Development effectiveness and efficiency: new priorities in software development

Potsdam, November 2, 2023

Roland Berger

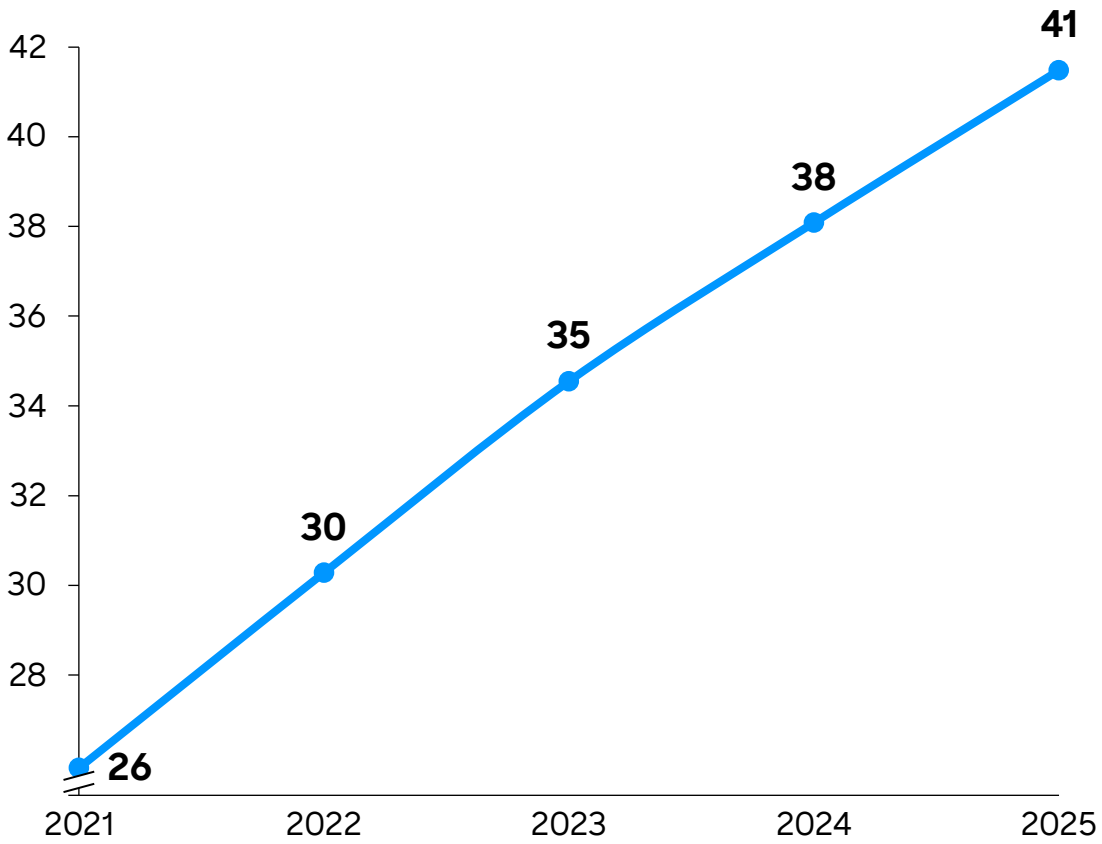
Contents

A. SW budget situation will become challenging

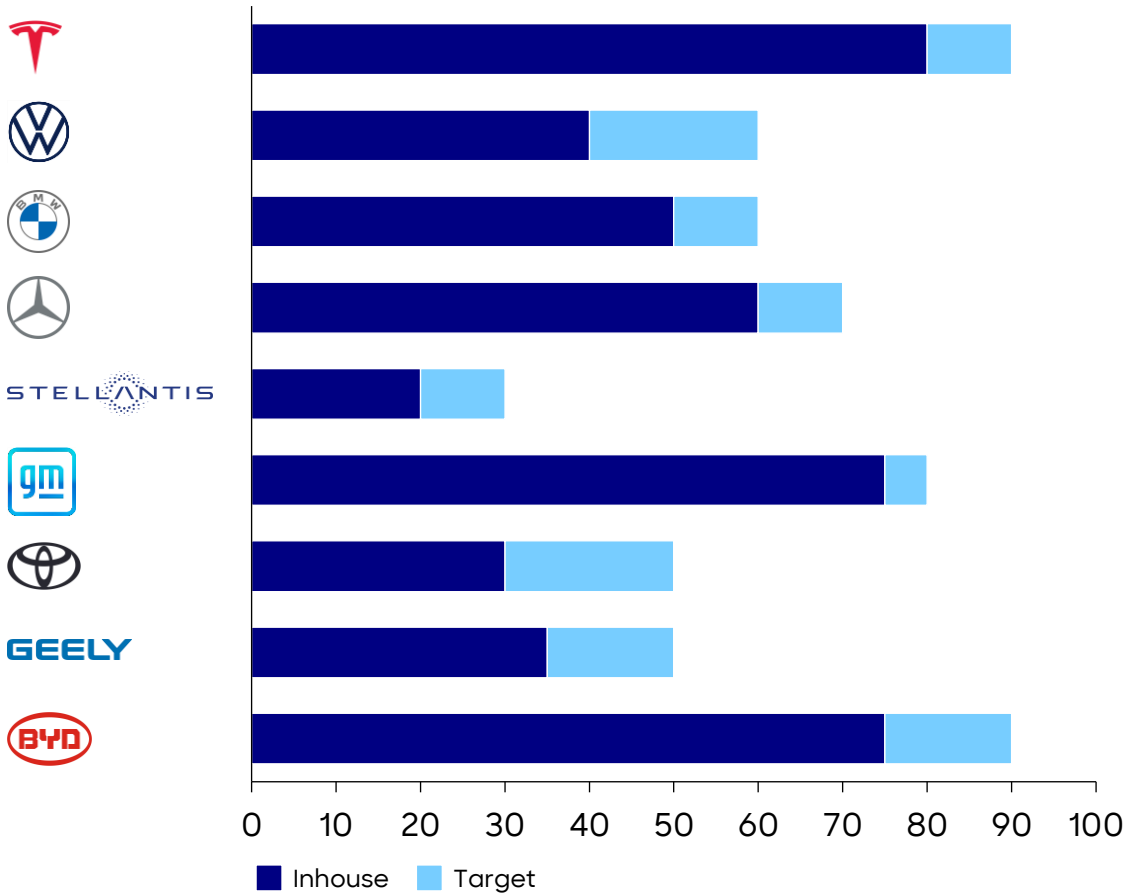
Software became a key differentiator for automotive OEMs and Tier-1

