


# Wenn Software Software schreibt

04/08/2021 Software wird immer mehr zum bestimmenden Faktor in der Automobilentwicklung. Marius Mihailovici, Geschäftsführer von Porsche Engineering Romania, wirft in seinem Beitrag einen Blick in die Zukunft der Softwareentwicklung und erklärt, warum sich der Job des Programmierers in den kommenden 20 Jahren komplett verändern könnte.

Autos sind schon heute rollende Computer. In ihnen arbeitet ein Netzwerk aus Dutzenden Recheneinheiten: Zwischen 70 und 100 dieser Electronic Control Units (ECUs) sind in modernen Fahrzeugen verbaut. ECUs steuern die Kraftstoffeinspritzung, regeln das Bremsverhalten oder uberwachen die Klimaanlage. Der nachste Schritt werden HCPs (High Performance Computing Platforms) sein, die deutlich mehr Rechenleistung in einem Steuergerat ermoglichen.

Die hohere Rechenleistung und die Integration sind notwendig, da die Anzahl der Codezeilen und die Komplexitat der Funktionen im Fahrzeug von Jahr zu Jahr ansteigen. Eine Zahl mag das verdeutlichen: 100 Millionen. So viele Zeilen Code stecken in einem modernen Auto. Zum Vergleich: Eine Boeing 787 Dreamliner bringt es gerade einmal auf 14 Millionen Zeilen. All diese unterschiedlichen Herausforderungen fuhren dazu, dass schnelles und flexibles Arbeiten gefragt ist. Ziele werden oft kurzfristig gesetzt und andern sich schnell. Man kann sagen: Software wird ergebnis getrieben

entwickelt.

In Cluj arbeiten unsere Softwareentwickler mit Tools, die sich seit 20 Jahren in der Softwareentwicklung bewahrt haben. So nutzen wir – wo immer moglich – agile Methoden. Sie beruhen auf kleinen Entwicklungsschritten, deren Ergebnisse in taglich stattfindenden Feedbackrunden uberpruft werden. Die einzelnen Teams haben dabei große Freiheit und arbeiten direkt miteinander, um sich mit ihrer jeweiligen Kompetenz unterstutzen zu konnen. Sie legen ihre Ziele taglich und flexibel selbst fest. Teamleiter haben hier oft nur noch moderierende Funktion und behalten das große Ganze im Auge.

Auch hinter dem Entertainmentsystem und der Navi gation stehen viele Zeilen Code. Hinzu kommt die Moglichkeit, Smartphones und andere Gerate mit dem Auto zu verbinden, was ebenfalls nur durch komplexe Software moglich ist. Und dabei bleibt es nicht. Software ubernimmt immer weitere wichtige Aufgaben im Auto. Zu den wichtigsten heutigen und kunftigen Funktionen gehoren der Datenaustausch mit anderen Verkehrsteilnehmern und der Infrastruktur, das Update der Fahrzeuge aus der Cloud und schließlich sogar das autonome Fahren.

Dabei findet die Entwicklung von Automotive-Software in einem besonderen Spannungsfeld statt. Es gilt, Sicherheitsbestimmungen und Kundenvorgaben in Form großer Lastenhefte zu erfullen. Die klassischen Entwicklungsprozesse der Branche sind in der Regel zeitgetrieben: Es gibt einen vorgegebenen Zeitplan, der bestimmte Meilensteine definiert. Unsere Auftrag geber erwarten regelmaßige Ergebnisse zu vorher festgelegten Terminen. Hinzu kommen die behordlichen Zulassungsprozesse, die sogenannte Homologation. Sie sieht zum Beispiel vor, dass Fahrzeuge erst auf den Markt kommen, wenn sie in einer bestimmten Stuckzahl gebaut und abgenommen worden sind. Letztendlich geht es in der Branche immer darum, sich von einem „fertigen" Zustand zum nachsten zu bewegen.

Ein weiteres aktuelles Paradigma der Software entwicklung ist „Continuous Integration". Darunter versteht man ein stark automatisiertes Verfahren, mit dem Softwareelemente am Ende eines Arbeitstages auf ihre Lauffahigkeit uberpruft und in das Gesamtsystem integriert werden. So konnen wir Fehler und Probleme schnell erkennen und beheben. Der Einsatz agiler Methoden und von Continuous Integration sorgt fur mehr Effizienz und bietet auch unseren Kunden einen Mehrwert: Er erleichtert die Prasentation von Zwischenergebnissen und gibt ihnen die Moglichkeit fur schnelles Feedback.

Fur die kommenden funf Jahre gehe ich von einem zunehmenden Einsatz der oben beschriebenen Methoden aus. Wir werden noch starker ergebnis- und nicht zeitgetrieben arbeiten. Teamhierarchien durften an Bedeutung verlieren, und ich sehe einen hoheren Grad an flachen und selbstorganisierenden Teams mit klaren Verantwortlichkeiten voraus. Zudem wird es wohl selbstverstandlich sein, flexibel von verschiedenen Orten aus zu arbeiten (gemaß dem Konzept: „Einstellen, wo die Kompetenz existiert"). In den nachsten Jahren werden wir außerdem eine zunehmende Automatisierung sehen: Immer haufiger wird Software von Software getestet und nicht mehr von Menschen. Manuelle Tests werden vollstandig verschwinden.

## Die Grenzen zwischen Auto und Umwelt verschwimmen

Entwicklungen außerhalb der Automobilbranche werden uns ebenfalls zum Umdenken zwingen, denn Autos sind zunehmend ins digitale Leben ihrer Fahrer integriert. So verbindet sich zum Beispiel das Smartphone beim Einsteigen automatisch mit dem Fahrzeug. Mediennutzung, Navigation und Kommunikation gehen nahtlos ineinander uber.

