

# Das softwaredefinierte Fahrzeug und wie man den etablierten, hardwarezentrierten Entwicklungsprozess transformieren kann

Andrea Denger, Program Manager Information Management, Software Products, Emission & Service, AVL List GmbH  
Peter Ebner, Chief Engineer Hybrid Systems, Systems Engineering & Powertrain Electrification, AVL List GmbH  
Katrin Moser, Marketing Manager, Integrated and Open Development Platform, AVL List GmbH  
Josef Zehetner, Chief Engineer System Architecture, Integrated and Open Development Platform, AVL List GmbH

Was haben ein Fotoalbum und ein Fahrzeug gemeinsam? In der Vergangenheit drehte sich alles um die Hardware, die physischen Objekte, die man anfassen kann. In den letzten Jahrzehnten wurden die Funktionen erweitert und die Bedeutung und der Einfluss von Software haben zugenommen. Software allein ist jedoch nicht die Lösung. So wie Sie Ihre digitalen Fotos nicht ohne ein Smartphone oder ein Tablet betrachten können, können Sie auch keine Software allein nutzen, um an Ihr gewünschtes Reiseziel zu gelangen.

Das Fahrerlebnis der Mobilitätskonsumenten wird auch weiterhin von Hardware und deren Funktionalität dominiert. Es wird davon bestimmt, was sie haptisch erfassen, physisch spüren können und wie sie sich mit den Fahrzeugfunktionen beschäftigen können. Mobilität findet nicht im Metaverse statt – eine reale Person interagiert mit dem Fahrzeug und bewegt sich physisch zu ihrem Ziel. Die Verbraucher sind nicht an der Nutzung einer Software an sich interessiert, sondern an einem Produkt, das individuelle Funktionen bietet. Für ein zufriedenstellendes Kundenerlebnis ist ein bedarfsorientierter End-to-End-Ansatz unerlässlich.

In der Automobilindustrie gibt es ein neues Modewort: Software-definiertes Fahrzeug. Beispielsweise verlagert sich die Unterscheidung zwischen den Fahrzeugvarianten von der Hardware hin zu reinen Software-/Kalibrierungsdaten (z.B. Verwendung der gleichen Motorhardware, aber unterschiedlicher Leistungsvarianten). Das Fahrzeug könnte zu einem softwaredefinierten IoT-Objekt (Internet der Dinge) werden. Und es besteht kein Zweifel, dass dieser Wandel viele weitere Möglichkeiten bietet – von digitalen Geschäftsmodellen bis hin zu digitalen Dienstleistungen, Konnektivität und Cloud-Diensten und neuen Arten der Zusammenarbeit bzw. von Partnerschaften.

Aber wir dürfen nicht vergessen, dass es immer noch die Hardware ist, die die Softwarefunktionen bzw. das Kundenerlebnis in unsere Welt bringt. Diese Hardware wird bei einem Fahrzeughersteller entwickelt, getestet, ausgeliefert und gewartet. Darüber hinaus bleibt die Hardware über die Lebensdauer größtenteils unverändert und muss zuverlässig, langlebig und sicher sein. Folglich darf die Software, die diese Hardware betreibt, die Haupteigenschaften der Hardware nicht verändern und muss die verfügbaren Kunden- und Umgebungschnittstellen nutzen.

Was würde dies in Bezug auf Softwaredefinierte Fahrzeuge erfordern? Wie können wir den traditionellen Entwicklungsprozess anpassen, oder brauchen wir einen neuen Entwicklungsansatz? In diesem Artikel werden wir zwei Bereiche diskutieren, die großes Potenzial haben:

