussagekräftigen Tests separat untersucht werden und die Anzahl der Integrationstests inklusive Perzeption so weit wie möglich minimiert wird.

BASIS FÜR AGILE ENTWICKLUNGSPROZESSE

Basierend auf den Schritten, die ein neuer Softwarestand in der CI/CT-Werkzeugkette durchläuft, kann ein Prozess zur Sicherstellung der Codequalität eingeführt werden. Viele Source-Code-Managementsysteme verfügen über eine solche Schnittstelle. Sie ermöglicht es, sogenannte Quality Gates zur Überführung neuer Softwarestände auf dem Hauptentwicklungspfad werkzeuggestützt einzurichten, indem hierfür ein selbst gewähltes Qualitätslevel und ein Review vorausgesetzt werden. Das Teammitglied muss nicht nur die formalen Bedingungen erfüllen, sondern als Abschluss seiner Implementierung anderen Teammitgliedern seine Arbeit erläutern. Basierend auf den objektiven Bewertungskriterien der statischen Codeanalyse kann eine sachliche Diskussion stattfinden.

Die Verwendung der CI/CT-Werkzeugkette bildet eine gute Ergänzung für ein agiles Entwicklungsprojekt. Folgender Ablauf ist beispielsweise denkbar: Am Ende eines Arbeitstags, nachdem Mitarbeitende weltweit ihre Codeänderungen eingepflegt haben, wird nachts zeitgesteuert der aktuelle Softwarestand gegen den gesamten Testkatalog getestet, **BILD 4**. Beim darauffolgenden sogenannten Daily Standup kann tagesaktuell der Entwicklungsstand des Fahrerassistenzsystems dargelegt und mit dem zu erreichenden Reifegrad am Ende des Sprints abgeglichen werden. Die so geschaffene Transparenz bildet die Grundlage für Vertrauen zwischen den Teammitgliedern und gegenüber der produktverantwortlichen Person. Etwaige Unsicherheiten können frühzeitig erkannt werden. Nicht zuletzt sollten auch alle unternehmensintern entwickelten Werkzeuge mithilfe einer CI/CT-Werkzeugkette entwickelt werden.

Darüber hinaus sind die Auswirkungen auf die Arbeitskultur, initiiert durch die Einführung eines CI/CT-Prozesses, nicht zu unterschätzen. Langfristig kann so der gemeinsame Arbeitsmodus des Teams positiv beeinflusst werden. Bisher wird häufig ein Ticketsystem eingesetzt. Dabei arbeitet jedes Teammitglied seine Tickets allein ab. Da im Review-Prozess andere Teammitglieder hinzugezogen werden müssen, um den entwickelten Code gemeinsam zu überarbeiten, kann ein Gemeinschaftsgefühl entstehen. Darauf basierend tauschen sich die Teammitglieder über anstehende Aufgaben aus und legen so einen Grundstein für einen kooperativen Arbeitsmodus. Dies wirkt sich in der Regel positiv auf den persönlichen Arbeitseinsatz und die Mitarbeiterzufriedenheit aus.

ZUSAMMENFASSUNG

Die entwickelte Umgebung verwendet CarMaker als offene Integrations- und Testplattform und ermöglicht es, verschiedene Testschritte durchzuführen – beispielsweise eine statische Codeanalyse, Build- und Integrationstests, Unit Tests und sowie funktionale Tests. So wird eine höhere Testabdeckung erzielt. Darüber hinaus werden Fehler vermieden beziehungsweise früher identifiziert. Schlussendlich wird ein höherer Reifegrad in früheren Softwareversionen erzielt. Da eine breiter und größer angelegte Teststrategie die handwerkliche Qualität auf Code-Ebene erhöht, führt dies wiederum automatisch zu einer schneller und besser entwickelten Software.

LITERATURHINWEISE

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

[2] Herrmann, M.: Simulation vor Ort und cloudbasiert. In: ATZ 124 (2022), Nr. 10, S. 42-46