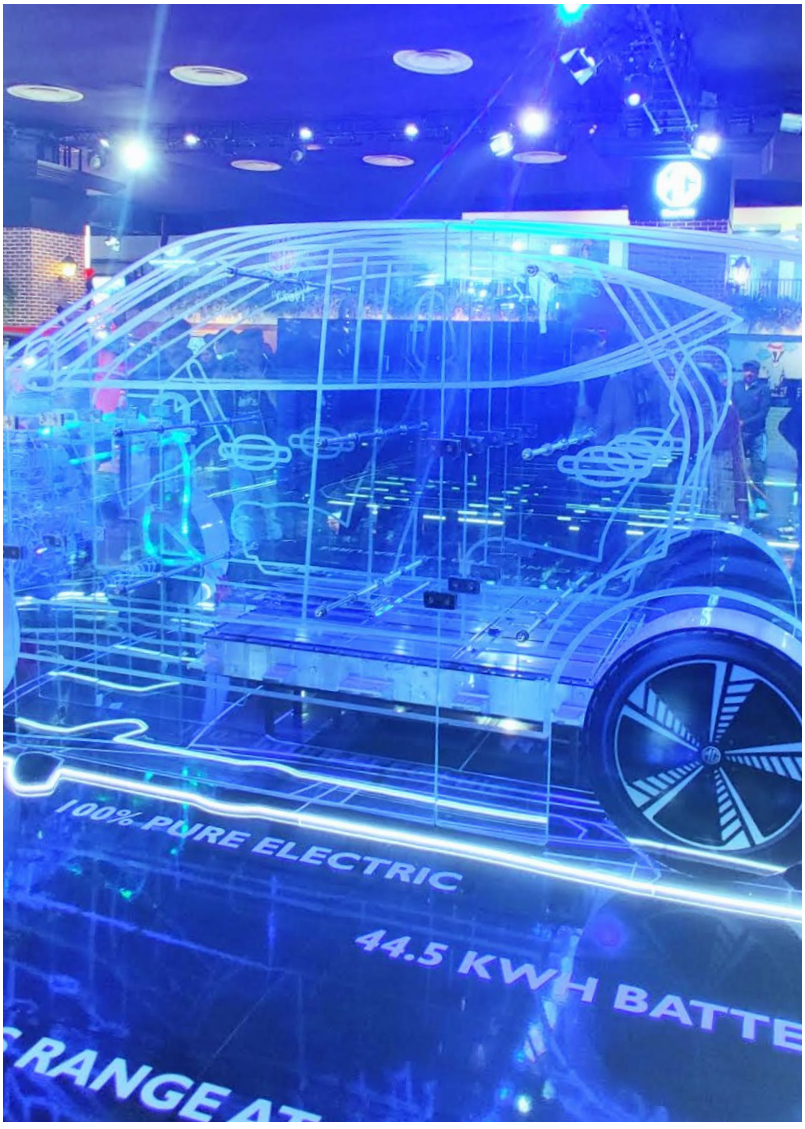
Photo by Raghav Basil
Kearney, Gurugram



Mastering the art of software

Unlocking the growth potential of cloud-enabled software-defined vehicles

BinaryCore | KEARNEY

Introduction 1

Automotive software: every day a new car—the power of cloud-enabled software defined vehicles 2

People and places: the driving forces behind a successful automotive software delivery footprint 5

Features and costs: how to calculate the automotive software budget for your ideal product portfolio 9

Excellence in end-to-end automotive software development 13

A guide to model-based systems engineering 17

Unlocking speed, quality, and cost: how to optimize hardware development in an embedded software environment 21

A new era for automotive business models 25

Introduction

"Software is eating the world." It's hard to imagine that even Marc Andreessen knew how true those words were when he made that statement 10 years ago. Mastering the art of software has become imperative for almost any business in any sector. Perhaps the most obvious example is the automotive industry, where today's cars comprise more lines of code in software than complex software-only products. Kearney's Automotive Software Benchmark Index shows that by 2030, traditional OEMs aim to derive more recurring revenue from software and connected services than from hardware unit sales. Improving software delivery capabilities is key to meeting essential strategic targets such as growth, higher make-ratio, and increased cost-effectiveness.

