

# Moderne Fahrzeugentwicklung entlang des V-Modells mit ASPICE

Paavo Schriegel, Dr.-Ing. Matthias Grünewald  
UNITY AG, Büren

## Abstract

Die Fahrzeugentwicklung hat in den letzten Jahren ein neues Level der Komplexität erreicht, da Themen wie autonomes Fahren und das Auto als digitales Produkt in den Fokus geraten sind. Um auf die kommenden Herausforderungen zu reagieren, begleitet UNITY die deutschen Automobilhersteller bei ihrer Transformation zu Softwareunternehmen, indem Prozesse, Methoden, Tools und Organisationen an die neuen Anforderungen angepasst werden. Dabei hat sich besonders die Veränderung der Organisation als kritischer Erfolgsfaktor herausgestellt.

## Herausforderungen der Automobilindustrie

Die Entwicklung komplexer Systeme aus Hard- und Software hat mit dem Eintreten von Tesla die Automobilbranche nachhaltig verändert. Das Auto ist nicht mehr ein analoges Produkt mit digitalen Funktionen, sondern viel mehr ein digitales Produkt mit analogen Funktionen. Neben einem mit dem Smartphone verbundenen Infotainment übernimmt das Auto weitere Aufgaben des Fahrers und wird in Zukunft wahrscheinlich völlig autonom fahren. Die daraus resultierende rasante Geschwindigkeit in der Entwicklung in Verbindung mit stetig steigenden Kundenanforderungen stellen den Automobilstandort Deutschland vor große Herausforderungen. Zum einen können sich die Fahrzeughersteller nicht mehr auf eine reine Softwareentwicklung durch die Zulieferer verlassen und zum anderen sind bestehende Fahrzeug- und Entwicklungsstrukturen nicht auf die neuen Anforderungen und Geschwindigkeit ausgerichtet. Um den Vorsprung von Tesla aufzuholen, haben die deutschen Automobilhersteller daher eigene Entwicklungskompetenzen aufgebaut, um leistungsstarke und zuverlässige Software erstellen und validieren zu können. Außerdem wurde die Hardware-Architektur von vielen verteilten auf wenige „Electronic Control Units“ (ECUs) umgestellt, von denen einige sehr leistungsfähige ECUs die anderen in Zonen steuern [1]. Diese besonders leistungsfähigen ECUs werden auch als Fahrzeugcomputer bezeichnet.

Die Fahrzeugcomputer reduzieren die Komplexität und ermöglichen die Entwicklung eines zentralen

Fahrzeug-Betriebssystems. Dadurch können Funktionen hauptsächlich in Software realisiert und durch „Over-The-Air“-Updates erweitert werden [2] [3]. Durch die neuen Strukturen sind die deutschen Fahrzeughersteller aktuell in der Lage, in mehreren Bereichen große Erfolge zu erzielen. Zum Beispiel erreichte Mercedes Benz als erster Fahrzeughersteller weltweit das SAE-Level 3<sup>1</sup> des autonomen Fahrens [4].

Um das aufgebaute Momentum beizubehalten, ist bei den Automobilherstellern aber auch ein erweiterter Wandel in der gesamten Entwicklung des Fahrzeuges erforderlich. So müssen bestehende Methoden für die Validierung und Verifizierung von Anforderungen weiter optimiert werden, um die Wartung der Fahrzeuge über den gesamten Lebenszyklus sicherzustellen. Diese Herausforderungen erreichen ein neues Komplexitätslevel im Zusammenspiel der unterschiedlichen ECUs, die in einem Fahrzeug verbaut sind. Neben der Interaktion und den daraus folgenden Abhängigkeiten untereinander muss auch die Integration zwischen der Software in unterschiedlichen Versionen mit allen aktuellen Fahrzeugmodellen und -konfigurationen sichergestellt werden. Ein weiterer Komplexitätstreiber ist ASPICE<sup>2</sup> als ein Quasi-Standard in der Automobilbranche. Dieser setzt eine Nachvollziehbarkeit von der Anforderung bis zur implementierten Funktion auf semantischer und syntaktischer Ebene voraus [5].

