Automotive Software Factory

Software Architecture for safety critical brake and steering systems

Dr. Ingo Alfter, ZF DIV-A ADWC / 22. September 2021

Author

Dr. Ingo Alfter is Chief Engineer at ZF Group Division A Active & Passive Safety Technology.

He is responsible for the Customer Product Platform Software Development of Braking and Steering Systems within ZF Group Division A.

He has been with ZF for the last 21 years and has held multiple managerial positions within ZF TRW.

Dr. Ingo Alfter holds a Diploma and a PhD in Physics from the University of Bonn.

Ingo.Alfter@ZF.com

Agenda

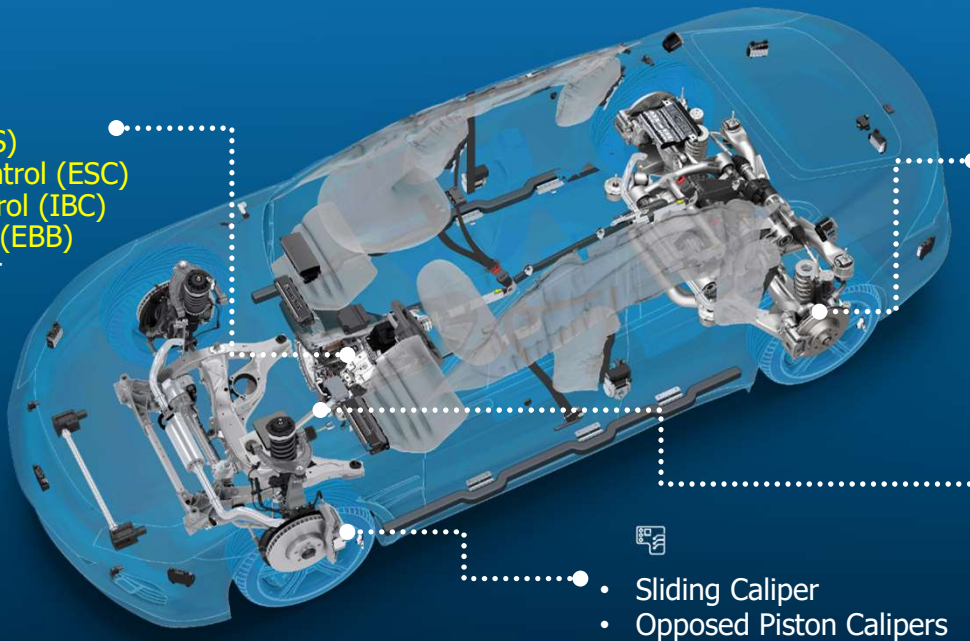
1. ZF Division A
2. Market Trends OEM E/E Architectures
3. Software Development

   Platform Concept
4. Summary

ZF Division A
Active Safety

ZF Division A – Active Safety – Portfolio for safety systems

- Anti-Lock Braking (ABS)
- Electronic Stability Control (ESC)
- Integrated Brake Control (IBC)
- Electric Brake Booster (EBB)
- Vacuum Brake Booster



- Electric Park Brake Caliper (EPB)
- Mechanical Integrated Park Brake Caliper (IPB)
- Drum brake
- DIH

- Electric Power Steering (EPS)
  - Column Drive (CD-EPS)
  - Belt Drive (BD-EPS)
  - Dual Pinion (DP-EPS)

- Sliding Caliper
- Opposed Piston Calipers

Electrified Powertrain : Vehicle Motion Control: Automated Driving: Integrated Safety:

Market Trends
OEM E/E Architectures

Future vehicle architectures – Domain- or Zone-based

Software-defined car

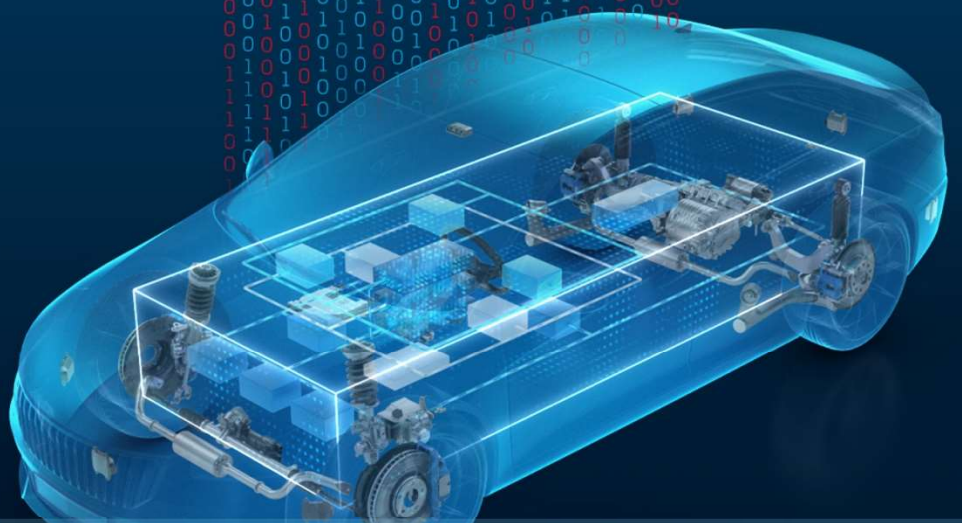
Hardware

Smart actuators

Intelligent Sensors

More processing power

Domain- or Zone-based E/E architectures



Software

Hardware independence

Flexibility

Scalability

Compatibility

Upgradability

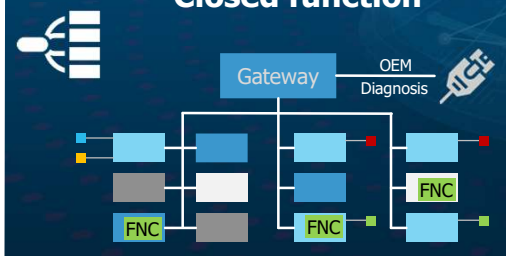
Comfort and combined handling functions may move from actuators into Domain- or Zone-based controllers with OEM DNA

Functional architecture in future E/E OEM platforms

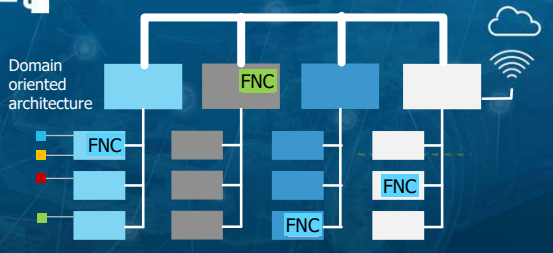
Increasing no. of software & complexity

General Trend
- Trend to become hardware independent
- Software (function) driven innovations → OEM DNA
- Functions on demand (pay per use)

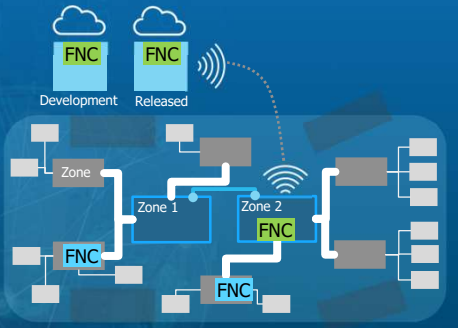
Closed function



Domain Oriented



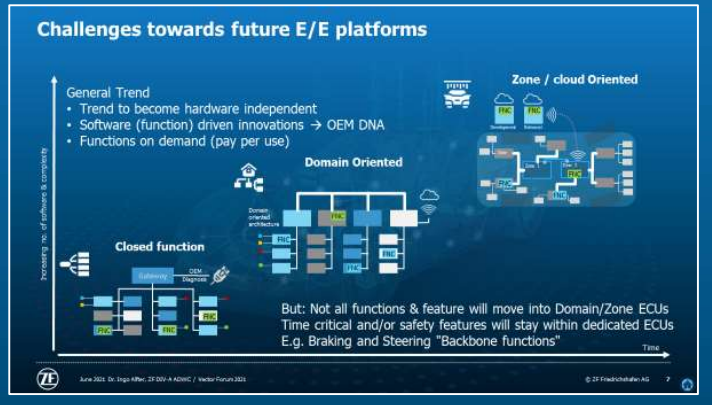
Zone / cloud Oriented



But: Not all functions & features will move into Domain / Zone ECUs
Time critical and / or safety features will stay within dedicated ECUs
E.g. Braking and Steering "Backbone functions"

