Anforderungen für erfolgreiche Projekte

Software-Industrialisierung als nächster Schritt zu autonom fahrenden Nutzfahrzeugen

Die Entwicklung sicherer und robuster Software ist der Schlüsselfaktor für autonome Nutzfahrzeuge auf ihrem Weg vom Prototyp in die Produktion. Dabei gilt es sieben Schlüsselfaktoren zu berücksichtigen.

Autorin: Dr. Manaswini Rath

Bild 1: Diese Faktoren gilt es bei der Software-Industrialisierung zu berücksichtigen.

Warum ist die Software-Industrialisierung die zentrale Herausforderung für AD-Produktionsprogramme der Stufen 2 und 4, und wie lässt sie sich umsetzen? In den letzten Jahren gab es großes Interesse für autonomes Fahren (AD) in der Nutzfahrzeugbranche. ADAS und autonome Fahrtechnologien (AD) von Nutzfahrzeugen (CVs) haben das Potential, die Kosten zu verbessern und die Sicherheit zu erhöhen. Diese Business Cases für autonome Nutzfahrzeuge wurden bereits bewiesen. Die CV-Branche zielt jetzt auf Modelle mit Funktionen wie Hub-to-Hub-Fahren, Highway Pilot mit Level-2- und späteren Level 4-Funktionen für Produktionsprogramme ab. Um die Markteinführung autonomer Nutzfahrzeuge (CV, Commercial Vehicle) zu beschleunigen, arbeiten CV-OEMs mit AD-Plattformanbietern wie Torc Robotics, Tu Simple, Aurora, Gatik und Anderen zusammen. Im Rahmen dieser Zusammenarbeit beziehen die OEMs die Software und in manchen Fällen die Hardwareplattform von dessen Plattformanbietern. Die Kontrolle über die Gesamtsystemintegration sowie die Einhaltung von ISO 26262 und der Homologationsanforderungen müssen jedoch bei den OEMs liegen.

Durch das Fehlen von Erfahrungen mit Nutzfahrzeugherstellern für autonome Software und die mangelnden Erfahrungen mit Plattformanbietern bei der Entwicklung von Produktionssoftware und -hardware für sicherheitskritische Systeme entsteht jedoch eine Lücke.

In einem Financial-Times-Interview im Januar 2021 betonte der damalige CEO von Waymo die Tatsache, dass die Umstellung

Eck-DATEN

Um sicherheitskritische ADAS- und AD-Software reif für die Produktion zu machen, sind ein fundiertes Fachwissen und viel Erfahrung notwendig, so dass im Nutzfahrzeug-Ökosystem Bedarf für die Zusammenarbeit mit einem passenden „Industrialisierungspartner" besteht, der über entsprechende Kompetenz verfügt. Dieser Beitrag stellt Methoden und Best Practices der Software-Industrialisierung für Sicherheitssysteme vor.

