



Axel Schneider, Continental Engineering Services GmbH

Einsatz einer modernen Software-Entwicklungsumgebung zur Erzielung einer maximalen Testabdeckung

Für dieses Interview waren wir im Gespräch mit Axel Schneider von der Continental Engineering Services GmbH in Frankfurt. Als Senior Engineer für Virtual Testing & Simulation ist er im Business Center Driver Assistance and Autonomous Driving tätig. Im Gespräch erläutert er, wie eine moderne Entwicklungsumgebung in Kombination mit dem virtuellen Fahrversuch dazu beiträgt, die Fahrzeugentwicklung im Hinblick auf immer komplexere Software-Funktionen effizienter zu gestalten und gleichzeitig eine größtmögliche Testabdeckung zu gewährleisten.

Es freut uns sehr, dass Sie sich Zeit für dieses Gespräch nehmen. Könnten Sie Ihr Tätigkeitsfeld bitte kurz vorstellen und darauf eingehen, welche Ziele Ihre Abteilung verfolgt?

Schneider: Sehr gerne. Persönlich habe ich mich in den vergangenen Jahren vor allem mit dem Aufbau von SIL-Testsystemen und der Integration von verschiedenen Serien-Algorithmen beschäftigt. Zentral waren dabei die Themen Continuous Testing, Automatisierung und Cloud-Simulation.

Unsere Abteilung bietet externen und konzerninternen Kunden Lösungen für Tests mit virtuellem Anteil an. Wir bauen Test Benches mit verschiedenen Integrationsstufen auf, also von SIL über HIL bis hin zu VIL. In vielen Projekten führen wir auch Testaufgaben durch – das sind vor allem Projekte im Bereich Fahrerassistenzsysteme und Bremssysteme. Wir sind aber auch für Simulationsstudien zuständig und unterstützen konzerninterne Projekte durch unsere Expertise im Bereich Teststrategien, aber alles mit virtuellem Anteil.

Der Software-Anteil in Fahrzeugen ist in den vergangenen Jahren rapide angestiegen. Auf der Apply & Innovate 2022 haben Sie ein Projekt vorgestellt, das Continental Engineering Services im Hinblick auf die daraus resultierenden Herausforderungen umgesetzt hat. Welches Ziel wurde dabei verfolgt?

Schneider: Richtig, bei der Apply & Innovate habe ich eine von uns entwickelte CI/CT/CD-Umgebung vorgestellt. Das ist die Kurzform von Continuous Integration, Testing und Delivery: einer automatisierten, im Software-Bereich etablierten Entwicklungsmethode.

Die von uns entwickelte Umgebung nutzt unter anderem CarMaker als offene Integrations- und Testplattform und ermöglicht es, verschiedene Testschritte durchzuführen. Diese Schritte bestehen beispielsweise aus einer statischen Code-Analyse, Build- und Integrationstests, Unit-Tests und nicht zuletzt auch

funktionalen Tests. Das Ziel dabei ist, eine hohe Code-Qualität zu erreichen und systematisch die Serien-Algorithmus-Komponenten zu testen.

Das führt dann zu einer höheren Testabdeckung und der Vermeidung von Bugs. Oder man entdeckt entstandene Bugs viel früher, was zu einem höheren Reifegrad in frühen Softwareversionen führt. Die handwerkliche Qualität auf Code-Ebene wird also durch eine breiter und größer angelegte Teststrategie deutlich erhöht – das führt dann automatisch zu einer schneller und besser entwickelten Software.

Um eine höhere Qualität zu erreichen, existieren in der Softwareentwicklung bereits vielfältige etablierte Prozesse. Lassen sich diese auf die Fahrzeugentwicklung adaptieren?

Schneider: Definitiv, die Methoden und Prozesse, die bei der Entwicklung jeder anderen Software verwendet werden, können zu einem gewissen Grad auch bei der Entwicklung von Software im Automotive-Bereich angewendet werden. Bei der Entwicklung von Open-Source-Software hat jede Programmiersprache im Grunde ihr eigenes Ökosystem mit Entwicklungsbegleitenden Tools, zum Beispiel für die bereits erwähnte statische Code-Analyse oder für funktionale Tests. Mit diesen Tools werden leistungsfähige Open-Source-Libraries entwickelt, die in Milliarden von Geräten integriert sind.