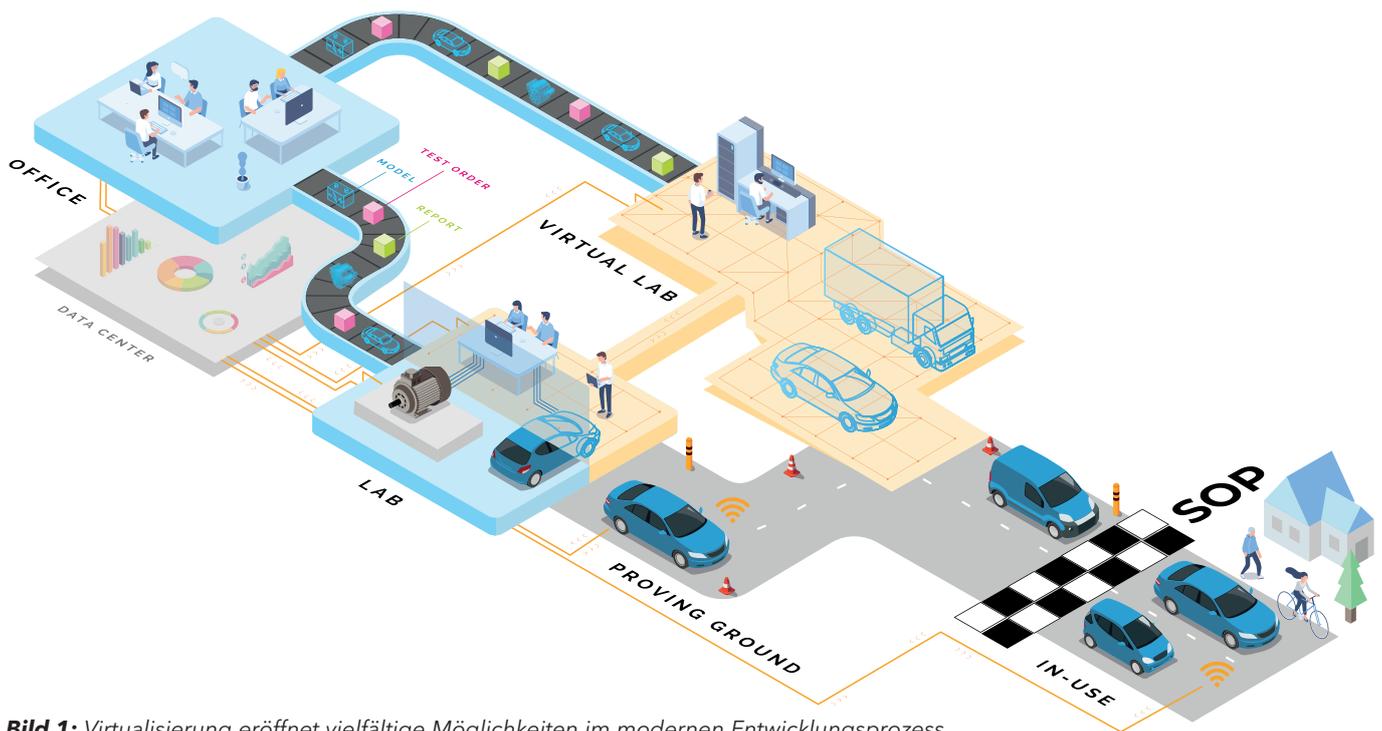
- Virtuelle Entwicklung mit Software-in-the-Loop (SiL)-Systemen
- Kontinuierliche Verifizierung und Validierung mit virtuellen, gemischten und realen Prototypen

Schauen wir uns jeden dieser Bereiche genauer an.

## Virtuelle Software-Entwicklung

Die Herausforderung beim physischen Testen – d. h. beim gemeinsamen Testen von Hardware und Software – ist nicht nur, dass es viel Zeit und Geld kostet. Die verfügbaren Hardware-Prototypen sind kostenintensiv und von der Anzahl der Varianten her eingeschränkt. Jeder festgestellte Fehler könnte entweder von der Software oder von einer nicht ordnungsgemäß funktionierenden Hardware herrühren.