Um den genannten Herausforderungen angemessen zu begegnen, bedarf es einer ganzheitlichen Teststrategie. Sie muss sicherstellen, dass die geänderten Funktionen in Entwicklungs- und auch in Kundenfahrzeugen aktuell und sicher gehalten werden. Für diesen Wandel reicht es jedoch nicht aus, nur die eigentliche Validierung und Verifizierung zu betrachten, sondern es muss das gesamte Ökosystem untersucht werden. Dies beginnt klassisch im Sinne des System Engineering mit der Anforderung über die einzelnen Architekturelemente und Testfälle bis zum Ausrollen der implementierten Funktion über OTA-Updates. Vor dem Hintergrund dieser Herausforderungen und der Erfahrungen aus den UNITY Projekten bei den deutschen Automobilherstellern wird nachfolgend eine gesamtheitliche Veränderung der Prozesse, Methoden, Tools und Organisation tiefer beleuchtet.

<sup>1</sup> SAE-Level 3 ist eine Stufe im Rahmen des autonomen Fahrens der Society of Automotive Engineers (SAE). Sie beschreibt den Grad der Automatisierung von Fahrzeugen [8].

<sup>2</sup> ASPICE (Automotive SPICE) ist ein internationaler Standard für die Prozessbewertung und -verbesserung in der Automobilindustrie [4].

## Standardisierte Prozesse als Enabler

Zentrales Mittel für die Befähigung einer umfassenden Verifikationsstrategie stellen resiliente und gelebte Prozesse dar. Das Framework ASPICE liefert hier bereits ein breites Portfolio an Bewertungskriterien für Prozesse und dessen Umsetzung bei den Fahrzeugherstellern sowie dessen Zulieferern. Diese reichen von der einfachen Erreichung der Prozessziele über zentrale Standardprozesse bis zu selbstoptimierenden und datengetriebenen Prozesslandschaften. Abbildung 1 verdeutlicht die 5 Stufen, anhand ASPICE die Entwicklungsprozesse bewertet. Für die Automobilhersteller sind aktuell die ersten beiden Level realistische Ziele und werden in der Praxis aktiv verfolgt.



Abbildung 1 Die 5 Bewertungsstufen von ASPICE [5]

Das Ziel des ersten ASPICE Reifegrades (Stufe 1) ist die Etablierung einer grundlegenden Infrastruktur für die Software-Entwicklung in der Automobilindustrie, einschließlich einer klaren Organisation und Dokumentation von Prozessen und Methoden. Das zusätzliche Ziel des zweiten ASPICE Reifegrades (Stufe 2) ist auch die Standardisierung der Entwicklungsprozesse und die Einrichtung eines umfassenden Qualitätsmanagementsystems, um die Wiederholbarkeit und Vorhersagbarkeit von Prozessen zu verbessern. Dies beinhaltet auch die Schulung des Personals und die Einführung von Bewertungen und Audits, um die Einhaltung der Prozesse sicherzustellen [5].

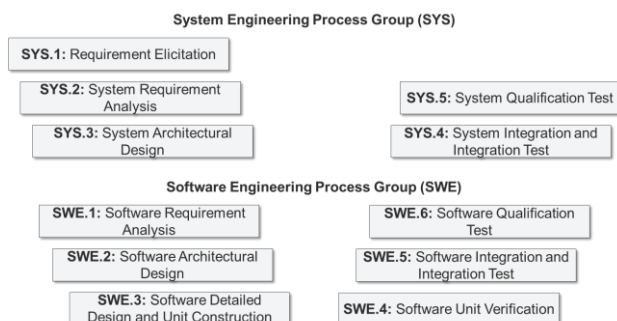


Abbildung 2: SYS- und SWE-Prozesse in ASPICE [5]