Software development budgets grew and ambitions for inhouse SW development rose

OEM's SW development budget[2] [USD bn]
(incl. external development and licenses)



OEM's inhouse SW development share[1] [%]

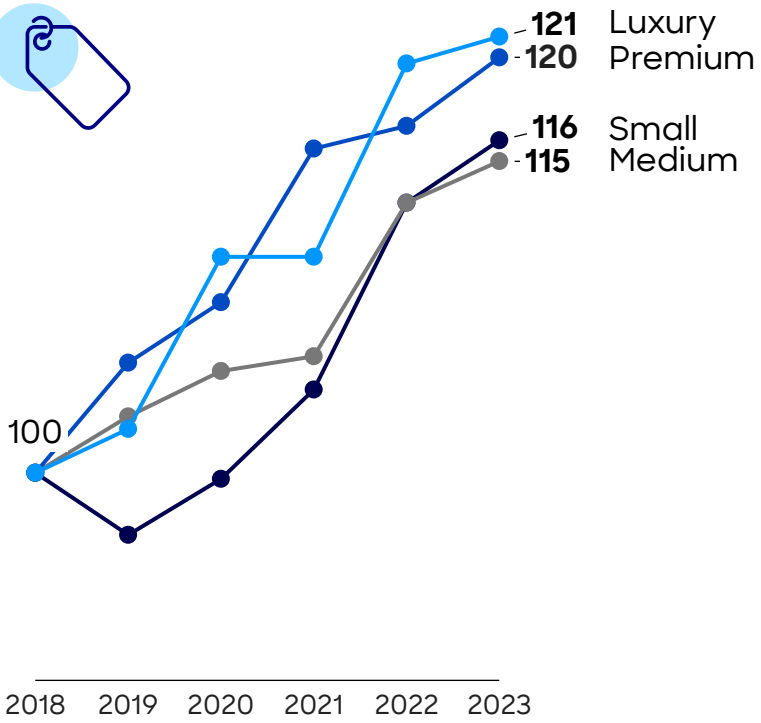


1) Budget plan as of Q3 2022

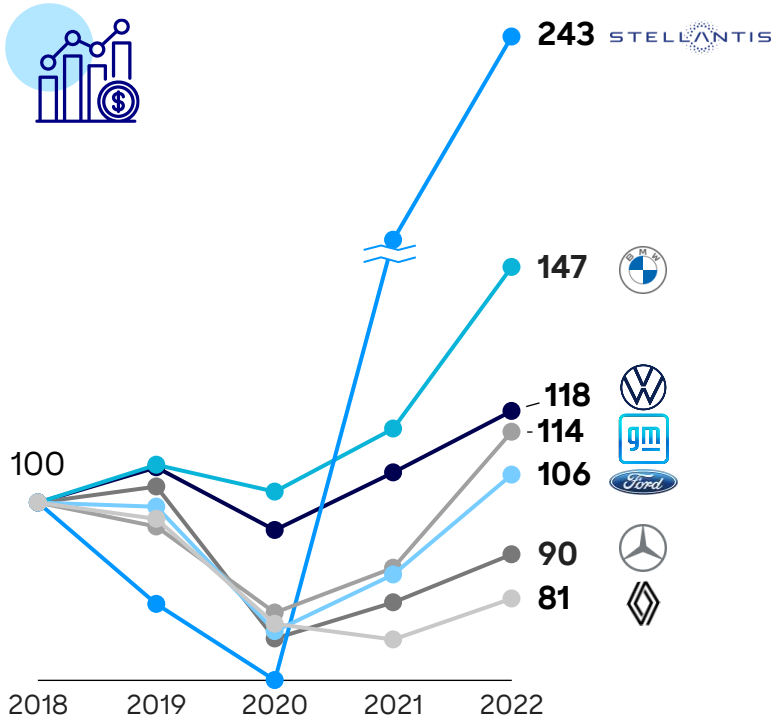
Source: Expert interviews; Roland Berger

… this had been possible as prices, revenues and profits grew

Price development [1)]