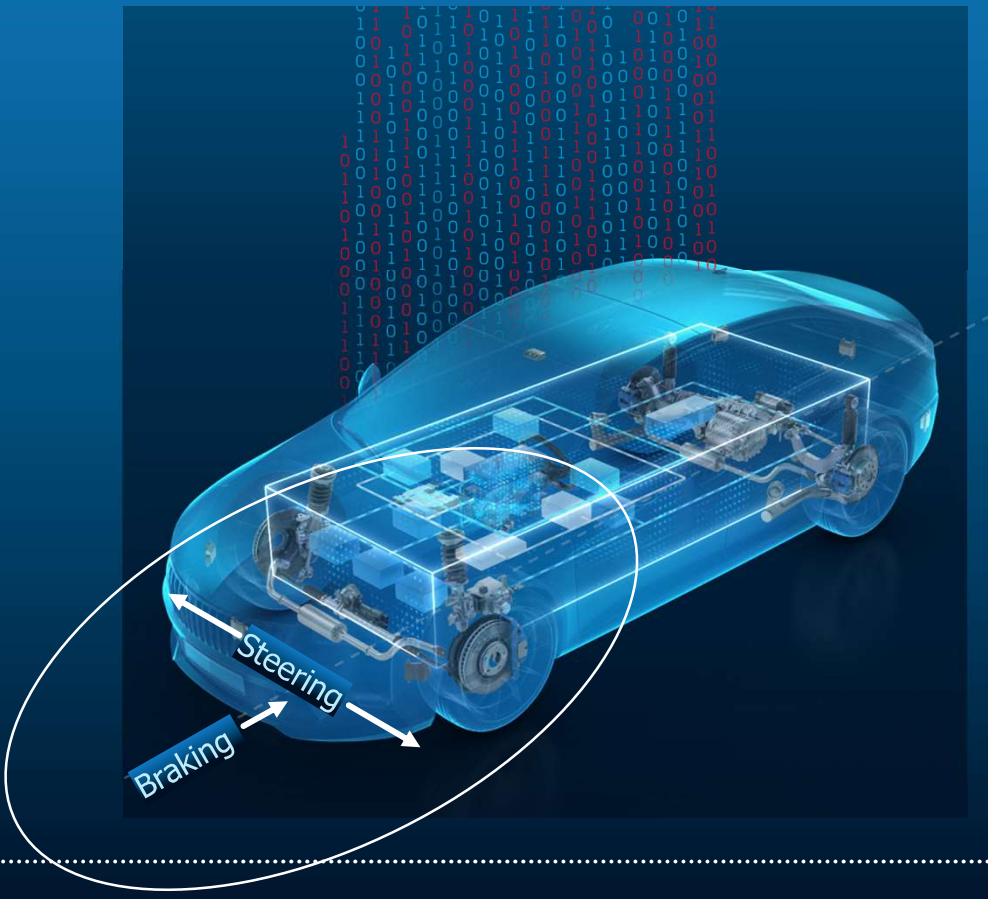
Time

Vehicle E/E Trends – Effect on Safety functions