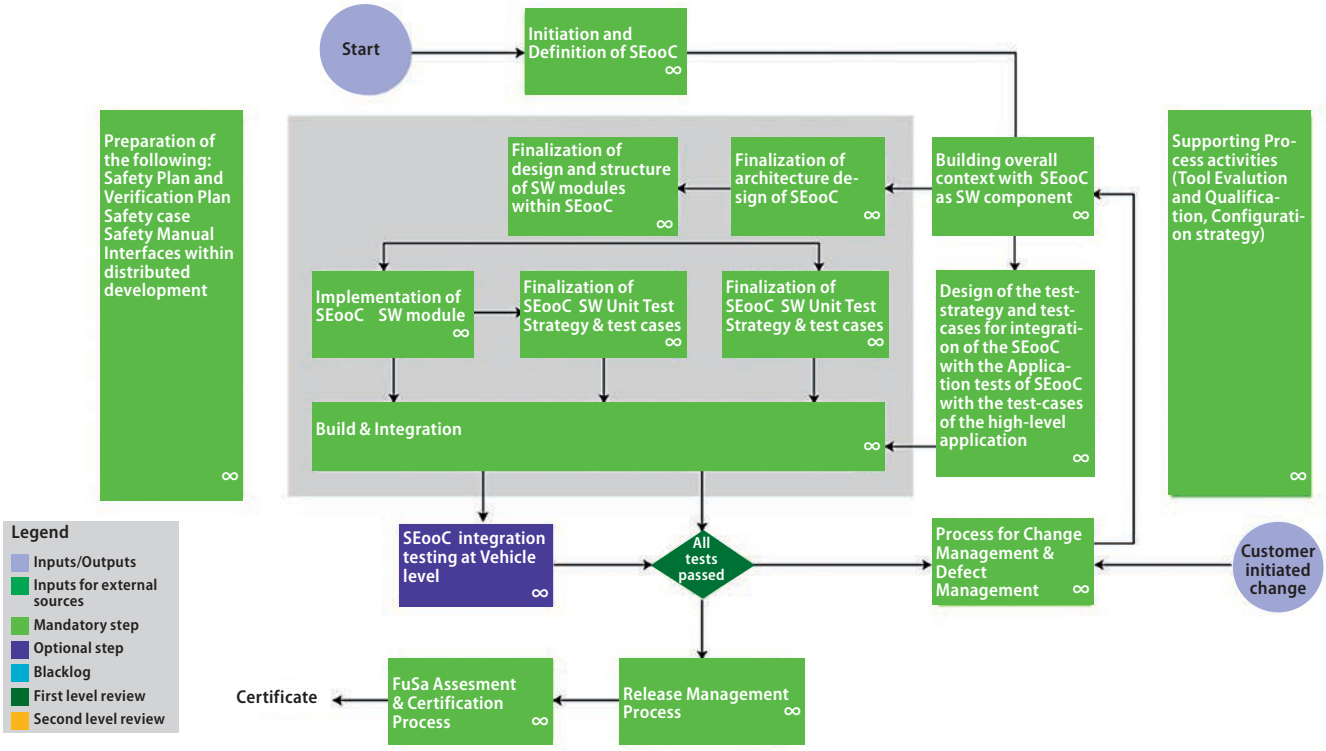


Bild 2: Der ADAS-Softwareentwicklungsprozess von KPIT im agilen Betriebsmodell.

vom Prototyp auf eine sichere produktionsbereite Plattform Erfahrung in den Bereichen Domain, Sicherheit, Simulation und Integration erfordert. Die Plattformsoftware zur Produktionsreife zu bringen und dabei alle Sicherheits- und Regulierungsanforderungen zu erfüllen, ist ein extrem komplexer Vorgang. Sicherheitskritische ADAS- und AD-Software reif für die Produktion zu machen, erfordert ein fundiertes Fachwissen und Erfahrung.

Daher besteht im Nutzfahrzeug-Ökosystem Bedarf für die Zusammenarbeit mit einem „Industrialisierungspartner", der über entsprechende Kompetenz verfügt. Dieser Beitrag stellt Methoden und Best Practices der Software-Industrialisierung für Sicherheitssysteme vor. In ADAS- und AD-Programmen gilt es, die in Bild 1 aufgelisteten Faktoren zu berücksichtigen. Jeder der in Bild 1 erwähnten Punkte umfasst Faktoren, die sich nur aus einem vollständigen Produktionsprogramm heraus verstehen lassen.

Einhaltung von ASPICE- und ISO-Prozessen

Im Allgemeinen konzentrieren sich Plattformanbieter auf die Entwicklung von ADAS- und AD-Funktionen sowie die Erstellung prototypfähiger Lösungen. Diese Lösungen werden dann am Konzeptfahrzeug in kontrollierter Umgebung auf der Straße demonstriert. Bei der Entwicklung dieser Prototypen werden die ASPICE- und ISO-Prozesse nicht befolgt, aber während der Produktionsphase ist es unerlässlich, einen maßgeschneiderten Prozess anzuwenden, alle Dokumentationen einzuhalten und zusätzlich die Entwickler zu schulen, um eine strikte Umsetzung beziehungsweise Einhaltung sicherzustellen.

Ein Software-Industrialisierungspartner übernimmt diese Aufgabe auf dem Weg vom Prototyp zur Produktion, wo er sicherheitskritische Prozesse einführt, die ASPICE als Basisprozess betrachten und alle erforderlichen Prozesse aus ISO 26262, ISO 15288 und anderen Prozessen zusammenführt. Die Implementierung des Prozesses muss dann auch an ein agiles Betriebsmodell angepasst werden. KPIT bietet maßgeschneiderte Prozesse und hat den ISO-Prozess in einer agilen Umgebung bereits umgesetzt.