In den Projekten zeigt sich, dass die Hersteller besonders in die ASPICE-Bereiche für das System- (SYS) und Software-Engineering (SWE) investieren (siehe Abbildung 2), um die Anforderungen ganzheitlich, kosteneffizient und im Sinne von ASPICE umzusetzen. Im Kern der aktuellen Bestrebungen steht

folglich eine sukzessive Erweiterung der prozessualen Fähigkeiten, die Erstellung der geforderten Artefakte und eine verlässliche Nachvollziehbarkeit zwischen diesen. Hierbei ist es wichtig, dass die Ziele für die Soll-Prozesse realistisch bleiben und im ersten Schritt eine Harmonisierung oder Standardisierung nicht zwingend notwendig ist, sondern die erzeugten Artefakte und deren Qualität je Entwicklungsteam im Fokus steht. Erst wenn dies erreicht wurde, kann eine zentrale Steuerung und Verwaltung der Prozesse für die zweite ASPICE Stufe in Angriff genommen werden.

In den einzelnen Disziplinen (SYS und SWE) besitzen die Fahrzeughersteller bereits bestehende und auch im Kontext der agilen Arbeit weitreichende Prozesse, mit denen einzelne ECUs in sich getestet werden können. Die relevanten Lösungsbereiche befinden sich in der Integration und Synchronisation mit anderen Entwicklungsteams sowie zwischen der Software- und Systemebene. Denn auch wenn eine Harmonisierung im ersten Schritt nicht notwendig und in der Praxis meist auch nicht realistisch ist, stellen die gewachsenen heterogenen Prozessstrukturen die zentrale Herausforderung für die Prozesse dar. Diese Herausforderungen werden bei einer Betrachtung der Fahrzeugarchitektur weiter verdeutlicht. Die Abbildung 3 zeigt, dass ein einzelnes Fahrzeug in mehrere Zonen unterteilt werden kann. Jede Zone besteht aus mehreren ECUs, die wiederum die Software-Komponenten ausführen. Einer dieser ECUs ist der besonders leistungsfähige Fahrzeugcomputer.

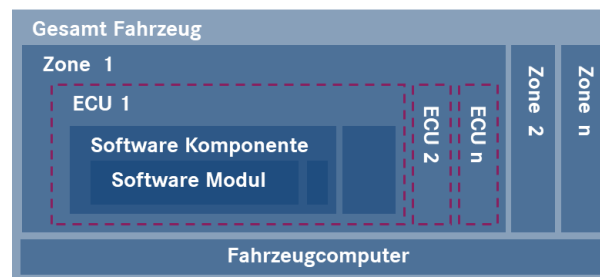


Abbildung 3: Schematische Darstellung der Fahrzeugarchitektur

Die Entwicklungsprozesse müssen diese Komplexität ebenfalls abbilden können und Anforderungen, die sich hier in Wechselwirkungen zwischen den ECUs und den Disziplinen bewegen, steuern. Dabei gilt es die manuellen Aufwände und das nötige Personenwissen zu minimieren.

Dies kann in einem ersten Schritt in der Bearbeitung der prozessualen Schnittstellen zwischen den einzelnen Zonen und ECUs erfolgen. Ziel ist es, die Schnittstellen so weit wie möglich zu beschreiben und zu standardisieren. So kann auch bei einer heterogenen Prozesslandschaft die Interaktion zwischen den einzelnen Bereichen erfolgen, ohne sie in ihrer Individualität einzuschränken. Zusammen mit einer umfassenden Dokumentation der Prozesse und der

Einführung von Steuerungs- und Verbesserungsmechanismen können so die ersten beiden ASPICE Stufen erreicht werden.

### Simulation als zentrale Methodik

In der Praxis hat sich der Einsatz von Methoden aus der agilen Softwareentwicklung gepaart mit dem verstärkten Einsatz von Simulationen als erfolgreich erwiesen. So können die Artefakte der einzelnen Bereiche frühzeitig integriert getestet werden. Die aktuell vorherrschenden Methoden in der Entwicklung von ECUs stellen daher Simulationen wie bspw. Software in the Loop (SIL) und Hardware in the Loop (HIL) da [6].