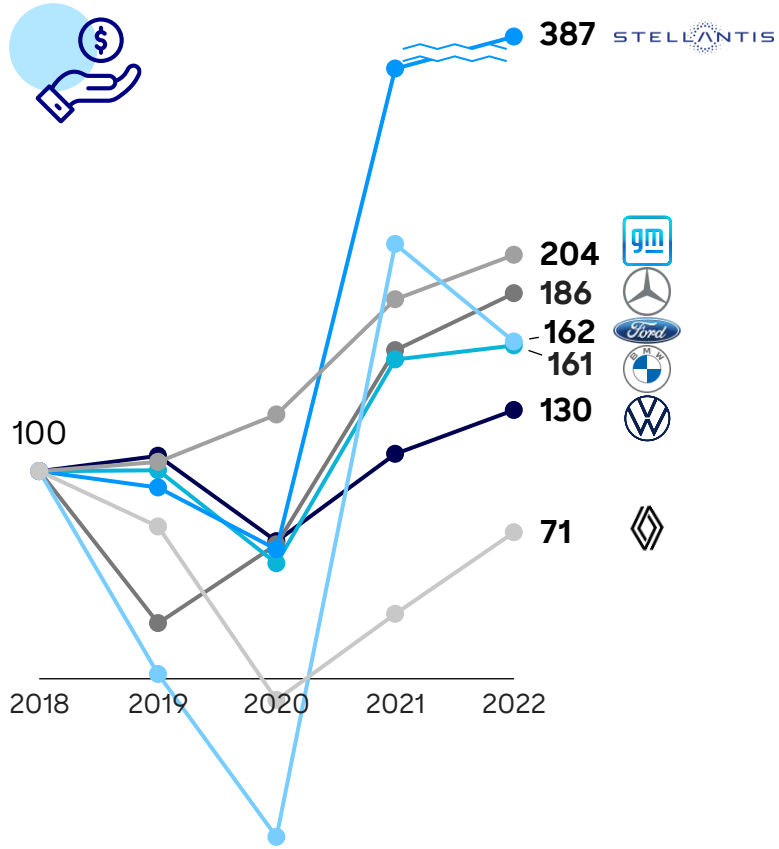


Revenue development



Stellantis: before 2021 PSA revenue only
Mercedes: before 2021 incl. Daimler Trucks

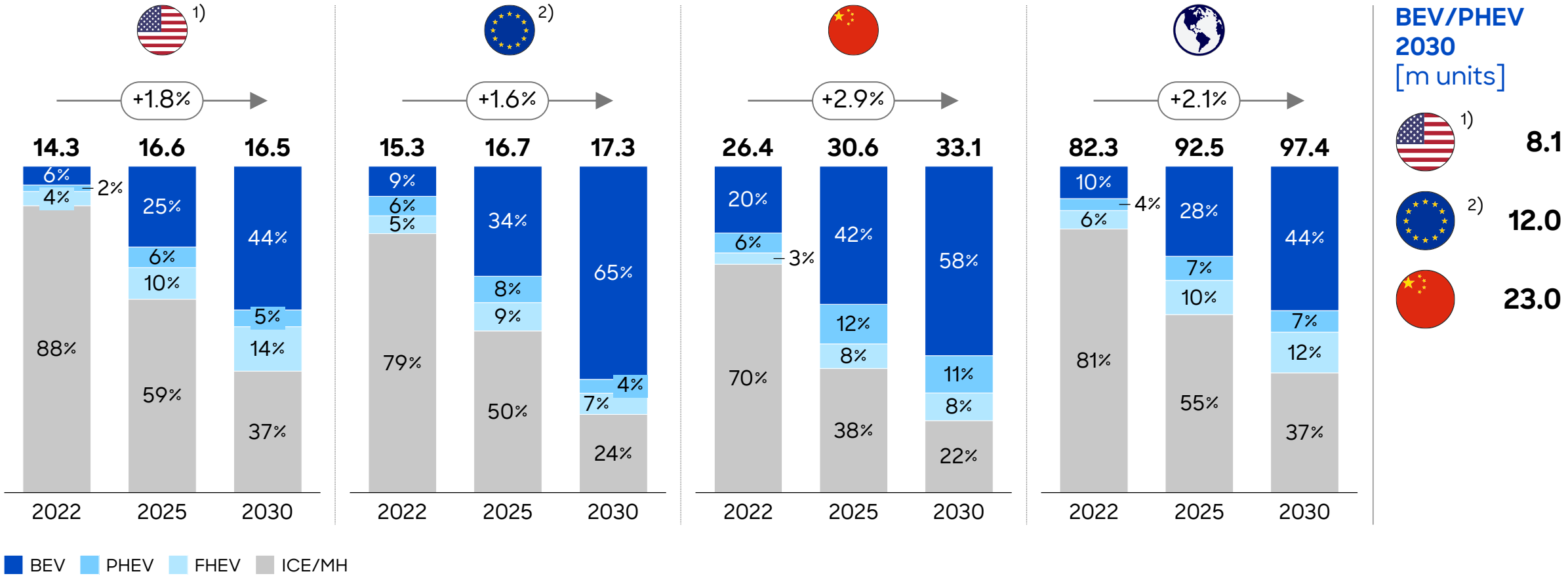
Profit development



1) Europe, North America; Based on selected vehicle sales prices of Audi, BMW, Mercedes-Benz, Renault, Skoda, VW and Volvo

BEV share will grow in all regions

PC production forecast by region & powertrain, 2022-2030, [m units, % of production]