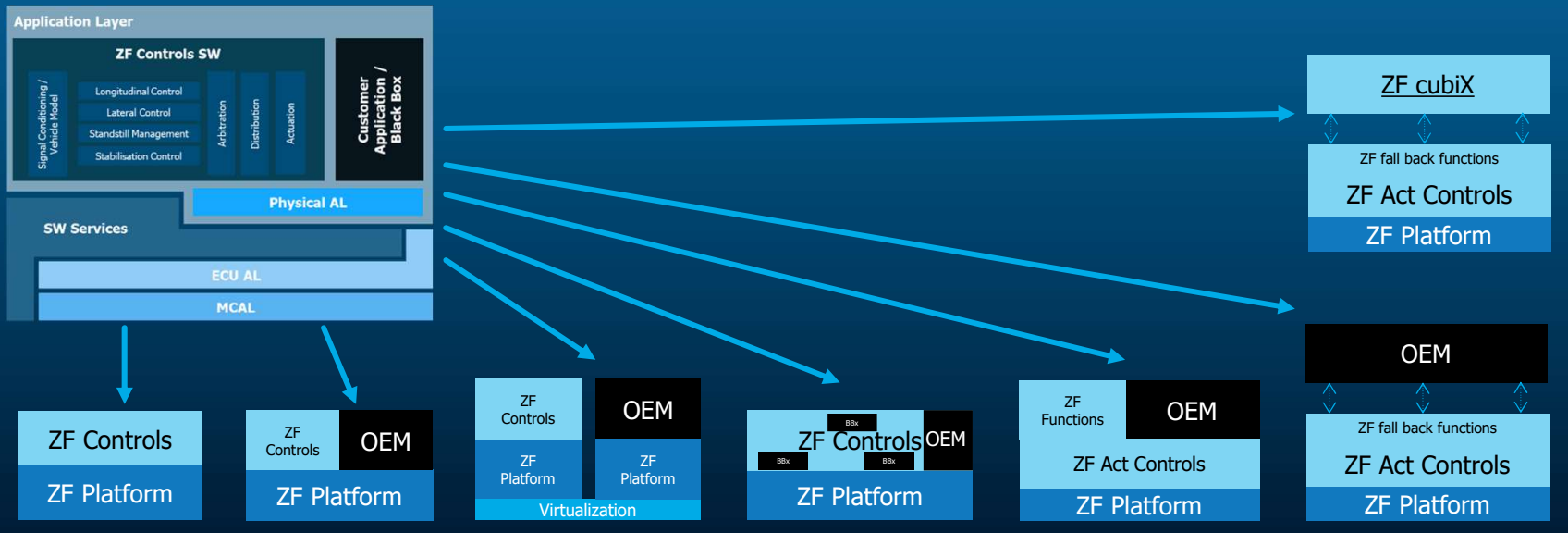
Braking and Steering ECUs with primary Safety functions
- Torque Control
- Force Control