Moderne Entwicklungsprozesse und Werkzeugketten versuchen, die Verifikation mit realer Hardware auf eine begrenzte Anzahl von Testläufen (z.B. zum Nachweis des Erreichens von bestimmten Qualitätsstufen) mit einer Variante zu reduzieren. Alle weiteren Testläufe und Kalibrierungsaufgaben für Software- oder Funktionsvarianten werden mit einem SiL-Ansatz durchgeführt, der eine funktionierende virtuelle Hardware in Kombination mit der Steuerungssoftware berücksichtigt. Dies reduziert den Entwicklungsaufwand, den Zeit- und Kostenaufwand sowie die benötigten Ressourcen auf der Hardwareseite drastisch (Bild 1).



**Bild 1:** Virtualisierung eröffnet vielfältige Möglichkeiten im modernen Entwicklungsprozess

Wir müssen nur noch entscheiden:

- Welche Tests sollten wir für welche Variante durchführen?
- Welche Tests können wir ohne Anpassung wiederverwenden?
- Welche Tests brauchen wir nicht für jede Variante zu wiederholen?

Darüber hinaus können wir auch Folgen des in Betracht ziehen:

- Abweichungen im Produkt aufgrund von Alterung oder Produktionstoleranzen
- Ungewöhnliche Nutzung und Missbrauch

Diese spezielle Art des Frontloading erlaubt es, Integrationstests für zwei oder mehr Steuerungssysteme innerhalb eines einzigen SiL-Systems durchzuführen, um das Zusammenspiel dieser – meist unabhängig voneinander entwickelten – Softwaremodule zu überprüfen. So können wir beispielsweise einen Test mit einem Motorsteuergerät, einem Getriebesteuergerät und einem Emissionsmodell durchführen und so eine gesamte globale Validierungsmatrix in einer Nacht auf dem SiL-System testen, um kritische Fälle oder Probleme im Zusammenhang mit Emissionen bei der Interaktion zwischen Steuergeräten zu identifizieren.

Die Komplexität wird aufgrund von Themen wie Over-the-Air (OTA) Software-Updates in der Nutzungsphase weiter zunehmen. Neben einfachen Funktionserweiterungen (ohne Auswirkungen auf die Hardware) sind auch drastische Änderungen des Gesamtsystemverhaltens denkbar (oder bereits umgesetzt). Nehmen wir zum Beispiel die Erhöhung der Ladeleistung einer Batterie im Feld. Welche Art von Validierung würde das erfordern? Im Feld gibt es sowohl neue Batterien als auch Batterien, die bereits einen beträchtlichen Teil ihrer Gesamtleistung zurückgelegt haben. Was muss getan werden, um sicherzustellen, dass diese neue Funktion im Fahrzeug alle Anforderungen erfüllt – sowohl in Bezug auf die Homologation als auch in Bezug auf Sicherheit, Zuverlässigkeit, Robustheit, Garantie usw.?

Modellbasiertes Systems Engineering (MBSE) ist eine der wichtigsten Methoden, um Komplexität zu strukturieren und Zusammenhänge zwischen Hardware und Software oder den zugehörigen Schnittstellen herzustellen. MBSE ermöglicht es auch, die spezifische funktionale Sicht (end-to-end) eines Produkts zu generieren, einschließlich der entsprechenden Hardware- und Softwareteile.

### Kontinuierliche Verifizierung & Validierung (V&V)

Der Wandel von einer hardwarezentrierten und produktstrukturgesteuerten Entwicklung hin zu einer softwaredefinierten, fahrzeugzentrierten Entwicklung hat mehrere Auswirkungen:

- Die Anzahl der mittels Software realisierten Funktionen im Fahrzeug wird kontinuierlich steigen, auch nach dem Produktionsstart (SOP) bzw. während der Nutzungsphase
- Der Entwicklungsaufwand für Softwarefunktionen wird drastisch steigen, unter anderem durch ADAS/AD Funktionen
- Softwarefunktionen müssen über Fahrzeugvarianten und -typen hinweg wiederverwendet werden (z.B. mittels eines Software-Baukasten-Systems bzw. Modularisierung)
- Klar definierte Schnittstellen zwischen Hardware und Software werden entscheidend sein