Diese werden bereits seit mehreren Jahren erfolgreich in der Automobilbranche eingesetzt. Sie ermöglichen Entwicklern frühzeitig und effizient zu testen, ohne dass reale Hardware erforderlich ist. Die Methoden tragen dazu bei, dass die Qualität und Geschwindigkeit der Entwicklung bereits verbessert wurde. Voraussetzung für die erfolgreiche Anwendung solcher Methoden ist die Verfügbarkeit der benötigten Informationen in einer vorgegebenen Qualität und Struktur sowie eine gesamtheitliche Übersicht über den aktuellen Teststatus des Fahrzeuges.

### Die Toolchain als technischer Backbone

Neben den Prozessen und der Methodik wird für die Entwicklung von Fahrzeugen eine gesamtheitliche Toolchain benötigt. Diese integriert von der Gesamtfahrzeugebene bis zur einzelnen Funktionsentwicklung alle nötigen Informationen, setzt sie miteinander in Beziehung und stellt sie dem Nutzer zur Verfügung. So können die Informationen über den gesamten Prozess und für die beschriebenen Methoden genutzt werden. Hierbei müssen ebenfalls die Fahrzeugarchitektur und die Organisation beachtet werden.

Kernziel der Toolchain ist die Komplexität für den Anwender beherrschbar zu machen und die Anforderungen aus Anwender- und ASPICE-Sicht zu unterstützen. Wie bereits bei den Prozessen gilt es zu prüfen, welche Tools bereits im Einsatz sind. Die Praxis zeigt, dass es analog zu den Prozessen in den Unternehmen keine einheitlichen Toolchains gibt, sondern starke Unterschiede zwischen den Abteilungen und deren Schwerpunkten bestehen. Dies lässt sich auf unterschiedliche Anforderungen als auch ein fehlendes übergeordnetes Architektur-Management zurückführen.

Auch beim Vorgehen zur Entwicklung der Toollandschaft kann sich an den Prozessen orientiert und ein iteratives Vorgehen genutzt werden. Ziel ist es weniger die einzelnen lokalen Toolchains zu optimieren bzw. zu harmonisieren, als vielmehr die Integration zwischen den Bereichen sicherzustellen und Standards für neue Projekte bereitzustellen. Kern der Toolchain stellen daher die Management-Tools dar, die den Lebenszyklus

der Entwicklung verwalten. Diese sollten unabhängig des Bereichs einheitlich sein und die in den Prozessen definierten Schnittstellen abbilden. Außerdem können die Anforderungen von ASPICE zwar durch die Tools nicht allein umgesetzt, aber durchaus unterstützt werden.

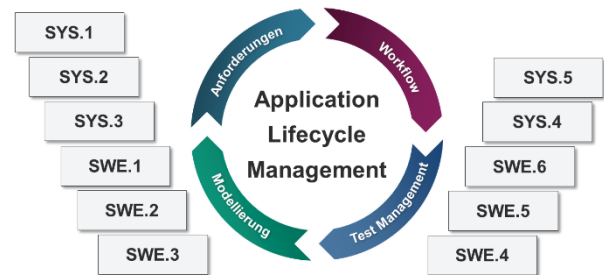


Abbildung 4: Aufgabenfelder der Management-Tools

Abbildung 4 fasst die vier notwendigen Aufgabenfelder zusammen, die die Management-Tools über alle Prozesse hinweg unterstützen müssen.