This, however, is easier said than done. Traditionally hardware-oriented companies are left chasing the same productivity levels and engineering prowess as software-native competitors. The difficulty is generally considered to stem from a fundamentally different engineering approach. Whereas hardware engineering mostly operates on a "first-time right" principle, software engineering embraces speed. Mark Zuckerberg's famous quote, "move fast and break things," encapsulates the approach. Another major challenge is the move from ambition to action. Organizations looking to integrate software stacks into their system environments cannot do anything without first building developer communities and coding libraries, and providing integration tools. For that, you need to attract best-in-class developers at a time when the war for software talent has never been so intense.

We believe that companies struggling to strengthen their software capabilities can learn a great deal from disruptors operating in traditionally hardware-driven industries. In this series of articles, we explore learnings from these challenger organizations and how they've come to be at the forefront of integrated hard- and software engineering. In the first article on software-defined vehicles, we examine how new customer requirements are mapping into software functions, creating the need for new product architecture.

We then go on to consider the implications of a reshaped product value chain—make vs. buy—and what it takes to build a new delivery model. The piece on software features looks at the cost considerations for value steering, while in "People and places: the driving forces behind a successful software delivery footprint," we examine how three different business profiles set up optimized global software delivery chains that consider their existing operating model, footprint, and competitive landscape. This article then goes on to outline five key steps to define your software delivery footprint, drawing on insight from Kearney's Global Services Location Index (GSLI) and data from more than 60 countries.

The fifth article addresses the optimization of product development across smart products, and how the synchronization of hardware and software unlocks speed, quality, and cost. We then provide a comprehensive guide to model-based systems engineering (MBSE). This covers what it takes to make the shift from a document-based approach and explains how MBSE can help organizations achieve a 20 to 30 percent reduction in time to market and a 10 to 20 percent reduction in development costs through team collaboration and model reuse. The outlined five stages of excellence describe organizations at different stages of the journey. The series concludes with an article on the need to look beyond traditional sourcing to joint business models as a new era beckons for the automotive industry.

We hope the articles included here provide valuable insight into some of the challenges you may currently be facing. To talk more about how we can help, or discuss any of the articles in more detail, please get in touch to carry on the conversation.

Automotive software: every day a new car—the power of cloud-enabled software-defined vehicles

Over the past decade, traditional original equipment manufacturers (OEMs) have struggled in adapting a digital mindset and techniques for their research and development in the pursuit of a software-defined vision. Too often progress stalls as inefficient, hardware-engineering-driven ways of working and outdated methods remain the status quo. But elsewhere in the market, we see that the vision o f a software-defined vehicle is possible.

In the first of a series of articles on automotive software, we detail how the meteoric rise of challenger OEMs and new entrants shows that software-defined vehicles have the power to make continuous innovation possible.

The core capabilities that enable digital product transformation and monetization are cloud native like in-vehicle computer platforms, the migration of car software across domains to a function-orientated architecture, and the ability to use microservices and containerized deployments. With frequent over-the-air updates and a focus on product development, not just bug fixes, the aim is vehicle feature releases multiple times a week. Improving the timeliness of in-vehicle updates, data collection, and management can keep a car digitally relevant and secure for 10+ years. Recent research of global manufacturers shows that cloud leaders adopt the cloud not just for cost savings, but also to grow revenue, accelerate innovation, improve teamwork, and shorten time to market.



As an example, Tesla can deploy hotfixes within 24 hours, including full integration testing and OTA updates across 80 percent of the fleet. Thanks to their software-first approach in product engineering and the vehicle architecture platform, they can regularly, and remotely, update models launched more than 10 years ago like telcos update mobile industry products. For this new breed of tech players, the ability to make real-time technical improvements not only increases the longevity of products, but it also unlocks new revenue streams through the monetization of applications, infotainment, and other software-enabled features while driving brand loyalty through customer-centric development and technology leadership.