Ziel ist es, Software und Hardware entkoppelt voneinander zu entwickeln: Hardware nach dem traditionellen V-Prozess, Software mit einem agilen DevOps-Ansatz. So kann jeder in seiner eigenen Geschwindigkeit arbeiten, während gleichzeitig die Software in Continuous Integration/Continuous Deployment (CI/CD) Zyklen gegen alle Fahrzeugvarianten validiert werden

kann, indem Hardware und Software zusammengeführt werden (Bild 2). Hierfür benötigen wir virtuelle, gemischte und reale Prototypen der Fahrzeughardware.

Mit Hilfe von funktionalen Prototypen können wir den Zustand eines Produkts während des gesamten Entwicklungsprojekts beurteilen: Wir vergleichen die erreichten Test-Ergebnisse virtueller, gemischter oder realer Prototypen mit den definierten Zielerfordernissen.

### Zusammenfassung

Die Transformation des Autos zu einem softwaredefinierten Fahrzeug ist bereits in vollem Gange. Um das Beste aus Ihren Entwicklungsprozessen herauszuholen, müssen Sie Ihre Arbeitsweise überdenken:

- Die Entwicklungsaufgaben zwischen der Hardware und der virtuellen Umgebung intelligent verteilen
- Klare Schnittstellen zwischen Hardware und Software definieren
- Kontinuierliche Verifizierung und Validierung einsetzen, um Software und Hardware integriert, gemeinsam zu testen
- MBSE liefert den erforderlichen Kontext und die Informationen

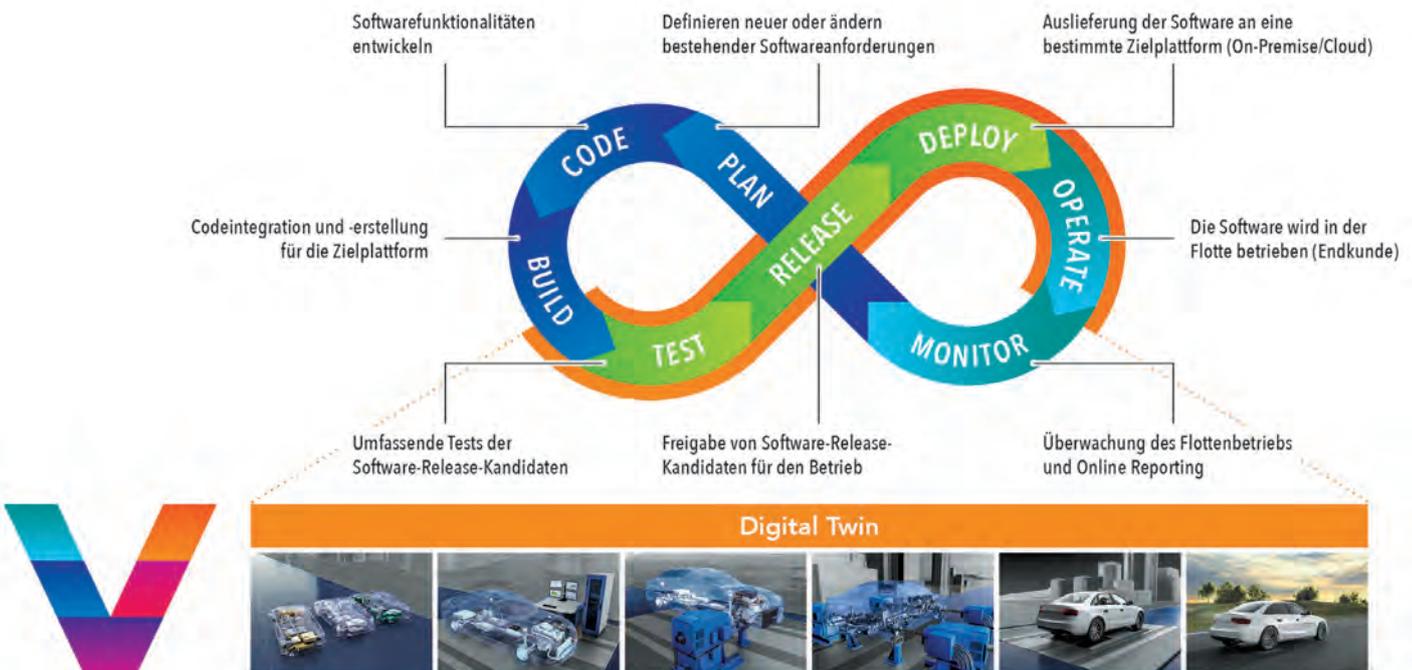
Der DevOps-Prozess und die Softwareentwicklung ermöglichen deutlich schnellere Freigabezyklen von neuen oder verbesserten Funktionen für die Verbraucher. Die erhöhte Frequenz wird zu einem Paradigmenwechsel in der Validierung führen: Die reine Simulation in der Produktentstehung und die kontinuierliche Validierung und Implementierung bis in die Nutzungsphase werden massiv an Bedeutung gewinnen.