Solche erfolgreichen Beispiele kann man als Blaupause nutzen, um die eigenen Methoden und Prozesse daran anzulehnen. CI/CT/CD ist schon lange fester Bestandteil von Open-Source-Software-Entwicklungsprozessen und unterstützt die Entwickler vor allem durch die Automatisierung von repetitiven Aufgaben wie der Code-Analyse und den Software-Tests.

Welche Kriterien sind Ihrer Meinung nach besonders wichtig, um ein solches Projekt erfolgreich umsetzen zu können?

Schneider: Grundsätzlich muss man erst mal sagen, dass jedes Projekt oder Tool unterschiedliche Anforderungen und Rahmenbedingungen hat. Es gibt hier also keine Vorgehensweise nach Schema F. Aber es gibt meiner Meinung nach Erfolgsfaktoren beim Design und den grundlegenden Kriterien.

Das wichtigste ist, dass die Toolkette von den Nutzern als Hilfsmittel akzeptiert wird. Ihnen muss bewusst sein, dass sie nicht zur Überwachung gedacht ist. Unsere Erfahrungen haben gezeigt, dass nur die Tools, die wirklich akzeptiert sind und als Hilfsmittel wahrgenommen werden, auch tatsächlich aktiv verwendet werden. Wichtig ist auch der Zeitraum, wie lange ein Entwickler auf die Ergebnisse wartet. Er muss relativ schnell erfahren, wie die Analyse seines neu entwickelten Codes verlaufen ist und ob die Tests erfolgreich durchgeführt wurden. Es geht hier darum, die Zeitspanne so kurz zu halten, dass der Entwickler im Fokus bleibt.

Können Sie definieren, welche Kriterien bei einer solchen Code-Analyse ausschlaggebend sind?

Schneider: Natürlich. Bei diesem Prozess sind die Transparenz sowie die Reproduzierbarkeit der Ergebnisse entscheidend. Wenn nicht genau bekannt ist, auf welchen Kriterien basierend der Code analysiert wird, kann sich das schnell demotivierend auswirken. Dem Entwickler muss es zudem möglich sein, alle Schritte, die automatisiert sind, einzeln auf seinem Rechner zu reproduzieren.

Persönlich finde ich es auch enorm wichtig, dass wirklich alle eigenen Tools in diesem Kontext mithilfe einer CI/CT-Toolkette entwickelt werden – also nicht nur das Endprodukt, das später dem Kunden geschickt wird. Die eigenen Tools, die die Entwickler bei ihrer Arbeit unterstützen, sollten ebenfalls eine hohe Qualität aufweisen.

Es existieren auch zusätzliche, nicht messbare Kriterien, die dazu einen Beitrag leisten. Beispielsweise hat in unserer Abteilung die Einführung



Der Interviewpartner Axel Schneider im Gespräch mit Henning Kemper von IPG Automotive

einer solchen Toolkette auch die Arbeitskultur ein Stück weit verändert. Dadurch, dass es ganz objektive Kriterien für die Bewertung des entwickelten Codes gibt, hat das die Möglichkeit für Diskussionen eröffnet. Wir haben Review-Prozesse erarbeitet, die einen kooperativen Austausch voraussetzen. Somit sind immer mehrere Mitarbeiter an einer Implementierung beteiligt.

Erhöhen objektive und messbare Kriterien Ihrer Erfahrung nach das Vertrauen in Software-Entwicklungsprozesse? Vielleicht sogar insbesondere in der Automobil- und Zuliefererindustrie, die tendenziell einen weniger ausgeprägten Software-Background hat?

Schneider: Definitiv. Die Automobilindustrie war bis vor einiger Zeit einfach keine Branche, in der viel Software entwickelt wurde. Früher bestand das Geschäftsmodell daraus, Fahrzeuge zu produzieren und zu verkaufen – die Software war immer eine Art Beiwerk, das man notwendigerweise mitliefern musste. Aber der Automotive-Sektor ist schon seit Jahren im Umbruch und die Software nimmt einen immer größeren Anteil der Wertschöpfung ein.