Quantifying the performance gap: software companies achieve more for less

While it's true that traditional OEMs regularly manage to present new models with one or two innovative software features, these can come at a 50 to 70 percent higher cost and effort than the leading digital players. This is because there are typically between 30 and 40 different computer platforms integrated within these vehicles, many of which cannot be updated remotely thus limiting the ability to continuously deploy new software and enable pervasive monetization.

Several other factors make pushing the limits of domain architectures too costly for traditional OEMs. The variety of chips and dependencies, also fueling the semiconductor crisis, with more than 100 electronic control units on 30+ chipsets each with a heavily customized base operating system drives the implementation workload. Outdated ways of working result in large and inefficient, often still hardware-dominated software R&D organizations. These organizations might have as many as 3,000 developers only developing 20 to 30 percent of the software for the vehicle inhouse and sourcing the rest from tier 1 or 2 suppliers. In stark contrast, benchmark examples typically have 1,500 developers developing 80 percent of the software across all domains inhouse. Lastly, slow and hierarchical working cultures lead to losses in the war for software talent needed to fuel innovation. Classic OEMs know that change is necessary but may be lacking the knowledge, mindset, and partnerships to make the shift.

Bridging the gap: automation as key enabler along the software toolchain

As a first step, organizations should act to improve efficiencies within and at the interfaces between research, design, and development phases. Our recent benchmark study has shown that traditional OEMs can realize up to 30 percent reduction in the test and integration error rate, as well as a 20 percent decrease in time to market by reducing complexity through design simplification, validation via digital prototypes, automated engineering processes, and enabling the end-to-end responsibility of developers in a collaborative environment. When it comes to software testing, integration, and homologation, organizations should take advantage of installed fleets to test new software pre-release. Challenger OEMs and tech giants—such as Microsoft with its Insider program—use a "shadow mode" capability to pilot new software in the real world, and analyze performance before a final release. Continuous integration testing and the reduction of hardware prototypes through digital twins and simulation can improve validation efficiency by a 30 percent reduction in cost.

Improving software delivery capability is key to meeting essential strategic targets such as growth, higher make-ratio, and increased cost-effectiveness.

Leapfrog the competition: three approaches to develop the software-defined vehicle

There are three options for traditional OEMs to develop the software-defined vehicle: stepwise transformation over multiple architectures with the support of tier 1 suppliers, partnering with a strong tech company to drive the innovation process, and a greenfield approach to create an entirely new, cutting-edge platform (see figure).

Considering an automotive software-defined future?

Partnering with a technology company will help you keep pace with challengers while maintaining a low risk profile and with reliance on the partner to simplify and orchestrate a complex ecosystem. However, to truly leapfrog the existing state of vehicle IT, it's time to use the full potential of proven cloud technologies. Cut the threads to legacy and let's work together to develop your future platform enabling you to deliver a new car every day.

The authors are grateful to Thomas Mueller, Global CTO – Engineering, Research and Development Services, Wipro Limited, for his thoughtful insights and strong collaboration. Wipro is a leading global information technology, consulting, and business process services company, helping clients grow through engineering, cognitive computing, hyper-automation, robotics, cloud, analytics and emerging technologies.

Learn more at www.wipro.com/engineeringNXT.

Figure
Traditional OEMs have three options for developing the software-defined vehicle

	Tier 1 suppliers	Tech partner	Greenfield
Description	Electric/electronics transformation of architecture over multiple generations	Partner with major tech player to deliver future platform and adapt the rest to it	Cut legacy relations and create a state-of-the-art platform, way of working, toolchain, and organization
Speed	10–15 years →	5 years →	3 years →
Costs	2–3 major platforms	2 platforms and licensing	1 platform
Pros	— Low technical risk — Moves along with original equipment manufacturer transformation speed	— Balanced in risk and innovation — Fast capability uplift — Ecosystem orchestration and change management provided by partner	— Bold approach — Best practices can be used from new electronic vehicle players — Leapfrog existing players with newest approach
Cons	— Competition continues to pull away — Low degree of innovation — Costs of an extra 1–2 platforms in between	— Vendor lock-in — High degree of outsourcing and potential knowledge loss	— Higher initial invest — Complex cut-over phase into all models — Requires ringfenced organization and best experts in the market

Source: Kearney analysis

People and places: the driving forces behind a successful automotive software delivery footprint

The automotive industry is struggling to attract the best talent and build up research and design hubs for in-house software development.