Die zentrale Bedeutung der Prozessumsetzung liegt nicht in der Prozessdefinition, sondern in der Prozesseinhaltung. Der in Bild 2 erwähnte Prozess wurde unter Berücksichtigung des Selbsterklärungsaspekts und der mit dem Prozess verbundenen Schulung entwickelt. Jeder Entwickler soll dadurch verstehen, was er in der Software tun muss, um den Sicherheitsprozess einzuhalten.

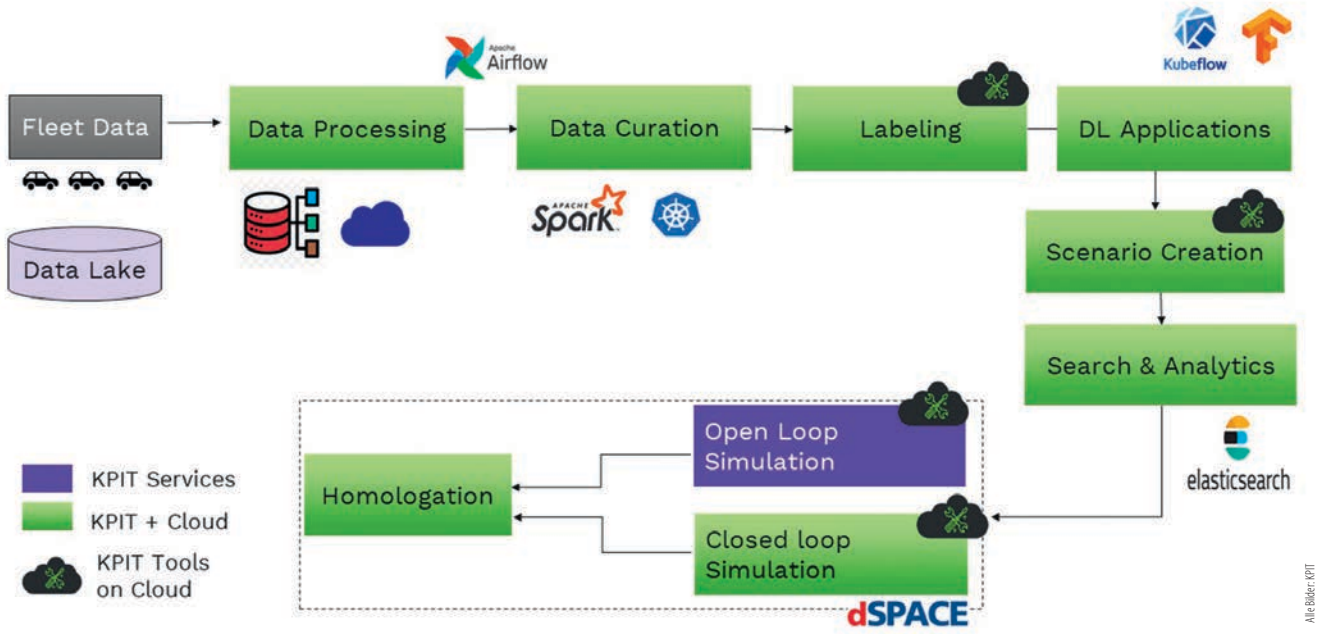


Bild 3: Die Komplettlösung von KPIT für virtuelle Simulation und Abdeckung.

Systementwicklung und -anforderungen

Die Systementwicklung ist ein wichtiger Bestandteil beim Aufbau sicherheitskritischer Systeme. Die Luft- und Raumfahrtindustrie, die ebenfalls sicherheitskritische Systeme für autonomes Fahren erstellt, folgt dem Systementwicklungsprozess bereits in der Konzeptphase mit größter Sorgfalt. Bei der Entwicklung von ADAS und AD ist zu beobachten, dass sich die Entwickler in der Konzeptphase auf die Erprobung des Konzepts in einem Fahrzeug konzentrieren und Systementwicklung, Festlegung von Detailanforderungen sowie Architekturentwicklung für die Folgephasen der Entwicklung vorgesehen sind.

Daher wird es während der Produktionsphase sehr wichtig, die Systemarchitektur sowie die logische und die physische Softwarearchitektur zusammen mit den System-, Software- und Hardwareanforderungen zu erfassen. Zu beachten ist auch, dass die Anzahl der Anforderungen in die Zehntausende geht und sehr große Aufwendungen in Kombination mit umfangreichen Fachkenntnissen erfordert.