Dazu zählt ein zentrales Anforderungsmanagement inkl. Rückverfolgbarkeit, Versionierung und Konsistenz, das Verwalten der Tests inkl. Rückführung der Testergebnisse, die Aufnahme der erstellten Modelle während des System- und Softwaredesigns und das Verwalten der Aufgaben, die während der Kollaboration zwischen den Teams entstehen. Dabei müssen die Management-Tools insbesondere die Verlinkung aller dieser Artefakte beherrschen, um eine Rückverfolgbarkeit zu ermöglichen. Dadurch werden alle notwendigen Informationen über alle Teams hinweg vollständig transparent und die Kollaboration wird sichergestellt. Als erfolgreich hat sich die Nutzung der Anforderung als zentrales Architekturelement erwiesen, da so auch in einer Toolbetrachtung der Nutzer in den Mittelpunkt gestellt wird.

In der Praxis setzen die Hersteller für die Aufgabenfelder Tools von unterschiedlichen Anbietern ein. Zum Beispiel wird für Anforderungen und Modellierung häufig die IBM Jazz Plattform und für den Workflow Atlassian Jira benutzt. Jira bietet aber kein eingebautes Test-Management, so dass Plugins von anderen Herstellern wie z.B. Zephyr oder XRay dazu kommen. Damit wird insbesondere die durchgängige Rückverfolgbarkeit aller Artefakte eine Herausforderung, da Schnittstellen zwischen den Tools unterschiedlicher Hersteller realisiert und gewartet werden müssen. Eine wichtige Erkenntnis aus der Praxis ist, dass die Entwicklungsteams meistens einen zu starken Fokus auf die Tools legen. Die Tools alleine stellen zwar einen wichtigen Teil dar, können aber lediglich als unterstützendes Element in Verbund mit den Prozessen und der Methodik zum gewünschten Ergebnis führen. Es ist also essenziell, dass zuerst die Prozesse und Methoden definiert werden müssen, bevor die Tools ausgewählt werden.

## Die Organisation als kritischer Erfolgsfaktor

Wenn in einem Unternehmen Prozesse, Methoden und Toolchains verändert werden sollen, muss sich die Organisation an die neuen Vorgehensweisen anpassen. Diese Veränderung hat sich in den UNITY Projekten als größte Herausforderung herausgestellt, da die beteiligten Menschen den Wandel akzeptieren und durchführen müssen. Denn er stellt die verbindende Schicht zwischen den beschriebenen Prozessen, Methoden und Tools dar und sollte daher stets im Mittelpunkt der Projekte stehen. Abbildung 5 zeigt dazu die vier wichtigsten Herausforderungen und wie die UNITY sie in ihren Projekten adressiert.



Abbildung 5: Die vier Herausforderungen des Wandels

Das bereits beschriebene Vorgehen zu Prozessen, Methoden und Tools sorgt bereits dafür, dass die ersten beiden Herausforderungen gelöst werden können. Wichtiger ist aber, dass die Unternehmenskultur die Veränderung ermöglicht und dass die Veränderung auch richtig umgesetzt wird. Damit ist ein essenzieller Bestandteil des Wandels in der deutschen Automobilbranche eine organisatorische Frage. Denn es müssen bestehende Strukturen in der Entwicklung und die Angst vor einer Veränderung frühzeitig thematisiert und neutralisiert werden. Zentraler Ansatz ist, dass die relevanten Stakeholder frühzeitig und regelmäßig abgeholt werden und zu jedem Zeitpunkt eine Kundenzentrierung vorgenommen wird und die Anforderungen aus der Sicht von ASPICE zu implementieren. Durch Aufzeigen der möglichen Verbesserungen können Skeptiker zu Sponsoren werden, die intrinsisch motiviert die Veränderung vorantreiben. Nur wenn diese organisatorischen Herausforderungen identifiziert, akzeptiert und mit Maßnahmen versehen sind, können die Veränderungen auch erfolgreich operationalisiert werden.

Hierfür kann auf Maßnahmen aus dem Change-Management zurückgegriffen werden, die Formate der Partizipation, des Informierens und der stetigen Interaktion bereitstellt. Bei einem gewissenhaften Einsatz der Maßnahmen kann auch die richtige Geschwindigkeit der Transformation erkannt und reguliert werden, um auf der einen Seite die

Organisation nicht zu überfordern, aber auf der anderen Seite auch eine merkliche Veränderung zu ermöglichen.

