

Wenn Software Software schreibt

Dieser Artikel gibt den Stand zum Zeitpunkt seiner Veröffentlichung wieder. Der Inhalt kann inzwischen veraltet sein.

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 u□berwachen die Klimaanlage. Der na□chste Schritt werden HCPs (High Performance Computing Platforms) sein, die deutlich mehr Rechenleistung in einem Steuergera□t ermo□glichen.

Die ho□here Rechenleistung und die Integration sind notwendig, da die Anzahl der Codezeilen und die Komplexita□t 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 fu×hren dazu, dass schnelles und flexibles Arbeiten gefragt ist. Ziele werden oft kurzfristig gesetzt und a×ndern 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 bewa×hrt haben. So nutzen wir – wo immer mo×glich – agile Methoden. Sie beruhen auf kleinen Entwicklungsschritten, deren Ergebnisse in ta×glich stattfindenden Feedbackrunden u×berpru×ft werden. Die einzelnen Teams haben dabei große Freiheit und arbeiten direkt miteinander, um sich mit ihrer jeweiligen Kompetenz unterstu×tzen zu ko×nnen. Sie legen ihre Ziele ta×glich 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 Mo×glichkeit, Smartphones und andere Gera×te mit dem Auto zu verbinden, was ebenfalls nur durch komplexe Software mo×glich ist. Und dabei bleibt es nicht. Software u×bernimmt immer weitere wichtige Aufgaben im Auto. Zu den wichtigsten heutigen und ku×nftigen Funktionen geho×ren 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 erfu×llen. Die klassischen Entwicklungsprozesse der Branche sind in der Regel zeitgetrieben: Es gibt einen vorgegebenen Zeitplan, der bestimmte Meilensteine definiert. Unsere Auftrag geber erwarten regelma×ßige Ergebnisse zu vorher festgelegten Terminen. Hinzu kommen die beho×rdlichen Zulassungsprozesse, die sogenannte Homologation. Sie sieht zum Beispiel vor, dass Fahrzeuge erst auf den Markt kommen, wenn sie in einer bestimmten Stu×ckzahl gebaut und abgenommen worden sind. Letztendlich geht es in der Branche immer darum, sich von einem „fertigen“ Zustand zum na×chsten 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 Lauffa×higkeit u×berpru×ft und in das Gesamtsystem integriert werden. So ko×nnen wir Fehler und Probleme schnell erkennen und beheben. Der Einsatz agiler Methoden und von Continuous Integration sorgt fu×r mehr Effizienz und bietet auch unseren Kunden einen Mehrwert: Er erleichtert die Pra×sentation von Zwischenergebnissen und gibt ihnen die Mo×glichkeit fu×r schnelles Feedback.

Fu×r die kommenden fu×nf Jahre gehe ich von einem zunehmenden Einsatz der oben beschriebenen Methoden aus. Wir werden noch sta×rker ergebnis- und nicht zeitgetrieben arbeiten. Teamhierarchien du×rften an Bedeutung verlieren, und ich sehe einen ho×heren Grad an flachen und selbstorganisierenden Teams mit klaren Verantwortlichkeiten voraus. Zudem wird es wohl selbstversta×ndlich sein, flexibel von verschiedenen Orten aus zu arbeiten (gema×ß dem Konzept:

„Einstellen, wo die Kompetenz existiert"). In den na□chsten Jahren werden wir außerdem eine zunehmende Automatisierung sehen: Immer ha□ufiger wird Software von Software getestet und nicht mehr von Menschen. Manuelle Tests werden vollsta□ndig 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 u□ber.

Es sind solche Nutzungsszenarien, die die Arbeit unserer Softwareentwickler inhaltlich vera□ndern werden – einfach, weil die Grenzen zwischen der reinen „Autosoftware" und anderen Anwendungen verschwimmen. Das erfordert u□brigens 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 Steuergera□te-Architekturen vera□ndern die Art, wie Automotive-Software entwickelt wird. Ich gehe davon aus, dass es im Auto der Zukunft einige wenige zentrale, sehr leistungsfa□hige Rechner vom HCP-Typ gibt, die zusammen mit nachgeordneten, einfacheren Einheiten das gesamte Fahrzeug steuern. Auf diesen zentralen Einheiten laufen auch alle Anwendungen, die u□ber Basisfunktionen hinausgehen, also zum Beispiel Entertainment, Datenverkehr oder die Kommunikationsanwendungen der Insassen. Diese zentralen Rechner mit einem echten Betriebssystem machen das Auto dann tatsa□chlich zu einem PC auf vier Ra□dern.

Fu□r Entwickler heißt das, dass sich die Methoden ihrer Arbeit gar nicht so sehr vera□ndern. Was sich a□ndert, 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 – regelma□ßige Updates erhalten. Die Software eines Autos wird also nicht mehr einmal entwickelt und installiert, sondern sta□ndig weiterentwickelt, auch wenn das Auto schon la□ngst beim Kunden ist.

Eine Zukunft ohne Softwareentwickler?

Die Frage ist gar nicht so abwegig: Wird es in Zukunft u□berhaupt noch Softwareentwickler geben? Manche Experten gehen davon aus, dass Ku□nstliche Intelligenz (KI) in den kommenden zwei Jahrzehnten die Softwareentwicklung komplett u□bernehmen wird – auch in der Automobilbranche. Allerdings gehen die Meinungen hieru□ber auseinander. Ich perso□nlich 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 Softwarequalita□t auf dem erwarteten Niveau ist.

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

Marius Mihailovici

Marius Mihailovici ist seit 2016 Gescha□ftsfu□hrer von Porsche Engineering Romania. Zuvor leitete er Forschung und Entwicklung bei Alcatel-Lucent S.A./Nokia Oyj, wo er auch als Software-Manager fu□r 2G- und 3G-Projekte ta□tig war. Bei World Telecom arbeitete er als Ingenieur fu□r 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 (Vorgängermodell)

*Weitere Informationen zum offiziellen Kraftstoffverbrauch und den offiziellen spezifischen CO_2-Emissionen neuer Personenkraftwagen können dem „Leitfaden über den Kraftstoffverbrauch, die CO_2-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.

Bildunterschriften

Pfad: Wenn Software Software schreibt/Bilder/Bild_1.jpg
Titel: Marius Mihailovici, Geschaftsfuhrer von Porsche Engineering Romania, 2021, Porsche AG
Bildunterschrift: Marius Mihailovici, Geschäftsführer von Porsche Engineering Romania

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