Es sind solche Nutzungsszenarien, die die Arbeit unserer Softwareentwickler inhaltlich verandern werden – einfach, weil die Grenzen zwischen der reinen „Autosoftware“ und anderen Anwendungen verschwimmen. Das erfordert ubrigens auch ein bestimmtes Mindset bei unseren Entwicklern: Bei uns arbeiten Menschen, die selbst den digitalen Lifestyle leben. Sie wissen nicht nur, was unsere Auftraggeber fordern, sondern auch, was die Fahrzeugnutzer von ihrem Auto erwarten.

## ECUs der Zukunft beeinflussen die Softwareentwicklung

Auch neue Steuergerate-Architekturen verandern die Art, wie Automotive-Software entwickelt wird. Ich gehe davon aus, dass es im Auto der Zukunft einige wenige zentrale, sehr leistungsfahige Rechner vom HCP-Typ gibt, die zusammen mit nachgeordneten, einfacheren Einheiten das gesamte Fahrzeug steuern. Auf diesen zentralen Einheiten laufen auch alle Anwendungen, die uber Basisfunktionen hinausgehen, also zum Beispiel Entertainment, Datenverkehr oder die Kommunikationsanwendungen der Insassen. Diese zentralen Rechner mit einem echten Betriebssystem machen das Auto dann tatsachlich zu einem PC auf vier Radern.

Fur Entwickler heißt das, dass sich die Methoden ihrer Arbeit gar nicht so sehr verandern. Was sich andert, sind die Systeme, mit denen sie es zu tun haben: Sie sind hierarchischer, haben weniger Komponenten und werden von einer Gesamtsoftware gesteuert. Und diese wird – wie jede Software – regelmaßige Updates erhalten. Die Software eines Autos wird also nicht mehr einmal entwickelt und installiert, sondern standig weiterentwickelt, auch wenn das Auto schon langst beim Kunden ist.

## Eine Zukunft ohne Softwareentwickler?

Die Frage ist gar nicht so abwegig: Wird es in Zukunft uberhaupt noch Softwareentwickler geben? Manche Experten gehen davon aus, dass Kunstliche Intelligenz (KI) in den kommenden zwei Jahrzehnten die Softwareentwicklung komplett ubernehmen wird – auch in der Automobilbranche. Allerdings gehen die Meinungen hieruber auseinander. Ich personlich kann mir vorstellen, dass wir weiterhin die Rahmenbedingungen vorgeben und die KI diese dann umsetzt. Wir werden immer noch Software-Architekten, Anforderungsingenieure und Software-Ingenieure brauchen, um zu definieren, was KI beziehungsweise neuronale Netze tun werden: Softwarefunktionen auf der Grundlage der definierten Anforderung generieren, kontinuierlich automatisch testen und korrigieren, bis die Softwarequalitat auf dem erwarteten Niveau ist.

Die Entwickler der Zukunft mussen darum in der Lage sein, in Gesamtsystemen zu denken. Sie mussen wissen, wie Endkunden ihre Fahrzeuge nutzen. Ihre Arbeit wird nicht nur von den Vorgaben der OEMs bestimmt, sondern immer starker von den Konsumenten. Eines allerdings kann ich mir nicht vorstellen: eine Software ganz ohne „Bugs". Fehler wird es immer geben. Wir mussen sie nur nicht mehr selbst finden und beheben: Das kunftige neue System wird das erledigen, uberwacht von menschlichen Softwareexperten.

## Marius Mihailovici

Marius Mihailovici ist seit 2016 Geschaftsfuhrer von Porsche Engineering Romania. Zuvor leitete er Forschung und Entwicklung bei Alcatel-Lucent S.A./Nokia Oyj, wo er auch als Software-Manager fur 2G- und 3G-Projekte tatig war. Bei World Telecom arbeitete er als Ingenieur fur Nachrichtentechnik.

## Info

Text: Marius Mihailovici

Text erstmalig erschienen im Porsche Engineering Magazin, Nr. 1/2021

## MEDIA ENQUIRIES



**Frederic Damköhler**

Senior Manager Corporate Communications Porsche Engineering
+49 (0) 711 / 911 16361
frederic.damkoehler@porsche.de

### Verbrauchsdaten

**Taycan Turbo S (2023)**
Kraftstoffverbrauch / Emissionen

WLTP*
Stromverbrauch* kombiniert (WLTP) 23,4 – 22,0 kWh/100 km
CO-Emissionen* kombiniert (WLTP) 0 g/km
CO2-Klasse A Klasse

*Weitere Informationen zum offiziellen Kraftstoffverbrauch und den offiziellen spezifischen CO-Emissionen neuer Personenkraftwagen können dem „Leitfaden über den Kraftstoffverbrauch, die CO-Emissionen und den Stromverbrauch neuer Personenkraftwagen“ entnommen werden, der an allen Verkaufsstellen und bei DAT (Deutsche Automobil Treuhand GmbH, Helmuth-Hirth-Str. 1, 73760 Ostfildern-Scharnhausen, www.dat.de) unentgeltlich erhältlich ist.

## Linksammlung

Link zu diesem Artikel
https://newsroom.porsche.com/de/2021/innovation/porsche-engineering-wenn-software-software-schreibt-23325.html

Externe Links
https://www.porscheengineering.com/peg/de/

**Dr. Ing. h.c. F. Porsche Aktiengesellschaft**
Porscheplatz 1
70435 Stuttgart