

Bitkom AK Software Engineering & Software Architektur mit
AK Automatisiertes, vernetztes & autonomes Fahren

Datum: 26. November 2024 | Ort: Webkonferenz

Code on the Road: Softwaregestützte Mobilität im Fokus

Der fortwährende Wandel in der Mobilitätsbranche manifestiert sich auf vielen Ebenen: Fahrzeuge entwickeln sich von einem Netzwerk von Embedded-Steuergeräten hin zu hochkomplexen Rechenzentren mit anspruchsvollen Steuerungs- und Vernetzungsaufgaben - ohne einen Administrator im Kofferraum. Auf der Ebene der kooperativen Mobilität wird die Vernetzung der Fahrzeuge untereinander und mit der Infrastruktur die Softwarekomplexität weiter erhöhen.

In einem Online-Event haben wir diese verschiedenen Veränderungen aus den Perspektiven Softwarearchitektur und Software Engineering beleuchtet.

1. Software Defined Vehicle: Open-Source-Projekte im Fokus

Felix Mölders, d-fine GmbH

In diesem Vortrag wurden ausgewählte Open-Source-Projekte zum Software Defined Vehicle vorgestellt, die die Entwicklung aktuell maßgeblich vorantreiben. Im Fokus stehen die Projekte Eclipse Ankaios, Eclipse eCAL und Automotive Grade Linux (AGL). Ankaios agiert als leistungsstarker Workload- und Container-Orchestrierer für Automotive High Performance Computing, der komplexe Workload-Setups effizient auf einer zentralen Architektur verwaltet. eCal ist eine brokerless Middleware, die die Integration von Softwarekomponenten durch nahtlose Kommunikation vereinfacht. AGL basiert auf dem Yocto Project und ist die wegweisende Lösung für Infotainment und Cluster-Instrumentation. Alle drei Projekte veranschaulichen die zunehmende Kooperation zwischen Zulieferern, Automotive-OEMs und IT-Unternehmen.

Key Takeaways

- Software Defined Vehicle (SDV) ist ein relativ neues Thema. Eine Einordnung von Elektrobit bietet Reifegrad-Levels.
- Zentrale Punkte von SDVs sind die Update-Fähigkeit und die Möglichkeit, Funktionen später digital nachzurüsten (Upgrade).

- Dazu müssen jedoch Kapazitäten in der Hardware verbaut werden, die zur Auslieferungszeit (noch nicht) genutzt werden.
- Open-Source-Projekte dienen der Standardisierung der Software »unter der Haube«. Anders als bisher wird hier der Code-First-Ansatz verfolgt (nicht Concept-First wie bisher z. B. bei AUTOSAR).
- Eine Zusammenarbeit in Open-Source-Projekten ist die einzige Möglichkeit der herstellerübergreifenden Zusammenarbeit, ohne in kartellrechtliche Probleme zu laufen.
- Das Problem der Zertifizierung der Open-Source-Software für sicherheitskritische Anwendungen ist noch ungelöst. Idee ist eine gemeinsame Zertifizierung über eine OSS-Organisation wie Eclipse.

2. Transforming Automotive: AI – The New Engine of Disruption in the Automotive Industry

Dr. Jan Wehinger, MHP – A Porsche Company

Der Schwerpunkt dieses Vortrags lag auf der Frage, was ein Software Defined Vehicle ist und wie künstliche Intelligenz die gesamte Wertschöpfungskette der Automobilindustrie verändern kann – von der Entwicklung über die Produktion bis hin zu Anwendungen im Fahrzeug. Zentral sind hierbei die Erwartungen und Vertrauensfragen von KI-Nutzern in Europa, China und den USA. Es wurde dargelegt, welche Funktionen die Nutzer überzeugen und zum Kauf motivieren und welchen Anbietern sie am meisten vertrauen.

Key Takeaways

- Im Zuge der Transformation zum Software Defined Vehicle werden aktuell bei OEMs und Zulieferern Prozesse, Methoden und Tools komplett überarbeitet und verändert.
- Die Kunden haben immer höhere Erwartungen an die Fahrzeuge: Virtual Reality, Augment Reality, Sprachassistenten, usw.
- Ein Auto ist kein »Fun Device«. Es ist üblicherweise die zweitgrößte Anschaffung eines Menschen.
- Fahrzeughersteller müssen aktuell gleichzeitig die Digitalisierung des Unternehmens (»Software Driven«) und des Produkts (»Software Defined«) vorantreiben.
- Kunden in Europa und den USA sind eher skeptisch gegenüber KI. Kunden in China sind sehr viel offener. KI ist dort ein Faktor bei der Kaufentscheidung.
- Die Bereitschaft, für KI zu bezahlen besteht weltweit fast nur beim Erwerb des Fahrzeugs. Pay-as-you-drive wird i.d.R. nicht akzeptiert.
- Wichtiger Blick: Start vom Customer Need, nicht vom möglichen AI Case.

