When things get complex

Complex Systems, challenges and where to focus

Georg Doll, Senior Expert, McKinsey Digital Munich

Georg Doll

Senior Expert at McKinsey Digital Munich

Georg Doll, Senior Expert at McKinsey Digital in Munich

As Member and Co-Lead of the EMEA Software Service line at McKinsey Digital, he supports clients along the product development LiveCycle.

With his background of over 20 years' experience in Automotive and embedded Software delivery and management of international teams. He supports clients in Market introduction planning, introduction of systems engineering, improving in project execution excellence and agile software development and talent management.

He has served Tier1s, semiconductor vendors and vehicle manufacturers around the world in Japan, Asia Pacific, EMEA and US.

In 2009 he was instrumental in setting up the GENIVI Alliance, and served as member of the Strategy Council and the Board for several years.

Abstract

Software is on the rise. Software is the no. 1 topic in the development of new functions. The software market is expected to grow from today to 2030 with an average CAGR of ~10%. So what could go wrong?

A closer look at the four major automotive trends shows that they depend on the success of software. OEMs and Tier1s have recognized the situation and are investing heavily in software.

To the extent that some speak of a "software defined car" or a "computer on wheels".

But where there is light, there is also shadow. Highly automated driving, connectivity, powertrain electrification and new mobility services introduce additional dependencies between functions.

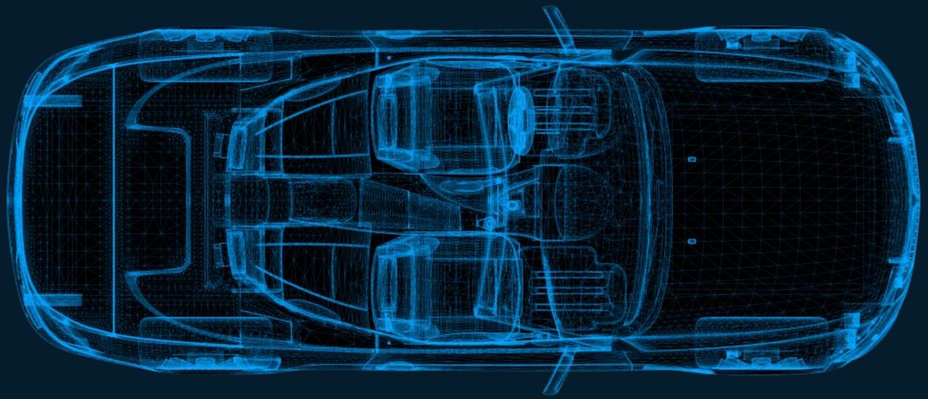
Dependencies that – as we know continue to increase the system complexity.

The development of software is a constant challenge for projects. What seems trivial at first glance turns out to be a much greater challenge than many people expect. Falling productivity, increasing communication, declining quality, constantly rising development costs and project delays are omnipresent.

These challenges can only be tackled with a holistic approach.

Successful organization have mastered the most important dimensions. Development tools, program management and talent management are just some of this dimensions.

Software is on the rise..



and it is complex.

Christoph Grote

SVP Electronics at BMW

“”

Today **95% of innovation** in automotive **is software** based

Olla Källenius

Chairman of the Board
Daimler AG and Mercedes-Benz AG

“”

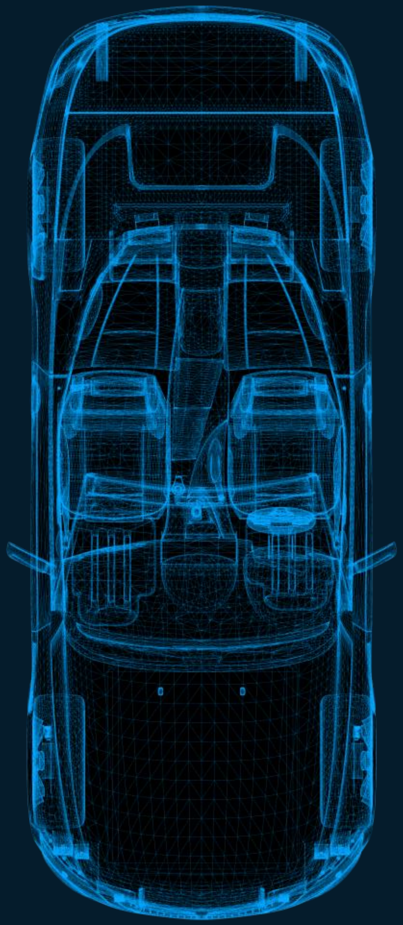
To stay relevant we have to control the Software in our vehicles