KPIT folgt einer Drei-Ansichten-System-Engineering-Methodik, die Skalenexpertise zur Entwicklung von Architektur und Anforderungen mit einer umfangreichen Bibliothek von Artefakten bietet, die Zeit- und Kosteneinsparungen im Programm erzielen.

Umsetzung von ISO 26262 in Prozess, System, Software und Validierung

Die Implementierung der funktionalen Sicherheit gemäß ISO 26262 ist ein Schlüsselfaktor. Beginnend mit dem Sicherheitsprozess sind eine Reihe von Sicherheitsanalysen auf System-, Software- und Hardwareebene durchzuführen.

- Funktionssicherheitskonzept (FSC)
- Gefahren- und Risikoanalyse (HARA)
- Technisches Sicherheitskonzept (FSC)
- Fehlermöglichkeits- und Einflussanalyse (FMEA)
- Fehlerbaumanalyse (FTA)

Neben diesen Aktivitäten sind auch detaillierte Sicherheitsanforderungen Teil der Produktionsphase.

- Anforderungen an die funktionale Sicherheit (FSR)
- Technische Sicherheitsanforderungen (TSR)
- Software-Sicherheitsanforderungen (SSR)
- Hardware-Sicherheitsanforderungen (SSR)

Neben den oben genannten Sicherheitsaktivitäten ist auch die Sicherheitsvalidierung von entscheidender Bedeutung. Es erfordert umfassende Domänenerfahrung und spezielles Fachwissen zum Ableiten von Testfällen anhand von Sicherheitsanforderungen und zum Durchführen von Fehlerinjektionstests sowohl für die Anwendungs- als auch für die Middleware-Software.Die Sicherheit der vorgesehenen Funktion (SOTIF) ist ein weiterer wichtiger Bereich, der in der Produktionsphase zu berücksichtigen ist. KPIT stellt fertig entwickelte Testfallbibliotheken und SOTIF-Szenarien bereit, mit denen sich bei der Produktion Zeit einsparen lässt. Das Unternehmen hat umfassende Engineering-Services für die Umsetzung von ISO 26262 sowie eine Reihe von Beschleunigern anzubieten, mit denen sich Zeit und Kosten optimieren lassen.

Software-Refactoring und -Industrialisierung, insbesondere für Komponenten der KI

Die zentralen Herausforderungen bei ADAS- und AD-Programmen der Stufe 3+ sind der Reifegrad sowie die Qualität der Funktionen und der Algorithmen. In der Prototypenphase findet die Feature-Validierung in einer kontrollierten Umgebung statt. Daher ist es von entscheidender Bedeutung, die Funktionsreife der Software während der Produktion zu erreichen.

Reifegrad und Qualität werden durch Refactoring und Optimierung des Codes erzielt. Der Code muss MISRA, Standards für funktionale Sicherheit und anderen Richtlinien entsprechen und auch die Grundlagen einer Embedded Umgebung befolgen.

Da die meisten der Algorithmen und Funktionen Modelle künstlicher Intelligenz (KI) enthalten, wird das Refactoring des Codes zu einer Herausforderung. Die Schwierigkeit liegt dabei eher in der Neuartigkeit, weil dies eine der ersten Anwendungen ist, bei denen künstliche Intelligenz für sicherheitskritische Anwendungen zum Einsatz kommt und es keine bewährten Methoden gibt, diese einsatzbereit für die Produktion zu machen. KPIT hat eine robuste und praxisbewährte KI-Industrialisierungsmethodik entwickelt.

Softwareintegration und -leistung in der Multicore-Plattform

Nach dem Software-Refactoring ist der nächste wichtige Schritt die Optimierung des Codes, damit er erstens auf der gewünschten eingebetteten Plattform und zweitens mit der gewünschten Geschwindigkeit ausgeführt wird. Dabei bestimmt die Reaktionszeit von der Erfassung bis zur Betätigung die Leistung des gesamten ADAS- und AD-Systems; diese Reaktionszeit hängt von der Qualität, dem Reifegrad und der eingebetteten Implementierung des Codes ab. Während der Implementierung des Prototyps erfolgen Tests von Funktionen oder Algorithmen nur als einzelne Komponenten, so dass die Reaktionszeit in diesem Fall nicht sonderlich kritisch ist.