Auch können Best Practices wie bspw. das Scaled Agile Framework (SAFe) bei der Transformation unterstützen. Es bietet für Unternehmen Möglichkeiten, um die beschriebenen Schnittstellen zusammenzuführen und die Entwickler in den Mittelpunkt zu stellen [7]. Dabei müssen aber unbedingt die Rollen des etablierten Managements zuerst geklärt werden, da agile Frameworks die Entscheidungskompetenzen an selbstorganisierende Teams übergeben und damit das Management sich ebenfalls verändern muss. Wichtig zu erwähnen ist aber auch, dass der Widerstand nicht immer rational ist und jedes einzelne Individuum durch seine eigene Veränderungsreise begleitet werden muss.

## Fazit

Zusammenfassend lässt sich sagen, dass bereits eine Vielzahl an Prozess-Frameworks, Methoden und Tools existieren, die eine Fahrzeugentwicklung unter den neuen Anforderungen in Bezug auf Geschwindigkeit und Qualität ermöglichen. Die Praxis zeigt aber auch, dass die organisatorische Veränderung der kritische Erfolgsfaktor für den Automobilstandort Deutschland ist. Nur bei einer gesamtheitlichen Veränderung können die bestehenden Strukturen aufgebrochen und mögliche Verbesserungen durch harmonisierte Prozesse, neuartige Tools und Methoden realisiert werden. Diese neuen Technologien in Zusammenhang mit einer veränderungsbereiten Organisation stellen auch den Grundstein für künftige Veränderungen dar. Dadurch kann auf zukünftige Einflüsse schneller reagiert und neue Innovationen aus Deutschland getrieben werden.

## Referenzen

- [1] Bosch, „Die E/E-Architektur der Zukunft,“ Bosch, [Online]. Available: <https://www.bosch-mobility-solutions.com/de/mobility-themen/ee-architektur/>. [Zugriff am 20 03 2023].
- [2] Mercedes Benz Group, „Mercedes Benz Operating System,“ [Online]. Available: <https://group.mercedes-benz.com/karriere/ueberuns/mercedes-benz-operating-system/>. [Zugriff am 19 03 2023].
- [3] CARIAD, „CARIAD Website - Unified Software,“ CARIAD, [Online]. Available: <https://cariad.technology/de/en/solutions/unified-software.html>. [Zugriff am 19 03 2023].
- [4] Mercedes Benz Group, „Mercedes-Benz DRIVE PILOT,“ [Online]. Available: <https://www.mercedes-benz.de/passengercars/technology/drive-pilot.html>. [Zugriff am 19 03 2023].
- [5] VDA QMC Working Group 13 / Automotive SIG, „Automotive SPICE,“ 01 11 2017. [Online]. Available: [https://www.automotivespice.com/fileadmin/software-download/AutomotiveSPICE\\_PAM\\_31.pdf](https://www.automotivespice.com/fileadmin/software-download/AutomotiveSPICE_PAM_31.pdf). [Zugriff am 19 03 2023].
- [6] G. Dr. Sievers, C. Seiger, M. Peperhowe, H. Krumm und S. Dr. Graf, „Driving Simulation Technologies for Sensor Simulation in SIL and HIL Environments,“ *Proc. Driving Simul. Conf. Europe*, pp. 127 - 130, 2018.
- [7] SCALED AGILE, INC, „Scaled Agile Framework,“ SCALED AGILE, INC, [Online]. Available: <https://scaledagileframework.com/>. [Zugriff am 19 03 2023].
- [8] SAE International, „Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles,“ 30 04 2021. [Online]. Available: [https://www.sae.org/standards/content/j3016\\_202104/](https://www.sae.org/standards/content/j3016_202104/). [Zugriff am 19 03 2023].