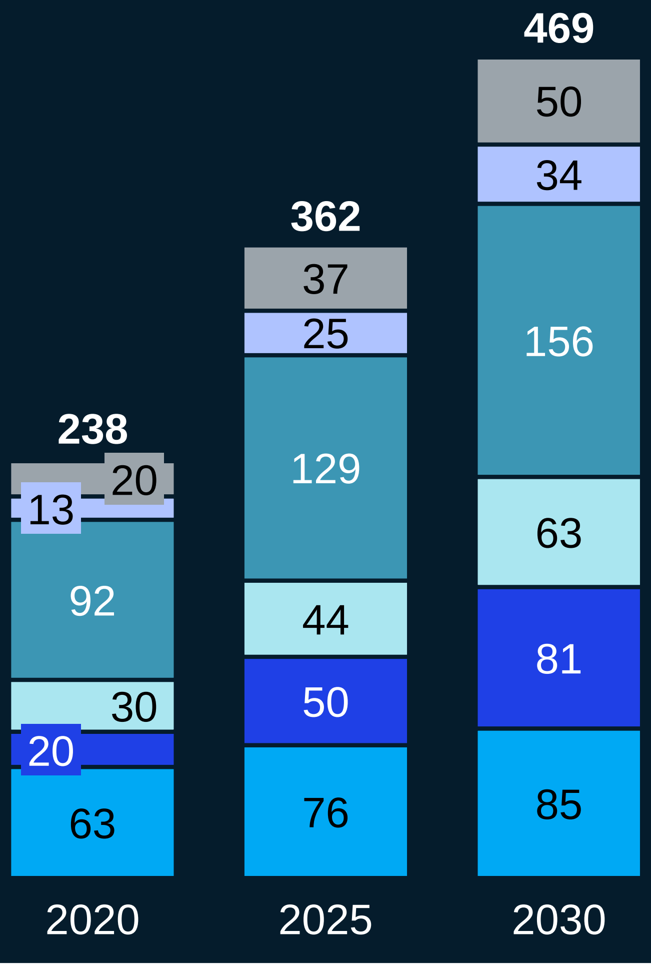
Linus Torwald

Principal developer Linux Kernel

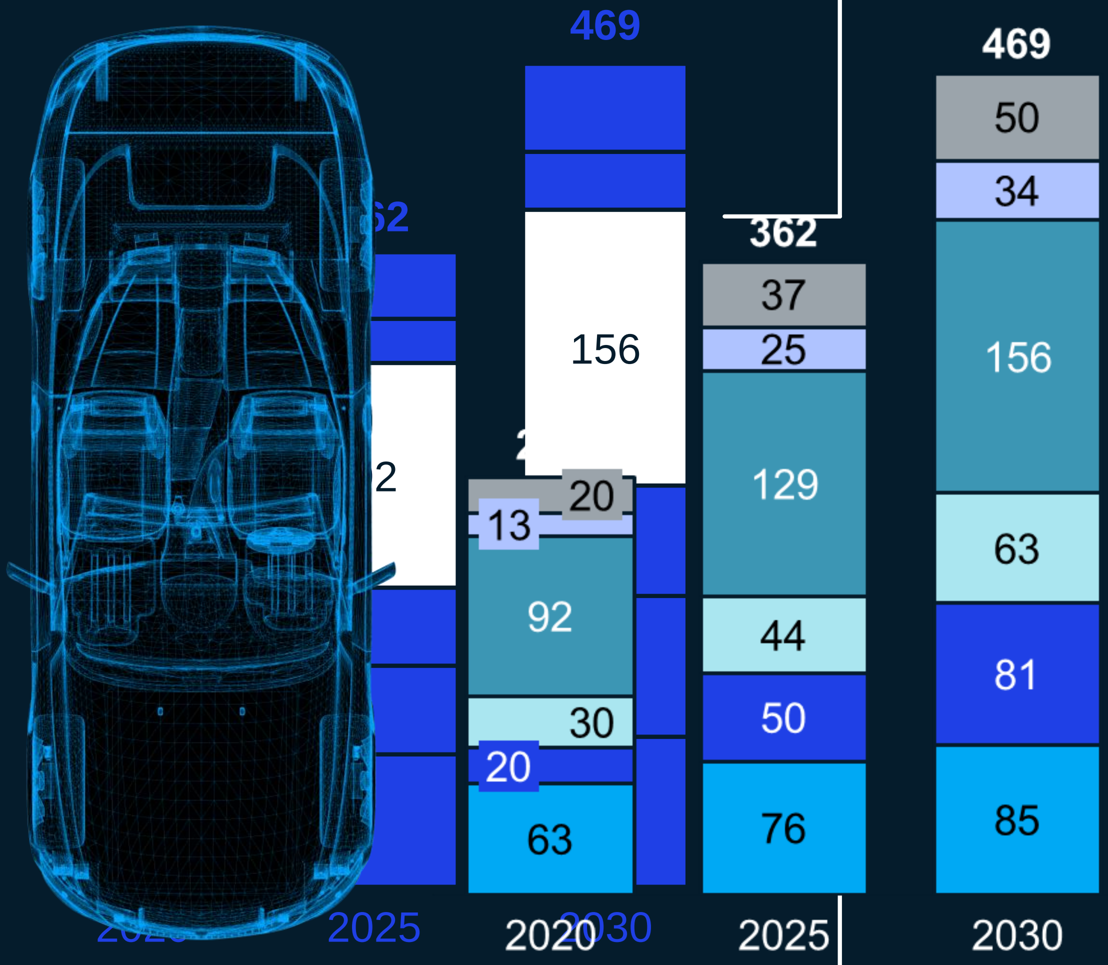
“”

The value of Software is not in the code, **its in the head of** the **people** who developed the code