3. Schneller und effizienter im Entwicklungsprozess: Eine Toolbox zur Schließung von Toolbrüchen im Softwareentwicklungsprozess

Michael Meitz, e:fs TechHub GmbH

In diesem Vortrag wurde eine Toolbox vorgestellt, die im Softwareentwicklungsprozess bei einem Automobilhersteller im Teilbereich Funktionale Sicherheit der Lenkungsentwicklung eingesetzt wurde. Mit dieser Toolbox konnten Toolbrüche innerhalb von nur einer Woche behoben werden. Dies führte zu einer signifikanten Reduzierung der Kosten sowie einer Verringerung menschlicher Fehler. Darüber hinaus wurde die Bereitstellung von Ergebnissen beschleunigt. Insgesamt trugen diese Maßnahmen zur Optimierung der Effizienz und Qualität im Entwicklungsprozess bei.

Key Takeaways

- Typische Entwicklungsumgebungen enthalten zahlreiche, komplexe Tools, die nicht miteinander interagieren.
- Toolbrüche sind ein großes Problem bei Fehlererkennung und -bearbeitung
- Bisher wurden mit viel Aufwand Schnittstellen realisiert, die nicht mehr genutzt werden, sobald die Entwickler das Entwicklungsteam verlassen.
- Mit einem Orchestrierungswerkzeug, das die einzelnen Schnittstellen integriert und einen zentralen Blick auf die Tool- und Schnittstellenwelt bietet, kann diesem Problem entgegengetreten werden.

4. Diskussionsrunde: »Code on the Road: Softwaregestützte Mobilität im Fokus«

Moderation: **Vorstand des AK Software Engineering & Software Architektur und Vorstand des AK Automatisiertes, vernetztes & autonomes Fahren**

In der Diskussionsrunde wurden offene Themen aufgegriffen, die während der Vorträge aufgekommen sind. Unter anderem wurden Fragen zum Software Defined Vehicle, der Standardisierung von Software und den damit einhergehenden Fragestellungen im Zusammenspiel der verschiedenen Akteure besprochen.

Key Takeaways

- Betrachtet werden muss nicht nur die Software im Fahrzeug, sondern auch die wachsende Softwarewelt in den Backends der Hersteller.
- Die Harmonisierung der Softwarearchitektur(en) in Fahrzeug und Backend ist schon innerhalb der Welt eines Herstellers eine große Herausforderung, Standardisierungen über Herstellergrenzen hinaus sind jedoch nur in wenigen Bereichen erfolgreich. So gibt es innerhalb der COVESA Beispiele für erfolgreich standardisierte Schnittstellen im V2X-Segment.

- Eine große Herausforderung für die Branche ist der mit dem Software Defined Vehicle einhergehende Wechsel von einer Einmalentwicklung für ein Produkt hin zur kontinuierlichen Weiterentwicklung der Software über den Lebenszyklus eines Produkts oder gar einer kompletten Produktlinie.

Zusammenfassung

Die Arbeitskreis-Sitzung »Code on the Road: Softwaregestützte Mobilität im Fokus« zeigte, dass sich in den letzten Jahren die Anzahl der Funktionen im Fahrzeug durch Fahrerassistenz, Vernetzung, Elektromobilität u.v.m. die Komplexität der Software innerhalb und außerhalb des Fahrzeugs immer weiter erhöht hat. Nun kommt mit dem Software Defined Vehicle noch die zeitliche Dimension hinzu: Die Softwareentwicklung endet nicht mehr mit dem Marktstart des Fahrzeugmodells, sondern muss begleitend über alle Fahrzeugprojekte über den gesamten Lebenszyklus der Fahrzeuge hinaus erfolgen und die damit einhergehende Varianz abbilden.

Die Standardisierung von Software und Architekturen über Herstellergrenzen bleibt weiterhin eine große Herausforderung für die Branche. Standardisierungsbemühungen laufen in verschiedenen Organisationsformen bis hin zu Open-Source-Projekten.



Dr. Frank Termer
Bereichsleiter Software
T 030 27576-232
f.termer@bitkom.org