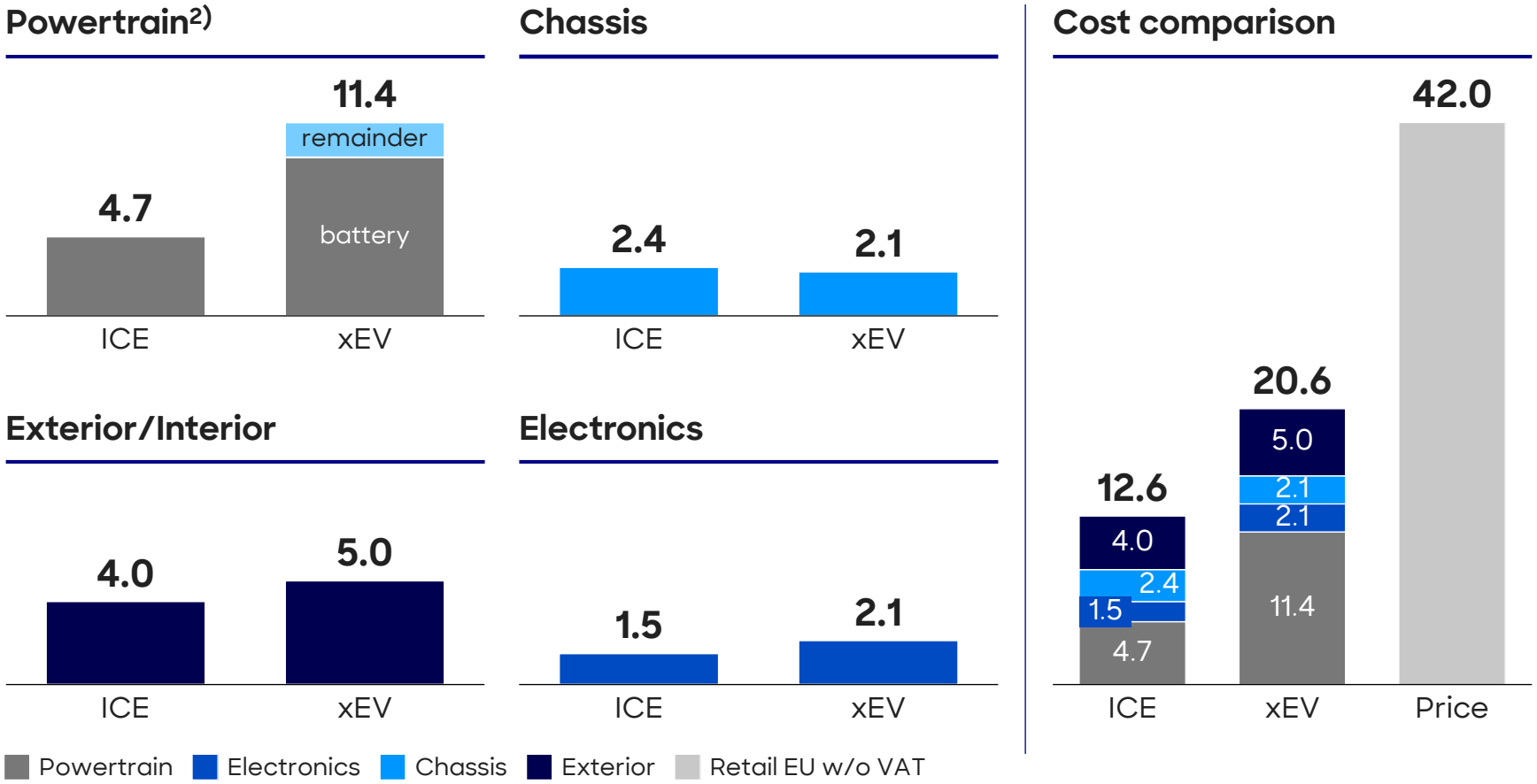


1) Incl. Mexico and Canada 2) EU27+UK, Norway, Switzerland FCEV are part of PHEV or FHEV

Source: IHS, Roland Berger

BEVs' higher material cost will become a major challenge – Little cost decline likely

Product cost ICE vs. xEV[1)] for C-segment SUV [k EUR]



Challenges

Limited additional willingness to pay for BEVs – **Follower and laggards expect price parity**

Minor cell cost reduction

Declining **ICE** volumes will cause a **long-term cost challenge**

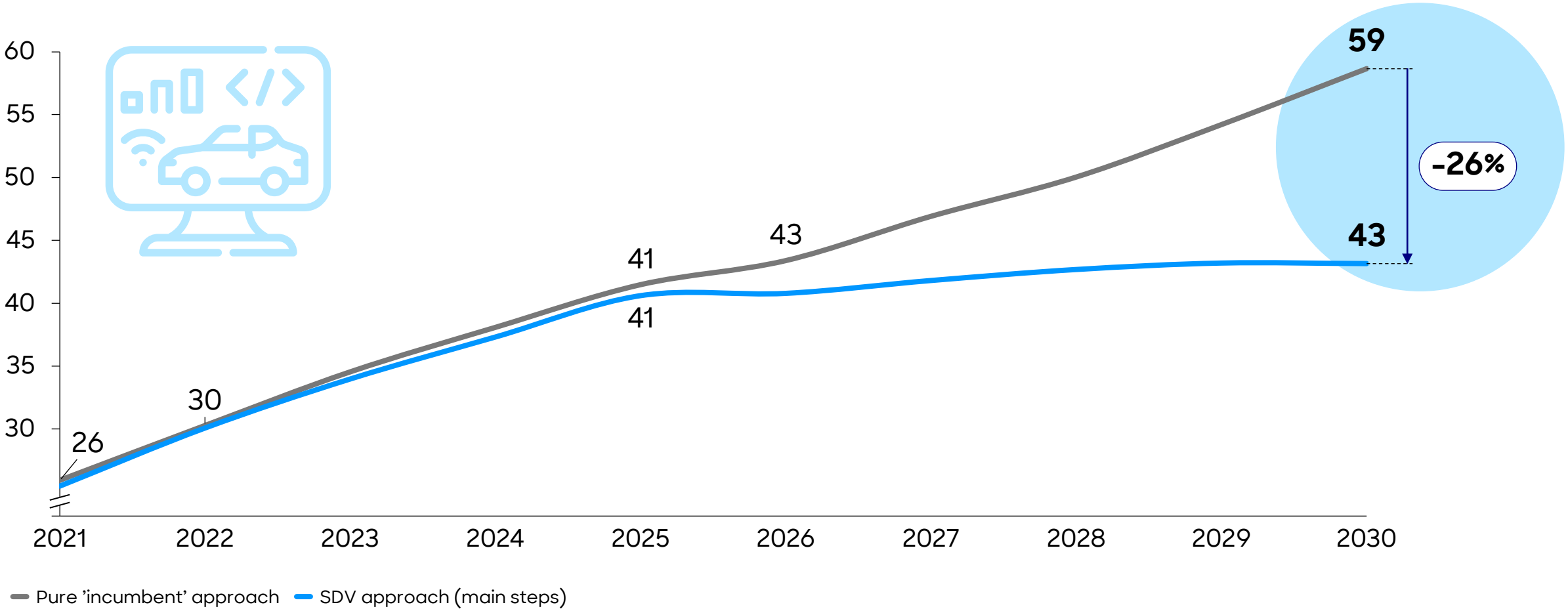
1) no development cost allocations and production costs; 2) BEV: 80 kWh battery, 150 kW peak; ICE: 170 kW, EURO 6d, AWD, DCT