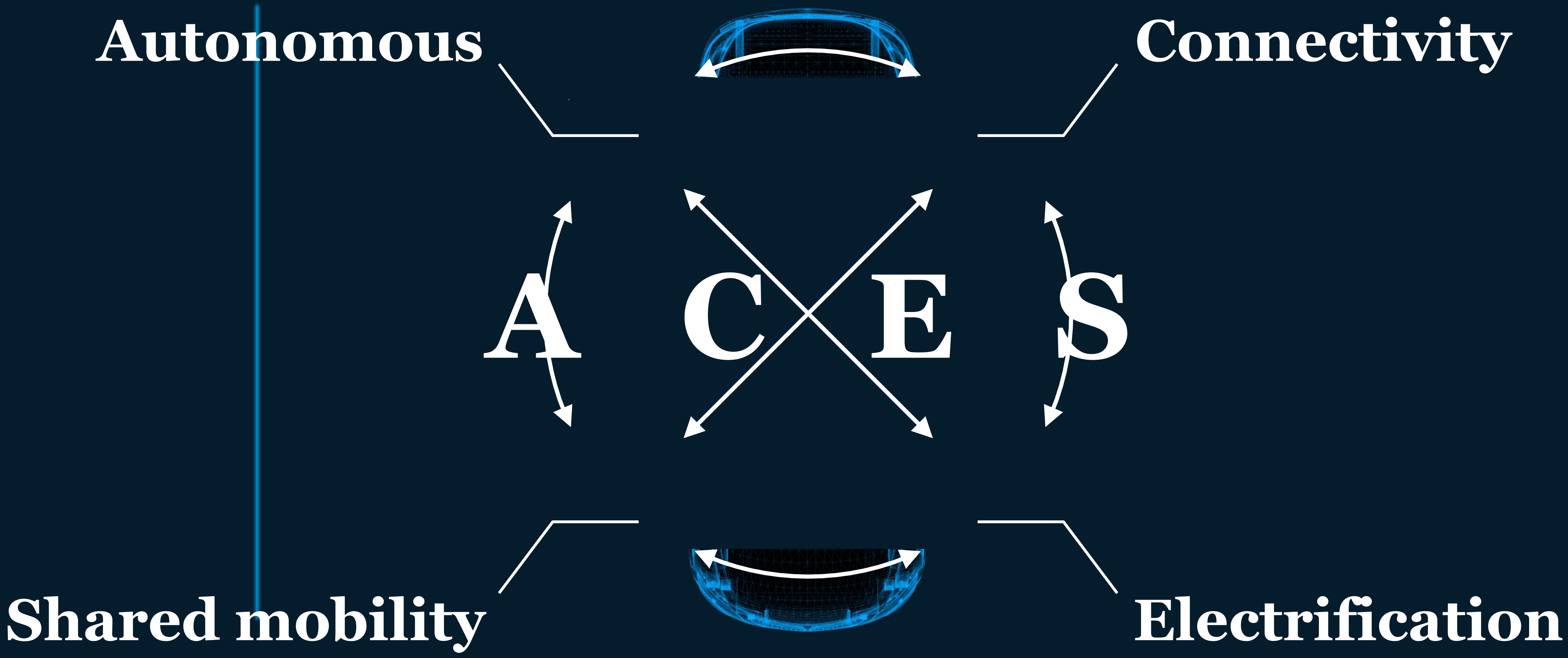


	2020	2025	2030	Estimated CAGR 2020 - 30	Components
	238	**362**	**469**	**7.0%**	
	20	37	50	9%	Software (functions, OS, middleware)
	13	25	34	10%	Integration, verification and validation
	92	129	156	6%	Electronics ECUs/DCUs
	30	44	63	7%	Sensors
	20	50	81	18%	Power electronics (excl. battery cells)
	63	76	85	3%	Other electronic components (harnesses, controls, switches, displays)
Automotive sales USD billions	2,755	3,027	3,800	**3%**	



Estimated CAGR 2020 - 30	Components
7.0%	
9%	Software (functions, OS, middleware)
10%	Integration, verification and validation
6%	Electroncis ECUs/DCUs
7%	Sensors
18%	Power electronics (excl. battery cells)
3%	Other electronic components (harnesses, controls, switches, displays)