at the wheels still require dedicated ECUs with Safety µC



Conclusion

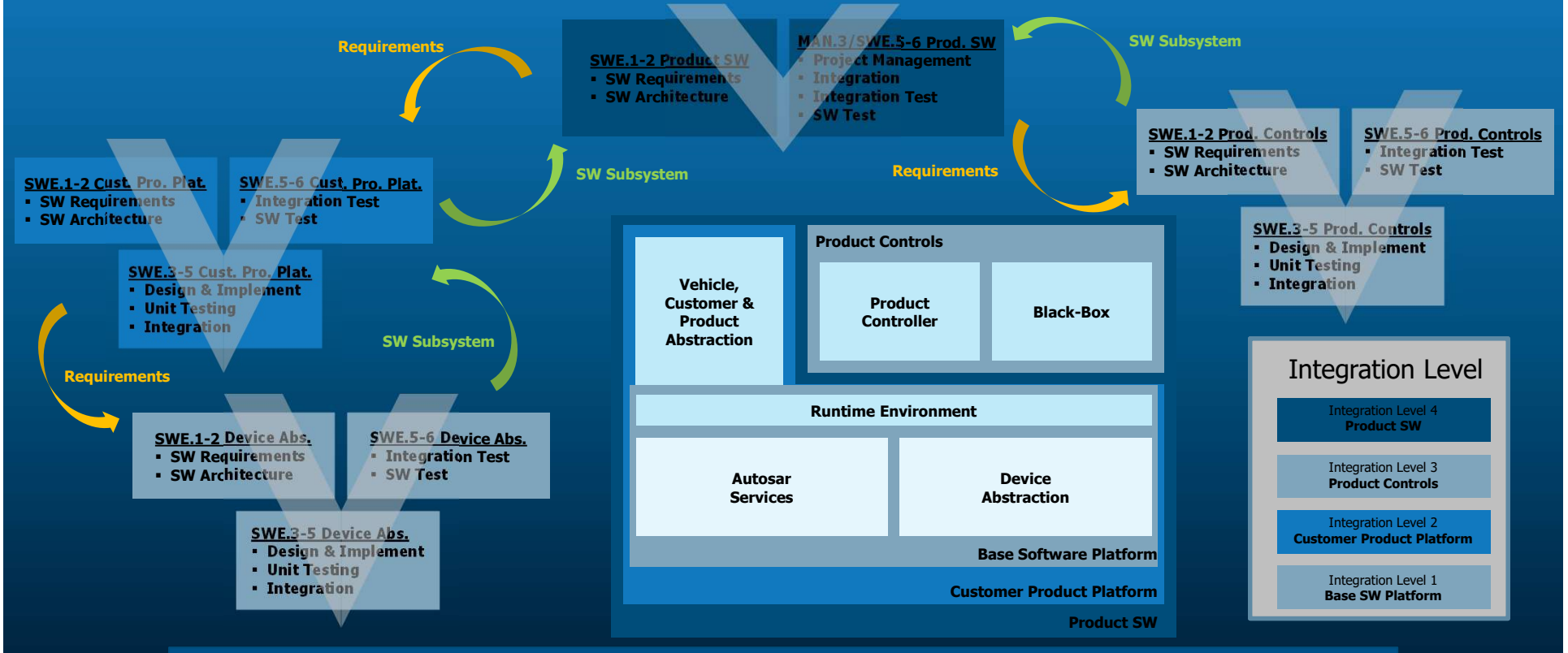
- Braking and steering systems remains primary safety actuator in all vehicle architectures
- Support of "classical" One ECU software architecture
- As well as Domain- / Zone-based architecture with distributed functions
- Scalable E/E architectures are required today and even more in the future



Software Development Platform Concept

„Automotive Software Factory“ @ ZF DIV-A

Stepwise Software Development – Platform Concept



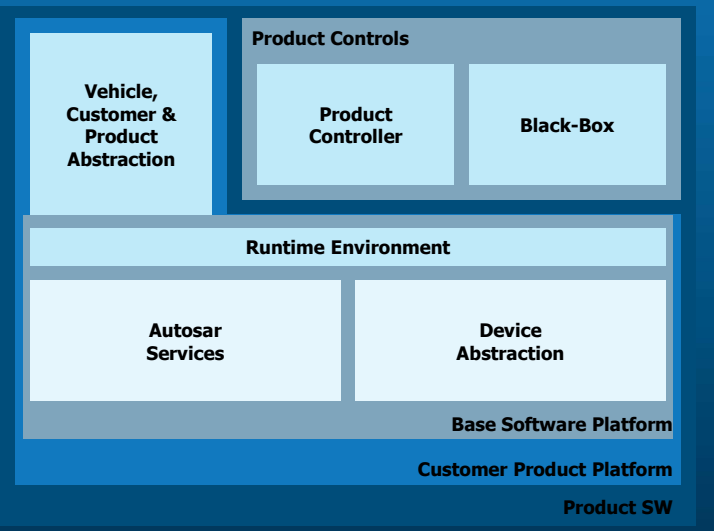
Development flow enables product/customer specific solution maintaining re-use and stepwise integration to ensure agility and quality

Stepwise Software Development – Platform Concept

Base Software Platform:
- Agile development with ~4 weeks sprints
- Planned releases into Customer Product Platform
- Dedicated Base Software platform teams for steering and braking products

Customer Product Platform
- Between 8 weeks and 3 month integration cycles
- Timing of integration cycles in alignment with OEM development cycles
- Dedicated customer software development teams

Product Controls:
- Annual core releases with service pack releases in between
- Planned releases into Product Software
- Core Function & Control development teams



Product Software:
- Continues integration with daily builds and 1-2 weeks release timings
- Dedicated customer software development teams with dedicated software project leads