Source: Roland Berger/Lazard

B. A more effective and efficient SW development approach

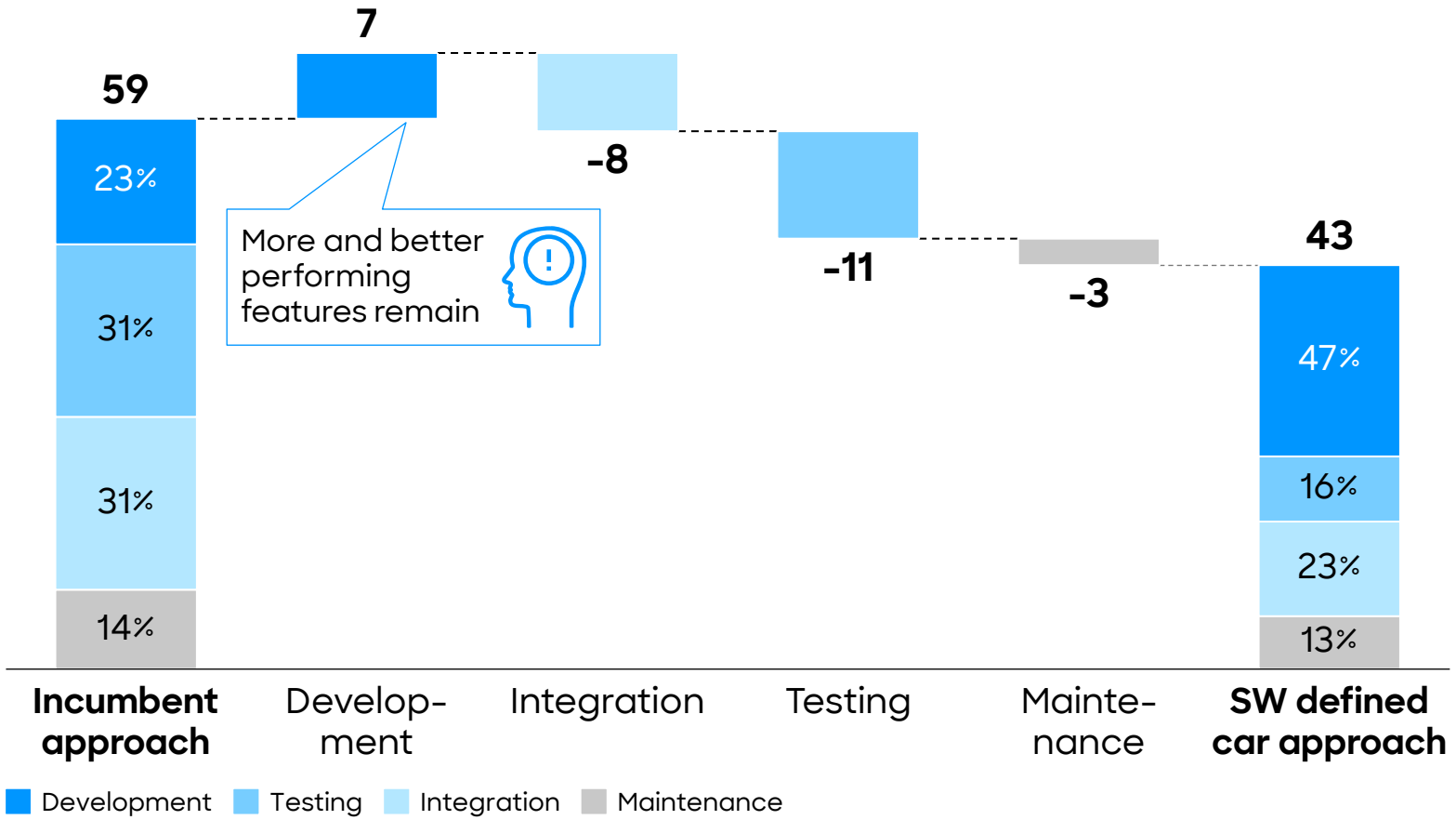
A new approach for software development is required, more effective and efficient approach for SW development is required

OEMs' in-vehicle SW budget 2021-2030 [USD bn]



The increased upfront effort to develop SW based on a 'SW defined car' approach is over-compens. by significant savings from testing, integration & maintenance

OEMs' in-vehicle SW spending 2030 – 'Incumbent' vs. SDV approach [USD bn]