	2020	2025	2030	
Automotive sales USD billions	2,755	3,027	3,800	**3%**



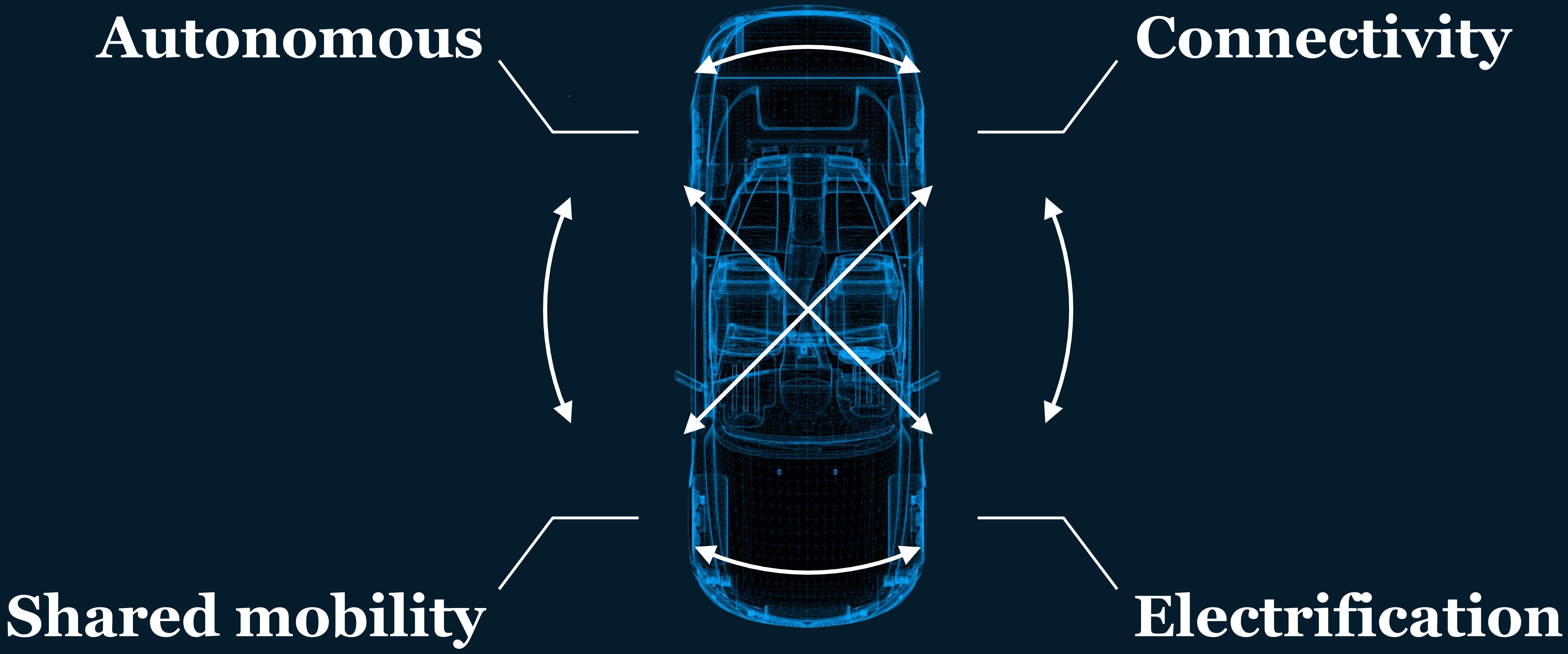
Autonomous

Connectivity

A C E S

Shared mobility

Electrification

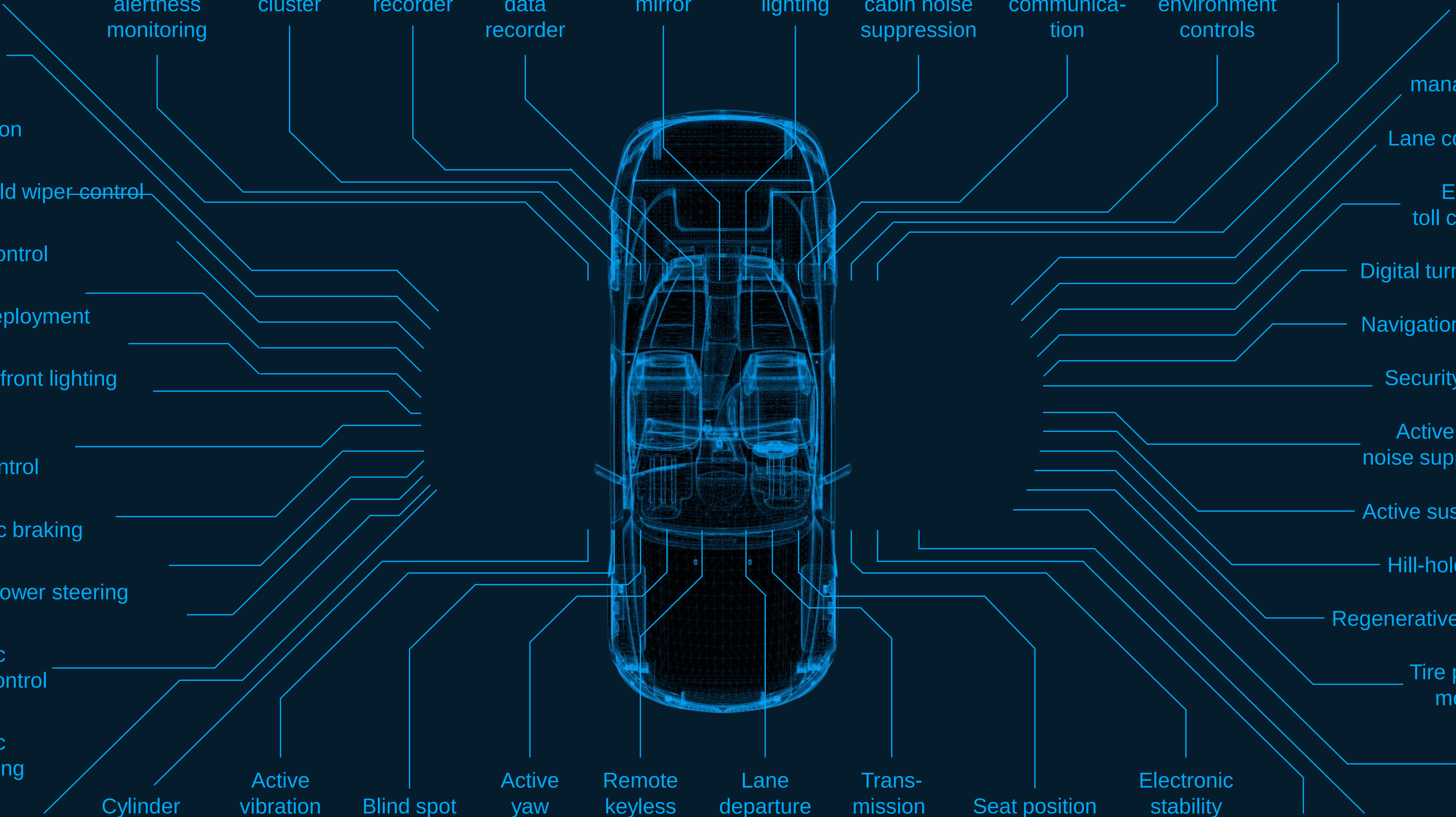


Autonomous

Connectivity

Shared mobility

Electrification



Parental controls

Driver alertness monitoring

Instrument cluster

Accident recorder

Event data recorder

Auto-dimming mirror

Interior lighting

Active cabin noise suppression

Voice/data communication

Cabin environment controls

DSRC

Entertainment system

Head-up display

Night vision

Windshield wiper control

Engine control

Airbag deployment

Adaptive front lighting

Adaptive cruise control

Automatic braking

Electric power steering

Electronic throttle control

Electronic valve timing

Idle stop/start

Cylinder deactivation

Active vibration control

Blind spot detection

Active yaw control

Remote keyless entry

Lane departure warning

Transmission control

Seat position control

Electronic stability control

OBDII

Anti-lock braking

Battery management

Lane correction

Electronic toll collection

Digital turn signals

Navigation system

Security system

Active exhaust noise suppression

Active suspension

Hill-hold control

Regenerative braking

Tire pressure monitoring

Parking system

Emission regulations drive need for new power train solutions

Eco functions require tighter function integration

Eco functions bring the need for new sensors

Electrification brings new technologies into the car

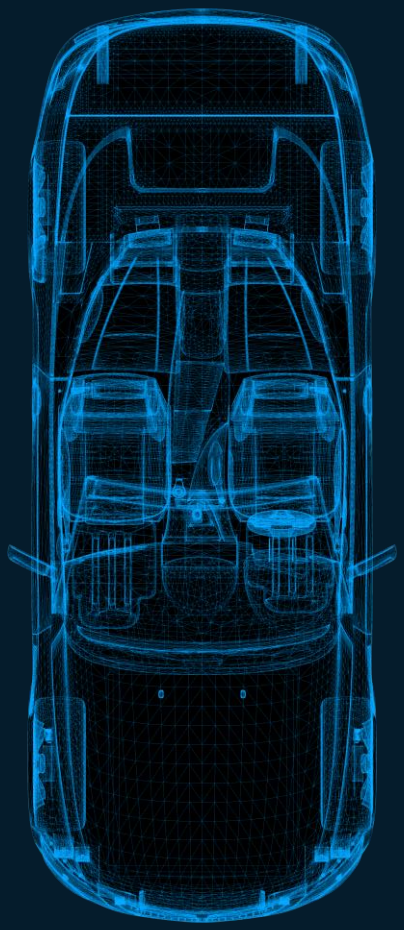
Road safety requirements drive the need for new sensors

Connectivity introduces security threads

Connectivity brings new features to the fleet

ADAS functions increase safety levels of vehicle functions

Highly automated driving increases functional dependencies



Emission regulations drive need for new power train solutions