Agile Approach – Phase 0 and 1

Project Phase 0 and 1 – "Setup Project" and "make it run"

-> agile method: *Kanban*

-> distributed team location, work from home

-> virtual meeting room with online Kanban-Board, daily Stand-up meetings

Target: Define/create stable software baseline and tool environment for a project start

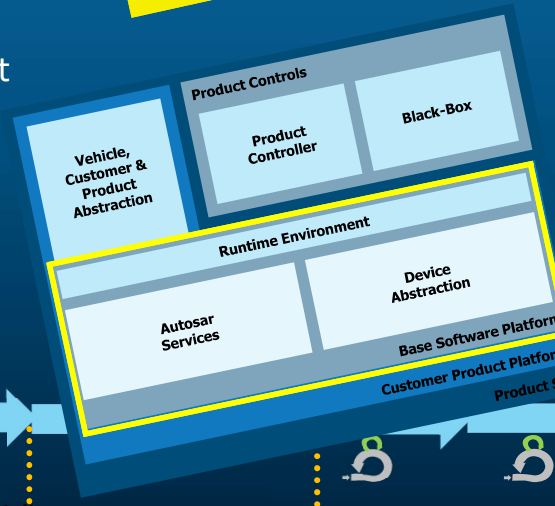
Timing: expected within ~ 2month

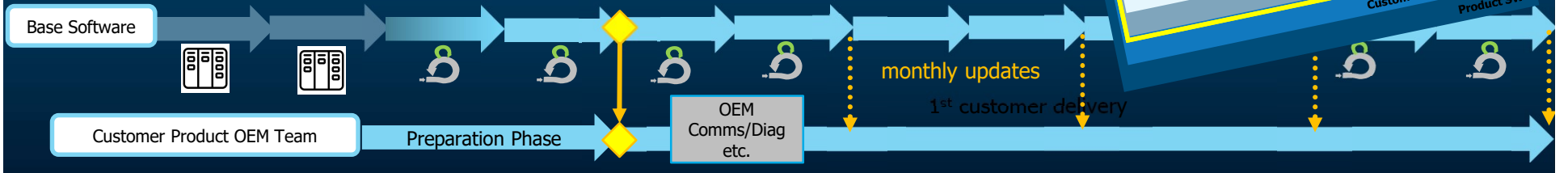
Definition of done: task counters running

Involved:

Base software platform team

Basic Setup
- Configure baseline checkpoints
- AUTOSAR SIP
- First low level control integration
- Core distribution implemented
- FlexRay/CAN and OSTask system
- First task counters running
- Setup Rhapsody, Vcast, SwRS etc.





Agile Approach – Phase 2

Project Phase 2 – "Bring in content and quality"

-> agile method: *Scrum (maybe scaled; parallel Teams in India)*

-> distributed team location, work from home

-> virtual meeting room with online Kanban-Board, daily Stand-up meetings

-> Continuous integration and software builds

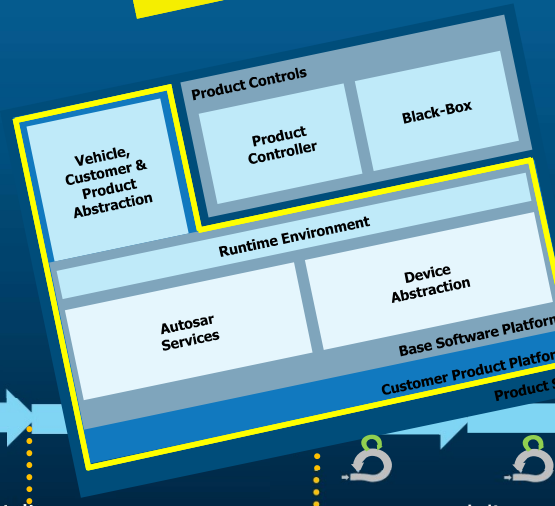
Target: Provide Software for next release