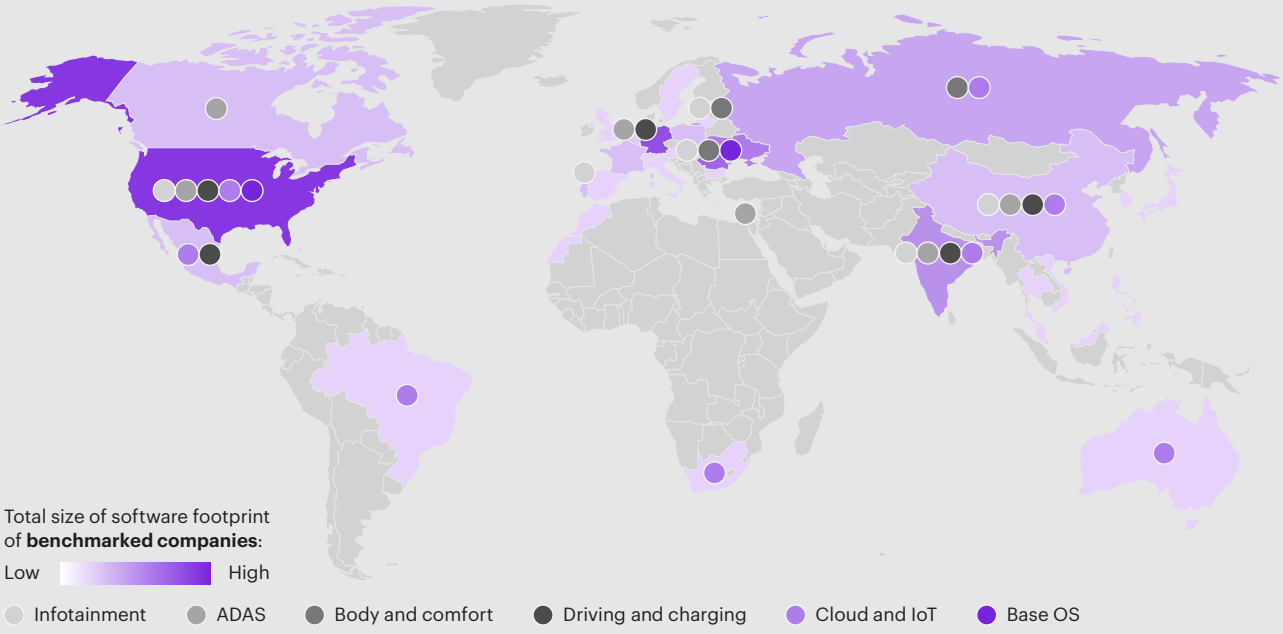
Over the past few years, a series of trends and seismic shifts have redefined the landscape. Today, established original equipment manufacturers (OEMs), new challengers, and tier 1 suppliers alike are navigating how to perform as software-driven companies. They know that improving their software delivery capability is key to meeting essential strategic targets such as growth, higher make-ratio, and increased cost-effectiveness.

With all the talk around emerging technologies and the pace of innovation, it can be easy to lose sight of the fact that people and places are still at the heart of this progress. Ultimately, the companies making waves in this space and leapfrogging competitors are those that focus on winning the ever-increasing war for tech and software talent. But every organization is different. Depending on the company's strategic ambitions and unique challenges, a careful balance must be struck between finding the right people and premises to achieve success (see figure 1 on page 15).



Figure 1

Global automotive software footprint showing the total size and focus domains of countries



Notes: ADAS is advanced driver assistance system. IoT is Internet of Things. OS is operating system.
Source: Kearney Automotive Software Benchmark Index

Key considerations when setting up a global software delivery chain

Today, automotive organizations everywhere are faced with the challenge of how to set up an optimized global software delivery chain that considers the existing operating model and footprint (domain-specific development, testing and integration), and the competitive set-up. Deciding on the right premises requires careful consideration of several key factors. Not only the draw for talent, value for money, and location manageability, but also whether the premises will create a long-term tech and software environment while improving core capabilities at pace.