Eco functions require tighter function integration

Eco functions bring the need for new sensors

Electrification brings new technologies into the car

Road safety requirements drive the need for new sensors

Connectivity introduces security threads

Connectivity brings new features to the fleet

ADAS functions increase safety levels of vehicle functions

Highly automated driving increases functional dependencies

widening gap between mplexity and productivity vels is driving to **performance d schedule challenges for tomotive players**

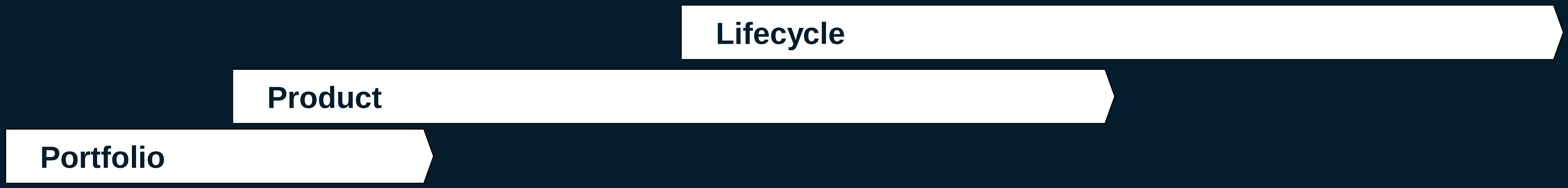
order to overcome those allenges, SW development ganizations need to both:

Manage the increase in **complexity**

ncrease efficiency to enable management of complexity

SOURCE: McKinsey Numetrics analytics on SW complexity and productivity in automotive

	1	2	3	4	5	6	7
Complexity dimension	**Functional redundancy**	**Versions variety**	**Interfering sub-systems**	**Closed systems**	**Point to point Interfaces**	**Multiple HW platforms**	**Code and documentation quality**
Description	Multiple applications covering same functionality redundantly in the portfolio Overlapping functionality between components in the same system	Multiple versions of the same application/ system are “live” at the same time	Multiple applications/ sub-systems within a SW platform are competing for similar resources (compute, storage, power)	Components within a system are developed as monoliths impeding accessibility of single elements for updates/maintenance and integration within new development	Unmanaged interdependencies between systems are developed as point-to-point interfaces leading to a high complexity and variety of interfaces within and beyond the system	Operating system, HW complexity, and testing environment with strong influence on system complexity	Code size, quality, and documentation as further sources of complexity throughout the lifecycle