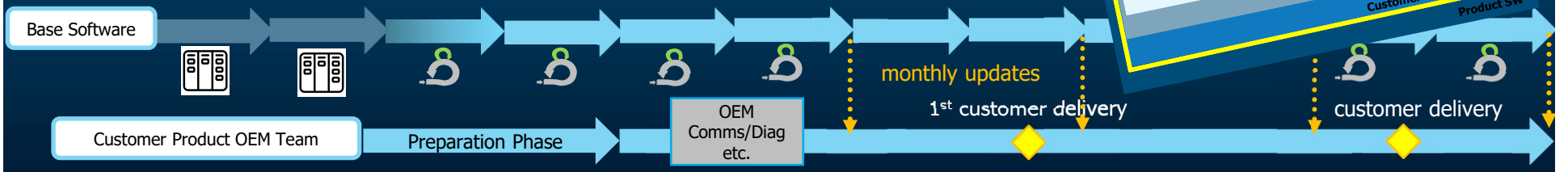
Timing: detailed planning towards next customer Milestone
feature based planning (rollout plan) towards 1st SOP

Definition of done: Content for customer milestone is integrated and tested

Involved:
Customer product software development team and base Software platform team

Basic Setup
- Adapt Software to platform
- Prepare interface to F&C and include annual release
- Prepare interface to OEM BB
- Establish platform features
- Adapt to latest HW revision
- Integration of CRs according feature based planning

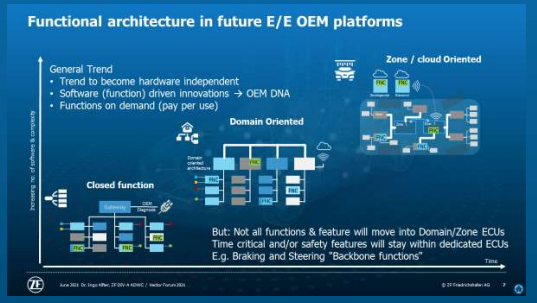


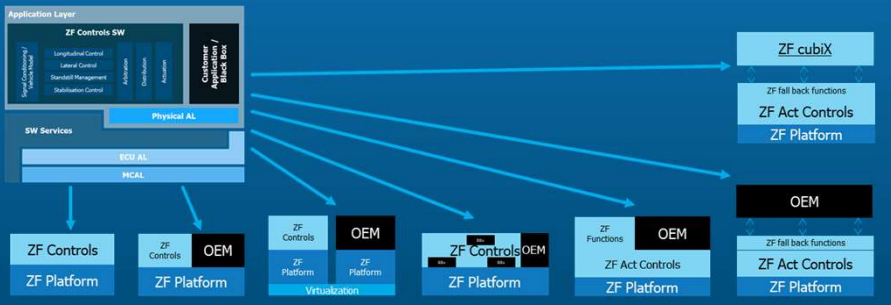


Summary

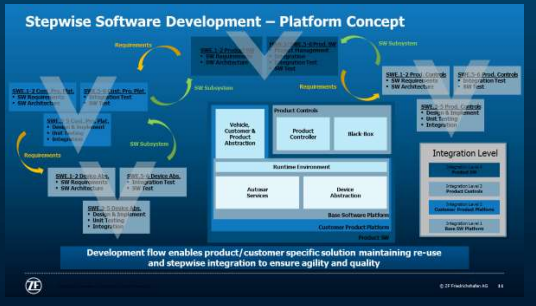
Market trends driven by Software Innovations …

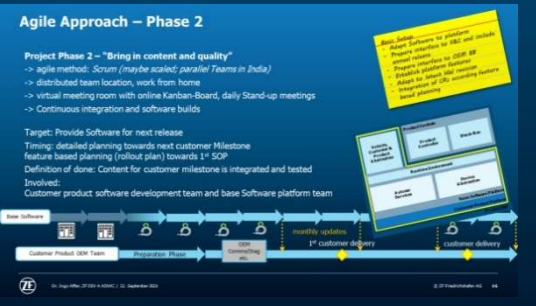
… requires Scalable E/E Architectures





ZF DIV-A Software Platform Concept with Stepwise development and Stepwise integration allowing testing in early phase and independent development





Thank you for listening

Q & A