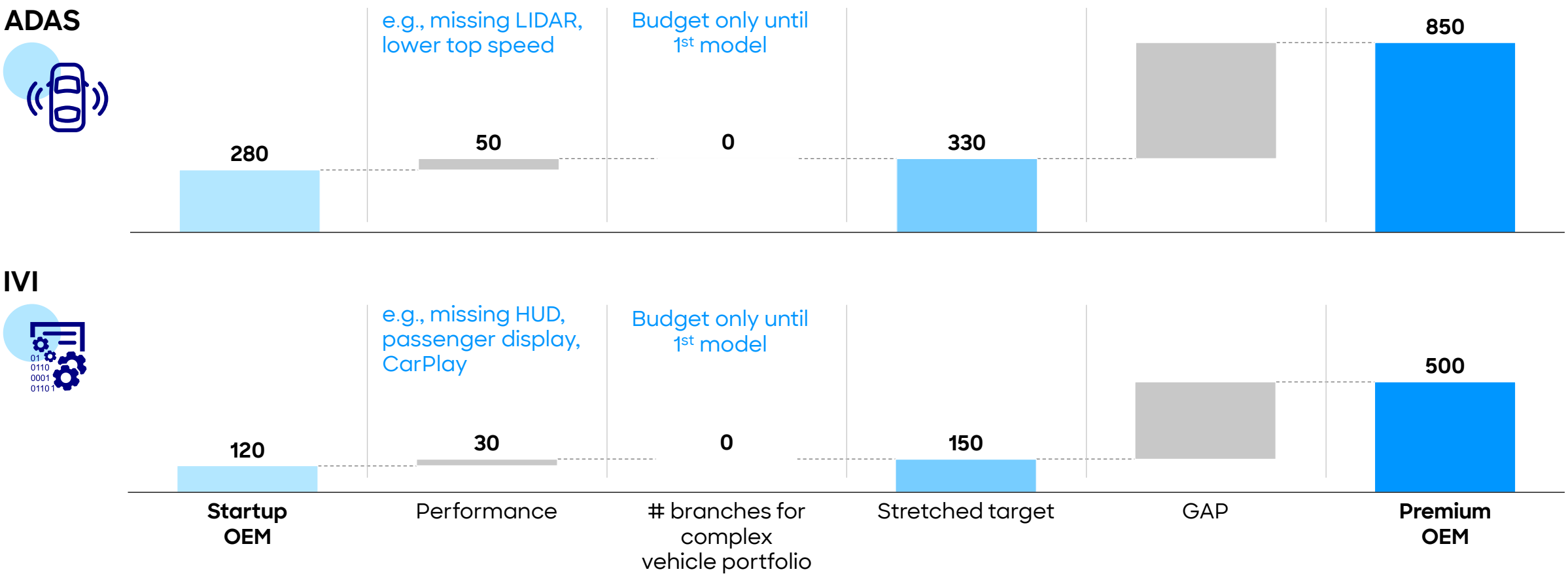
Enabler – OEM view

- A Reduce **variant complexity – Implement a SW first Governance**
- B Shift to new **SW architecture**
- C Rethink captive share and leverage **partnerships**
- D1 Dissolve **typical SW R&D inefficiencies**
- D2 Improve **DevOPS**

Source: Expert interviews

Major projects at SW pioneer OEMs are 2 to 3,5 times more effective and efficient compared to traditional OEMs

Development budget for SW stack [EUR m]



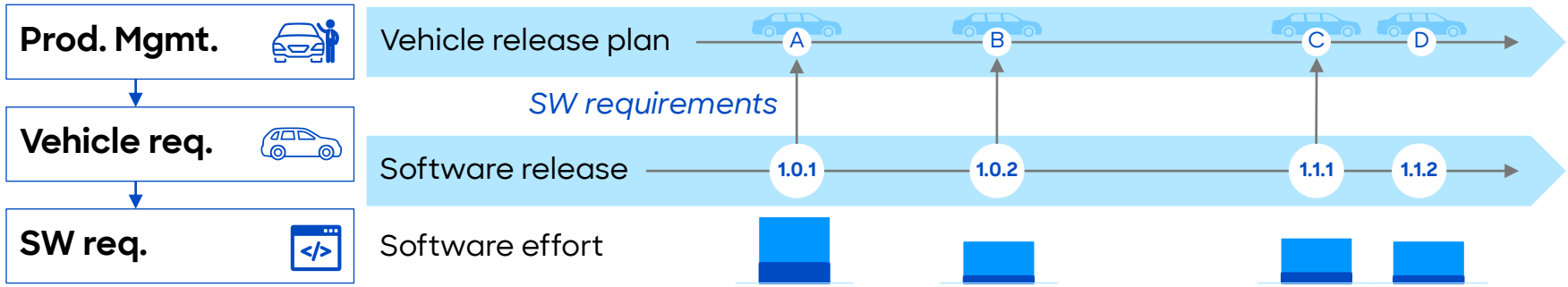
Source: Expert interviews

Illustrative

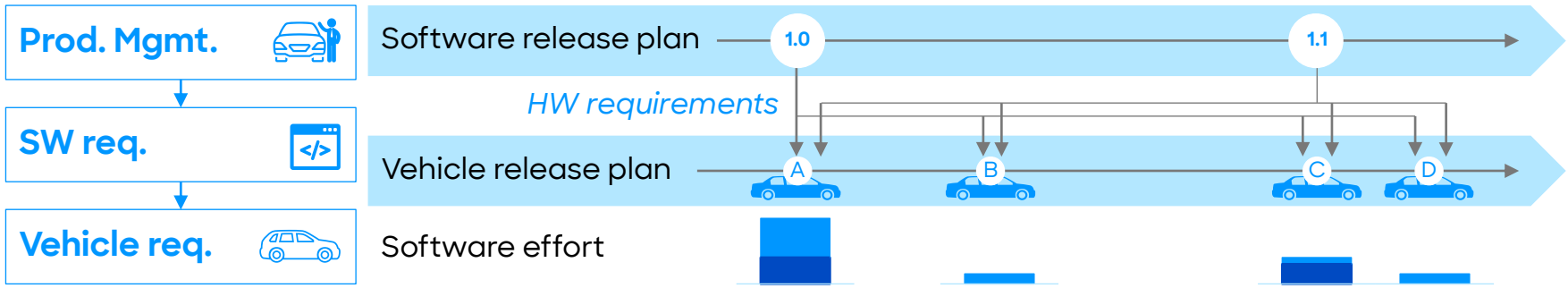
When SW defines the HW in a SW first approach, software adaptation effort per vehicle is drastically reduced

Concept illustration "Vehicle first" vs. "Software first" approach

Incumbent approach ("Vehicle first")



Software-defined approach ("SW first")



A: New vehicle model on new platform with new EE; B: New vehicle model on new platform with same EE
C: New vehicle model on new platform with new EE; D: New vehicle model on new platform with same EE

Left V Right V

Source: Roland Berger

Incumbent:

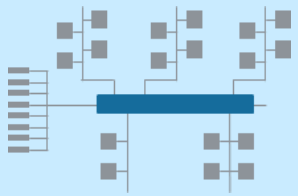
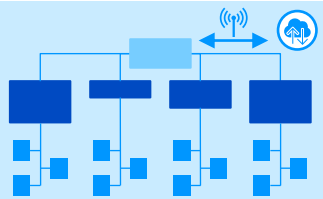
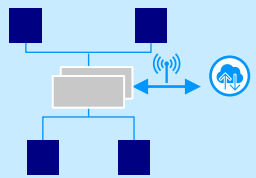
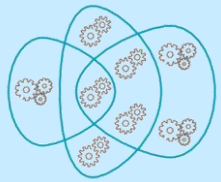
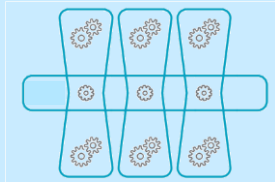
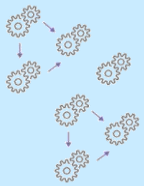
- HW defines the SW
- No/limited HW abstraction
- Each vehicle with individual SW variant
- High integration and testing costs of up to 70% even for same/similar functions

SW first:

- Innovation from SW releases
- "Identical" SW across models – largely reduced adaptation efforts
- HW with headroom

Advanced EE architectures go hand in hand with modern SW approaches and are therefore a prerequisite for efficiency gains

Definition of 'incumbent' vs. SDV approach

	'Incumbent' approach		SDV approach
Hardware (**HW** defines the SW)	**Distributed E/E** 	**Domain-centralized E/E** 	**Central+ zonal E/E** 
Software	**Monolithic architecture** 	**Service-orientated architecture** 	**Microservice architecture** 
Continuous deployment	Impossible	Costly + risky	Easy Low integration efforts
Functional orientation	Mostly in isolated ECUs	Mostly limited to each domain	Cross-domain
Maintenance	High	High/medium	Low

SW defines the HW

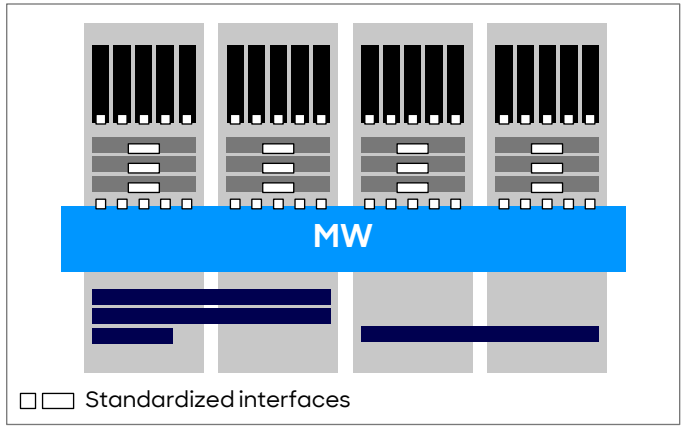
Gateway | DCU | ECU | Zonal ECUs (domain-independent) | Central/vehicle computer | Cloud/backend

Source: Roland Berger

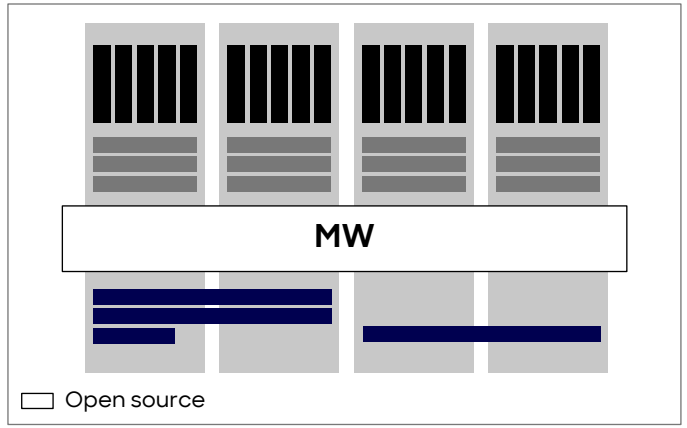
Open standards and open source are 2 intensively discussed options to realize a cross OEM middleware

Realization options for Cross OEM MW

Open standards



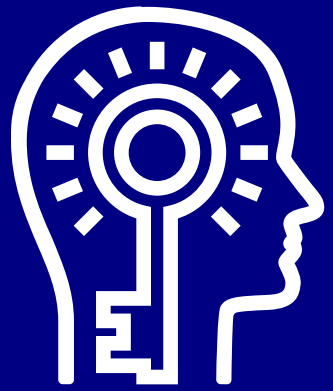
Open source



	Open standards	Open source
Description	• OEMs agree on **standards for all key interfaces**	• **Solutions / technology is open source** while a partner is maintaining and improving MW and toolchain based on contract • Red Hat's business model is the blueprint
Benefits	• Middleware provides **can easily be exchanged**	• **Low exchange cost** for Middleware provider
Status	• **AUTOSAR and ECLIPSE** are key initiative	• **First suppliers** are in discussions with OEMS

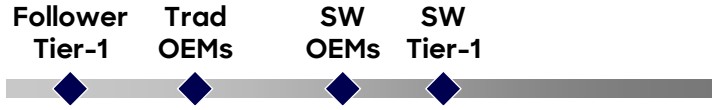
■ Apps (customer feature) ■ Service ■ Cross OEM middleware (OS, HW abstraction) ■ HW (ECU) ■ Domain

Open standards can be agreed on domain levels, however **open-sourced MW with stronger impact** if fully implemented and accepted by OEMs.