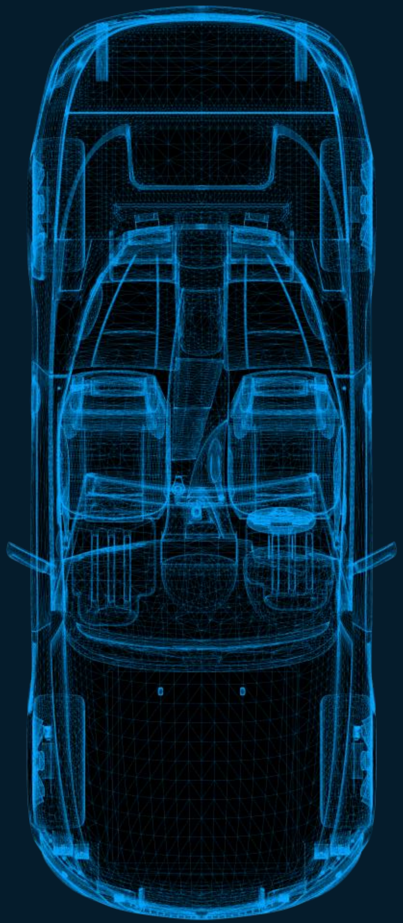
Lifecycle

Product

Portfolio

3 Interfering sub-systems

5 Point to point Interfaces



Cloud platform

—

Connectivity (back-haul)