Auch die Art und Weise, wie Sie mit anderen zusammenarbeiten, müssen Sie neu überdenken: Im Kontext der digitalen Transformation und modellbasierter Herangehensweisen werden Abläufe zu Closed-Loop-Ansätzen - von den Anforderungen über das Design bis zur Validierung, mit eindeutigen Schnittstellen von Hard- und Software. Daran schließt sich die Produktion in einer vernetzten Smart Factory an, die jeden Auslieferungszustand eines Autos kennt, als Basis für z.B. nachgelagerte OTA-Dienste in der Nutzungsphase. Klar definierte Standards und Richtlinien sowie eine allgemeine Offenheit der Systeme werden es Unternehmen ermöglichen, ihre eigenen Grenzen der Interoperabilität zu überwinden und gemeinsame Vorteile zu erzielen.



## Kontakt

Andrea Denger  
 Program Manager Information  
 Management, Software Products,  
 Emission & Service  
 AVL List GmbH  
 +43 316 787 8756  
 andrea.denger@avl.com



**Bild 2:** DevOps und V-Prozess verbunden in einem symbiotischen Gesamtprozess



Kontakt

Peter Ebner  
 Chief Engineer Hybrid Systems,  
 Systems Engineering & Powertrain  
 Electrification  
 AVL List GmbH  
 +43 316 787 3299  
 peter.ebner@avl.com



Kontakt

Katrin Moser  
 Marketing Manager, Integrated and  
 Open Development Platform  
 AVL List GmbH  
 +43 316 787 7345  
 katrin.moser@avl.com



Kontakt

Josef Zehetner  
 Chief Engineer System Architecture,  
 Integrated and Open Development  
 Platform  
 AVL List GmbH  
 +43 316 787 1765  
 josef.zehetner@avl.com



prostep ivip Web-Seminare

**Austausch - Information - Inspiration**

Ein weiterer Service des prostep ivip Vereins für seine Mitglieder\*: die kostenlose Teilnahme am monatlichen Web-Seminar! Die Veranstaltungsreihe findet zumeist am letzten Freitag eines Monats statt und besteht aus einer halbstündigen Präsentation; im Anschluss erfolgt eine Diskussionsrunde innerhalb der internationalen Teilnehmergruppe.

Mehr Informationen zu den Web-Seminaren und die Möglichkeit zur Anmeldung erhalten Sie bei Sarah.Giese@prostep.com bzw. im Netz unter [www.prostep.org](http://www.prostep.org).

*\*Die Web-Seminare stehen teilweise auch Nicht-Mitgliedern offen*



**Die Termine für 2022:**

**30. September 2022** DDP  
 Nur für Mitglieder

**28. Oktober 2022** New Project Proposals  
 offen

**25. November 2022** ReqIF  
 offen