In der Produktionsphase jedoch kommen Middleware-Komponenten wie Autosar, Adaptive Autosar und Sicherheitskomponenten hinzu, um den Stack zu komplettieren. Das Erreichen der gewünschten Mikrosekunden-Anforderungen in Bezug auf die Antwortzeit hängt zum einen davon ab, wie der Code beim Refactoring optimiert wird, zum anderen von den Architekturentscheidungen zur Partitionierung. Daher kommt der Softwarearchitektur, welche die Partitionierung von Software bezüglich der Ausführung auf verschiedenen Grundlagen in Betracht zieht, eine sehr hohe Bedeutung zu. KPIT bringt Erfahrung mit fast allen SoCs (System On Chip) mit, die im ADAS- und AD-Bereich implementiert sind, sowie Best Practices und Methoden für die Partitionierung und Optimierung.

Virtuelle Simulation und Gewährleistung der Abdeckung

Die Validierung autonomer Software durch Simulation ist heutzutage eine unverzichtbare Praxis. Auch die Validierung durch Simulation wird immer ausgereifter. Allerdings ist die Frage, wie die Abdeckung in der Simulation sichergestellt werden kann, in Produktionsprogrammen noch immer ein ungelöstes Problem.

KPIT hat eine Validierungsstrategie entwickelt, um die Abdeckung von kritischen Grenz- und SOTIF-Fällen sicherzustellen. Die zur Sicherstellung der Abdeckung berücksichtigten Eingaben umfassen Domänenerfahrung, Analyse von Unfalldaten von über 20 Jahren, Sicherheitsanalysen sowie von NHTSA und ISO empfohlene ODDs (Operation Design Domains, Betriebsdesign-Domänen). Bild 3 zeigt die von KPIT entwickelten Bibliothek von Grenz- und SOTIF-Fällen, mit der sich im großen Umfang Zeit und Kosten einsparen lassen.

Homologation und Laufleistungsabdeckung

Die letzten, aber alles andere als unwichtigen Aktivitäten sind die Laufleistungsabdeckung und die Homologation. Dazu gehört nicht nur die SIL/HIL-Methodik sondern auch die Integration von Datenerhebung, Datenaufnahme, Datenmanagement, Analyse und Cloud-Implementierungen. Zusammen mit führenden Industriepartnern hat KPIT eine End-to-End-Methodik entwickelt, um die Prüfstands- und Testmethoden zur Gewährleistung der Homologation umzusetzen.

Schließung der Lücke mit Software-Industrialisierung

Der nächste Schritt besteht darin, Prototypen oder Technologie-Demonstrationen für das autonome Fahren in die Produktion zu bringen, die von Nutzfahrzeug-OEMs und ihren Plattformpartnern entwickelt wurden. Die wichtigsten Anforderungen für diesen Schritt sind das Know-how in puncto ADAS/AD-Software, die Erfahrung mit Produktionsprogrammen und die Skalierung zur Optimierung von Zeit und Kosten. Mit einem adäquaten Partner lassen sich Einsparungen von bis zu 35 Prozent durch fertige Komponenten und Automatisierungstools und -frameworks erzielen. *(av)* ■

Autorin
Dr. Manaswini Rath
Vice President & Global Head of Autonomous Driving, KPIT Technologies

noffz.com

NOFFZ TECHNOLOGIES

SMART TEST SYSTEMS FOR THE FUTURE OF MOBILITY

Product Validation

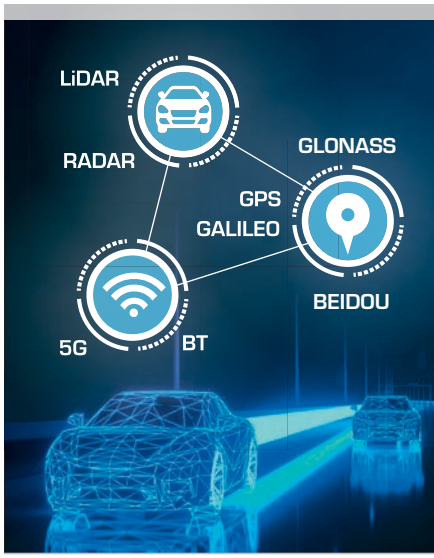
UTP 7033
Cellular, GNSS & Wireless

Board-Level-Test

UTP 9011
Multi DUT RF Test

End-of-Line-Test

UTP 5065RTS
Radar Test System & 5G OTA Test



FAST > FLEXIBLE > FOCUSED