—

UI/UX/HMI

—

Applications

—

Artificial intelligence/Advanced analytics

—

Middleware layer/OS

—

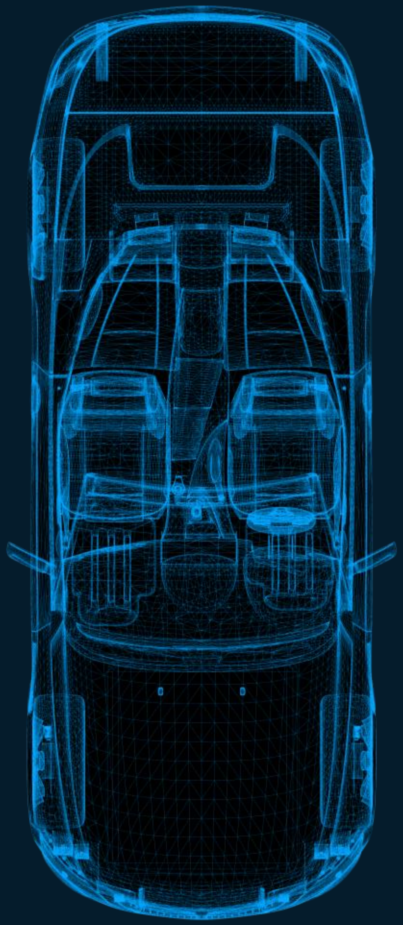
E/E hardware

—

Sensors | Actuators | Power components

—

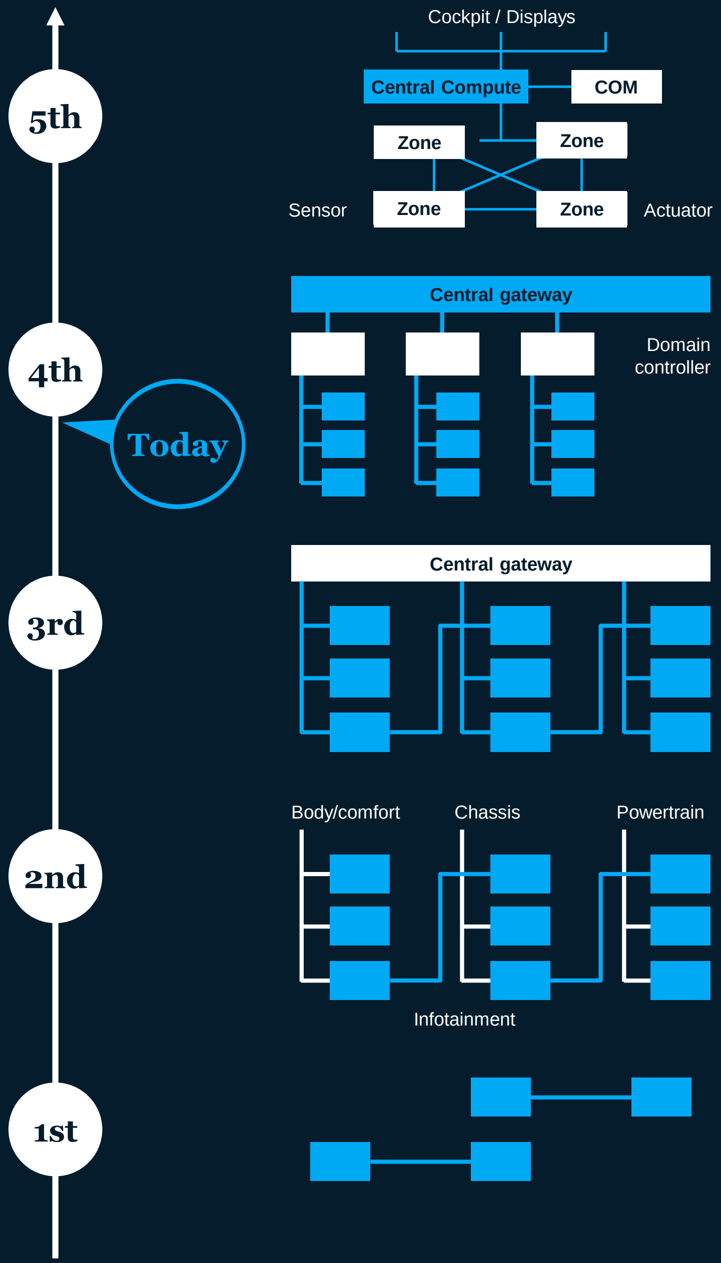
Vehicle



Vehicle centralized

Domain-centralized

Distributed



- Computer on wheels
- Centralized Computing
- Cross-functional connection
- Collaboration of ECUs
- Isolated Functions

“How to **standardize Software** across my different product lines and product generations?”

“How can I ensure that my **1 billion USD** software investment is delivered **on time and on budget**?”

“How to **transform 10.000 hardware-oriented developers** to an agile-minded, software-driven organization?”

“How to **organize software developers** across my divisions”

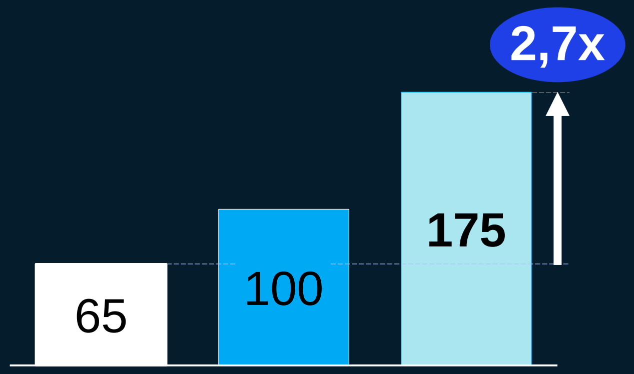
“How do I get **access to the best software talent**?”

“How to transform our management systems to drive **world class embedded software performance?**”

“My software org is a black box to me. How do I assess and **boost the embedded software dev. productivity** of my 5.000 distributed developers and my suppliers?”

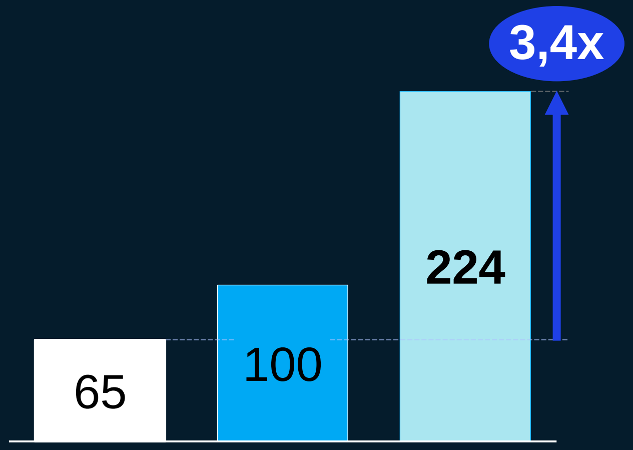
SOURCE: McKinsey

Productivity



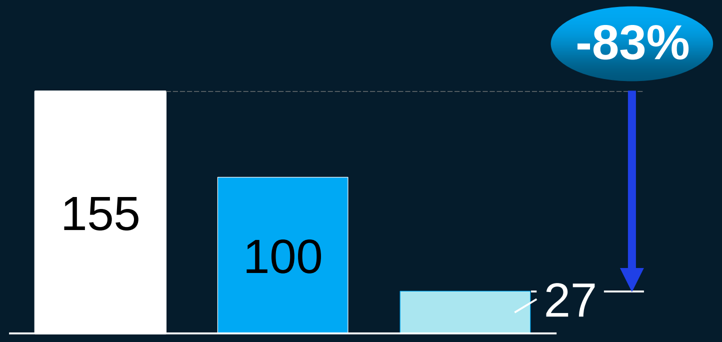
Complexity units per man week[1]

Development throughput



Complexity units per week[1]

Quality



Residual design defects[1]

Top organizations show the potential increase in Software development performance for average and bottom quartile organizations

□ Bottom quartile ■ Average[1] ■ Top quartile

[1] Average indexed to 100

SOURCE: Numetrics embedded SW project database

Developer Velocity Index (DVI)

Deep structured interviews

100+ industry experts

Comprehensive survey

440 large organizations across 12 industries and 9 countries

Statistical correlation analysis

Business performance (financial performance, innovation, customer experience, brand, talent) against the various dimensions of DVI

Technology

Working practices

Organizational enablement

Technology

Architecture

- Software architecture
- Data architecture

Infrastructure and platform

- Public cloud adoption (IaaS, PaaS)
- Infrastructure as code

Testing

- Test automation
- Test driven development

Security and compliance

- Security practices
- Compliance practices

Tooling

- Tools (planning, development, DevOps, collaboration)
- AI assistance
- Low code/no code