Das heißt, die Automobilhersteller und Zulieferer wandeln sich immer mehr zu Softwarefirmen – und in solchen Firmen führt kein Weg daran vorbei, mit der Zeit zu gehen und moderne Entwicklungsprozesse anzuwenden.

Welche Rolle spielt CarMaker in der automatisierten CI/CT/CD-Prozesskette?

Schneider: Da wir Software für reale Fahrzeuge entwickeln, benötigen wir natürlich ein virtuelles Gegenstück, in das wir die zu entwickelnde Fahrfunktion integrieren können – das ist der virtuelle Prototyp. CarMaker bildet

da als offene Integrations- und Testplattform einen zentralen Baustein, ohne den wir keine Möglichkeit hätten, unsere Software mit validen Daten zu versorgen. Funktionale Tests der Fahrfunktion wären in einer rein realen Umgebung nur sehr schwer oder in manchen Fällen überhaupt nicht durchführbar. Bei CarMaker kommen uns vor allem die hohe Modularität und die hohe Parametrierbarkeit zugute.

Können Sie bitte noch näher darauf eingehen, welche Maßnahmen Sie durchführen, um eine maximale Testabdeckung zu erzielen?

Schneider: Ja, natürlich – das ist im Prinzip recht simpel. In unserem Prozess haben wir die zu erfüllenden Anforderungen mit einer vordefinierten Anzahl von Testfällen verknüpft. Beim Design der Tests sind die Fahrzeugparametrierung und das Test-szenario variabel gestaltet und können an vordefinierten Stellen über einen Parametersatz verändert werden. Mithilfe von Parametervariationen können wir dann ein einzelnes Szenario mit einem einzigen Fahrzeugmodell skaliert in einer extrem großen Anzahl von Testfällen einsetzen. Wenn alle Tests erfolgreich durchgeführt wurden, ist die Anforderung erfüllt.

Diese Methode hat sich für uns bewährt. In einem Projekt haben wir es so geschafft, aus 24 erstellten Manövern 50.000 verschiedene Tests zu generieren. So müssen wir nur wenige Manöver aktiv warten und die Komplexität bleibt noch in

einem überschaubaren Rahmen.

Wie kann man sich diese Parametervariationen vorstellen: Werden dafür immer wieder minimale Parameteranpassungen vorgenommen und auf dieser Basis entsprechend viele verschiedene Tests generiert?

Schneider: Exakt. Wir können uns beispielsweise einen Testfall vor Augen führen, in dem die Detektion von anderen Verkehrsteilnehmern rund um das eigene Fahrzeug getestet werden muss. Allein durch minimale Anpassungen der Bewegungen dieser Verkehrsteilnehmer entstehen schnell enorme Mengen möglicher Szenarien, die allesamt abgesichert werden müssen. So ist die Anzahl von 50.000 Tests recht schnell erreicht.

Das sind beeindruckende Zahlen. Wie stellen Sie die nötige Rechenleistung für eine so große Testanzahl bereit, um den Zeitaufwand im Rahmen zu halten?

Schneider: Was die technische Umsetzung betrifft, haben wir uns für eine Skalierung der CarMaker-Simulation in der Cloud entschieden. Dabei können wir beliebig viele Ressourcen dazu buchen und auch spontan wieder abwählen – das ermöglicht es uns, sogar hohe Lastspitzen kostensparend abzudecken.

Können Sie bereits abschätzen, wie sich die Thematik Ihrer Meinung nach in näherer Zukunft entwickeln wird?

Schneider: Meine Meinung deckt sich hier mit der Einschätzung von Experten, mit denen ich mich über das Thema virtuelles Testen unterhalte: Virtuelle Methoden sind stark auf dem Vormarsch. Gerade wenn man auf Fahrfunktionen der SAE-Level 3 schaut, also automatisierte Fahrfunktionen, werden diese größtenteils virtuell freigegeben und nur noch vereinzelte Tests im realen Fahrzeug durchgeführt. Auch die Anzahl der zu erwartenden Tests lässt eigentlich keinen anderen Schluss zu.

Vielen Dank, dass Sie sich die Zeit für dieses aufschlussreiche Gespräch genommen haben.