SW R&D inefficiencies to be addressed, dedicated SW R&D steering model & KPIs required, as well as state-of-the-art dev & V&V methods & cost transparency

Key success factors SW R&D inefficiencies and player assessment

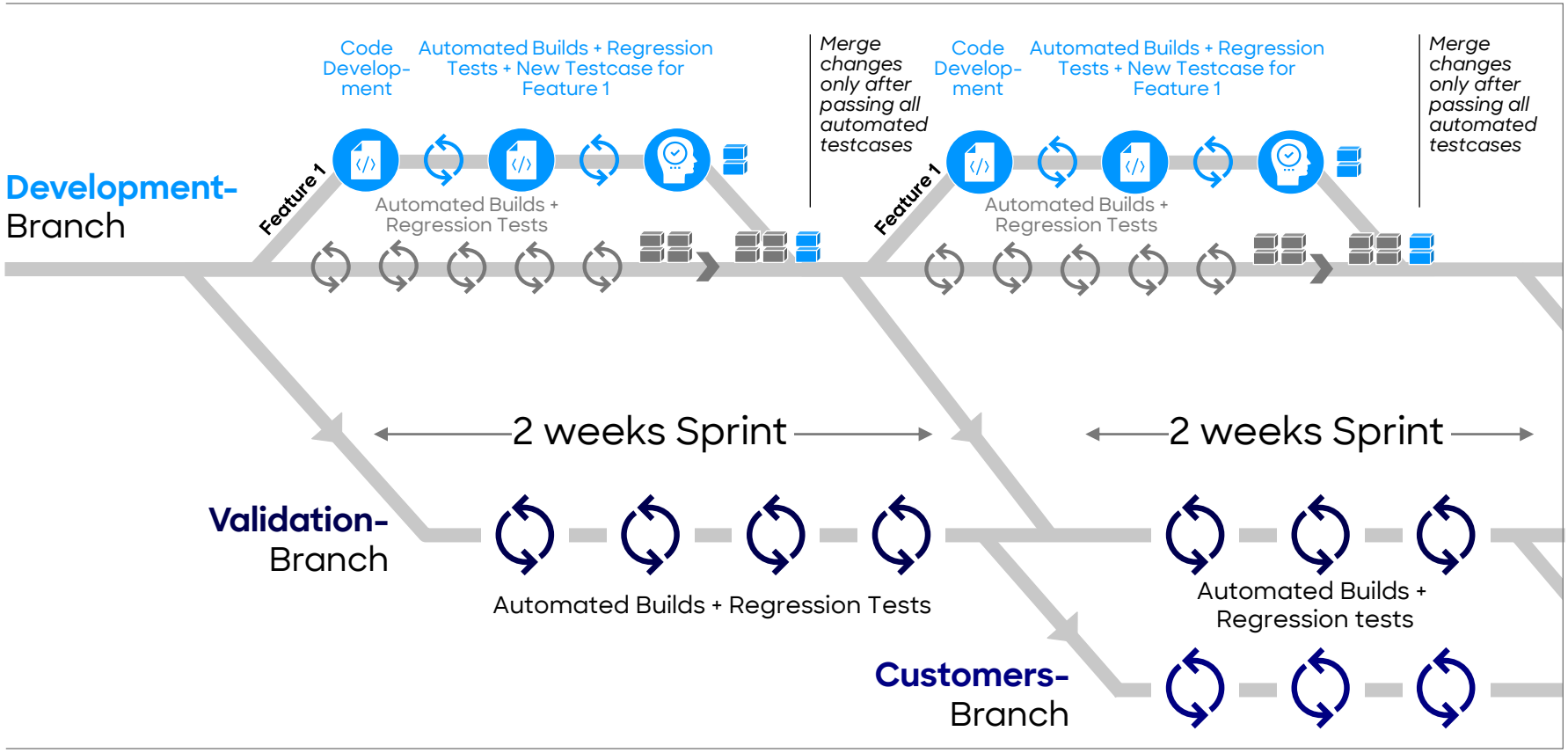
Success factor	Description	Assessment
Dedicated steering model and KPIs within R&D 	• Shift of **financial steering KPIs** towards "recuring revenues" & SW business models • **Technical early-warning KPIs** required to drive efficient steering models • Balancing **SW R&D development steering in a still HW-mindset** industry	
Transparency on SW development cost drivers 	• **Lack of understanding** of how tailoring and architecture inefficiencies drive SW development costs • Often **inefficient SW procurement**, due to unfamiliarity with SW costing • Partly lack of ability to deal with **SW specifics (licensing costs, OtA updates etc.)** in HW-heavy industry	
State-of-the-art development & testing methods (e.g., reuse, virtual testing) 	• More dedicated SW validation & verification activities, such as **virtual testing** and increase use of **SiL (& HiL) testing** • Leverage of **system engineering development** approach and **quality gate checks** • **Drive re-use and standardization** of development (higher targets than in HW R&D)	

SW OEMs = OEMs build on SW competencies
Trad OEMs = Established OEMs

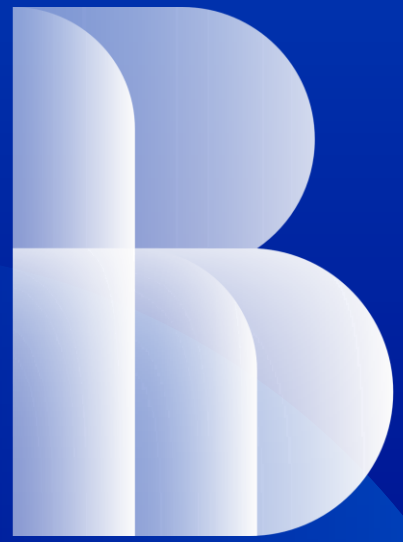
SW Tier-1 = Tier-1 with strong heritage in large SW projects
Follower Tier-1 = Supplier with limited SW experience

Separate development, validation, and customer Branch provide a stable basis for releasing and integrating features, fixing bugs and stabilizing SW functions

CI/CD (Continuous Integration/ Continuous Delivery)



- **Development-Branch:** All new functions are developed and defined here
- **Validation-Branch:** No new functions are merged here, but only used for validation-/ error correction
- **Customer-Branch:** Stable code base for customers and only open for critical bugs (hotfixes)
- After each sprint, the validation-Branch becomes a new customer Branch, and the development-Branch becomes a new validation-Branch
- Regression tests are regularly performed in all code Branches



Roland
Berger