Working practices

CI/CD practices

- Repeatable builds, continuous integration, delivery, and deployment

Engineering practices

- Code reviews
- Coding guidelines
- Technical debt management

Agile team practices

- Ceremonies
- Definition of done
- WIP management

Open source, inner source

- Open source usage and contribution
- Inner source adoption

Organizational enablement

Team characteristics

- Cross-functional teams
- Autonomous scope
- Co-location
- Limited context switching
- **Product management**
- Product manager/owner capabilities
- Product telemetry
- Rapid prototyping
- Clear product vision and requirement
- Linkage between strategy and team metrics

Organizational agility

- Agile funding mechanism
- Portfolio management
- Dependency management

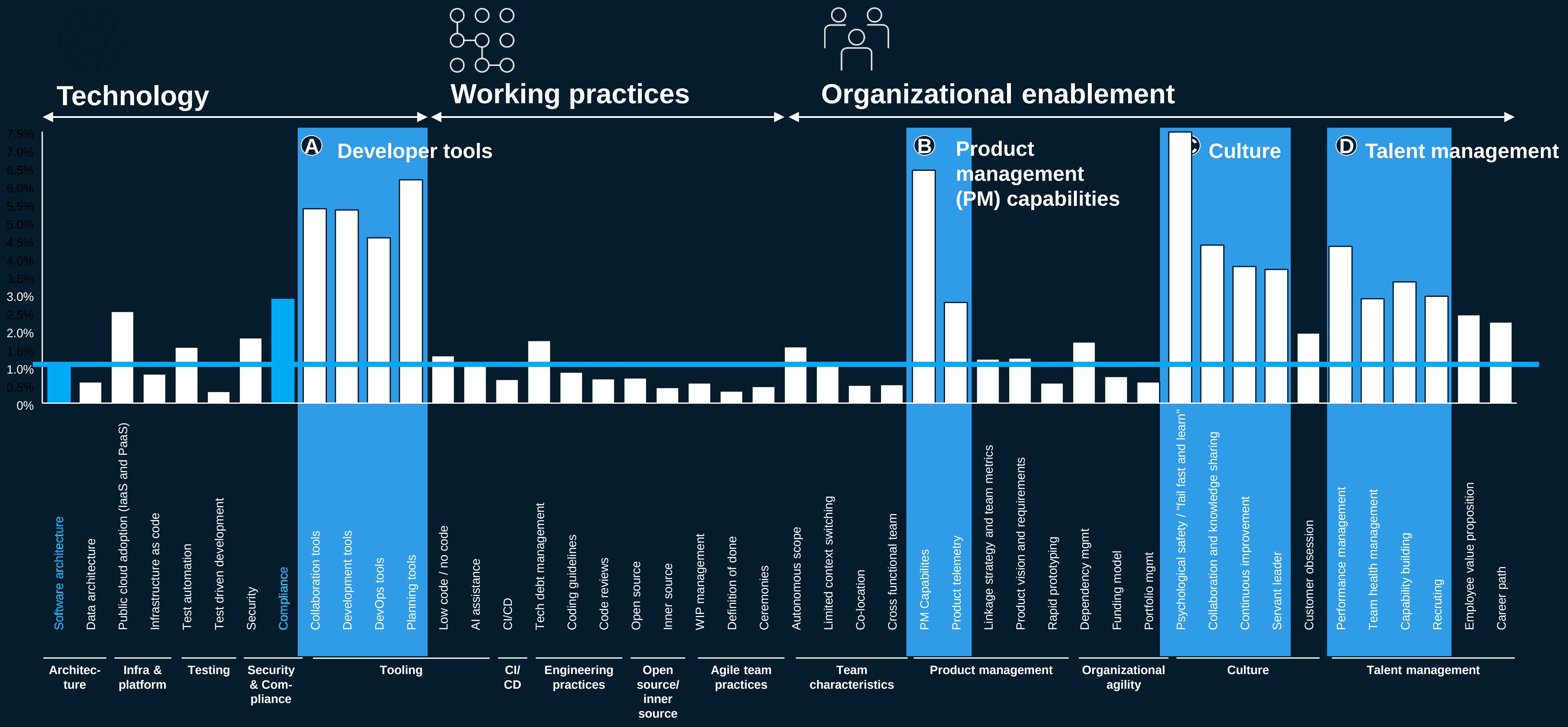
Culture

- Collaboration and knowledge sharing
- Continuous improvement
- Culture of customer obsession
- Psychological safety/fail fast and learn
- Servant leadership

Talent management

- Recruiting
- Employee value proposition
- Capability building
- Career path
- Performance management
- Team health management

DVI is calculated as a weighted average of the scores for the 43 drivers across 3 dimensions



1. Calculated using Johnson's Relative Weights: % importance is relative importance of driver on business outcomes. Total sums to 100%. Higher % indicates stronger impact on business performance

2. Average score for Innovation, Customer Satisfaction, Brand, and Talent

Source: McK Developer Velocity Survey, Expert Interview

□ Foundational drivers R2 = 0.6 N = 440

Drivers relative importance1 on overall business performance indicators2 % Importance1

Software is a people business

The key success factors

- Culture
- Talent Management
- Development Tools
- Product Management

Linus Torwald

Principal developer Linux Kernel

“”

The value of Software is not in the code, **its in the head of** the **people** who developed the code

McKinsey & Company

Thank You