Based on our Automotive Software Benchmark Index, we identified the preferred approaches for three different profiles: established OEMs, challenger OEMs, and tier 1 suppliers.

Established OEMs

In order to limit complexity and build on their existing reputation and recruiting channels, established OEMs tend to grow close to their headquarters, building software centers in countries such as Germany, the United States, and China. Due to the larger size (>3,000 developers) these are then supported by tech hubs around the world.

Challenger OEMs

In contrast, for challenger OEMs, talent drives footprint. Hiring where the best software talent is available comes first—prioritizing maximizing efficiency, which often comes with higher personal expenses per headcount. They focus on building a higher make-ratio (limiting outsourcing), with minimum personnel required (<2,000 developers). The superior output quality compensates for the higher staff cost. Tesla as an example hires 1,500 developers in the highly competitive area of Palo Alto. Although this comes at a considerable cost, they benefit from the transfer of knowledge within the Silicon Valley talent ecosystem. Here it's common practice for talent to rotate between roles at the tech giants, bringing with them cutting-edge expertise and a culture of continuous learning.

Tier 1 suppliers

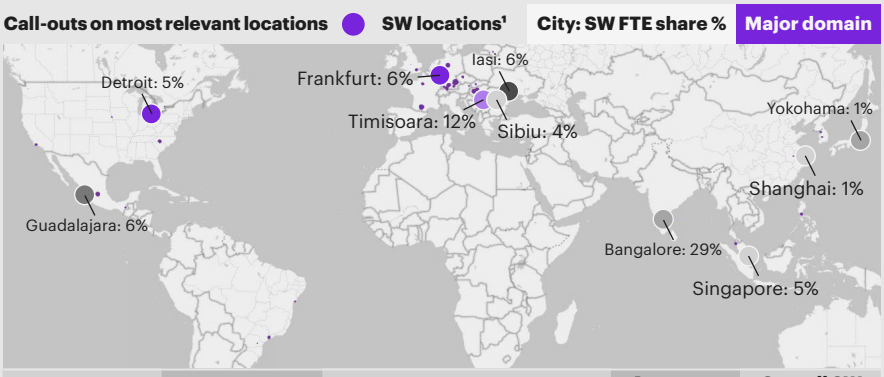
Overall, tier 1 suppliers (>10,000 developers) and volume OEMs look for the best value for money globally with a domain-specific footprint. They focus on nearshore locations in fast-follower centers such as Romania, Poland, Bulgaria, Portugal, and off-shore delivery centers such as Bangalore. This approach allows them to expand carefully, balancing the staff cost-to-talent ratio while realizing the best-cost country efficiencies (see figure 2).

Five steps to define your software delivery footprint

This best-in-class five-step approach is based on the competitor footprint and country overview from our Global Services Location Index (GSLI), which provides valuable macro- and microdata on broad economic and tech insights based on 47 metrics across four key themes and 15 sub-themes for more than 60 countries.

Figure 2

Example of software footprint and focus domains of tier 1 suppliers at a city level



Top four countries	Top cities	SW FTE Current	SW FTE Trend	Average SW FTE [k€/y]²	Overall SW staff costs [€ mn/y]
India	Bangalore, Pune	4,500 (29%)	↑	20	91
Romania	Timisoara, Sibiu, Carei	3,400 (22%)	→	23	79
Germany	Berlin	2,200 (14%)	↗	101	223
Mexico	Guadalajara, Santiago de Querétaro	1,300 (8%)	↑	29	38

- Infotainment
- ADAS
- Body and comfort
- Driving and charging
- Cloud and IoT
- Base OS

Notes: SW is software. FTE is full-time equivalent. ADAS is advanced driver assistance system. IoT is Internet of Things. OS is operating system.

Source: Kearney Automotive Software Benchmark Index

It also uses our Automotive Software Benchmark Index, where we have extended the GSLI to include software-specific factors, developer/architect costs, and competition footprint, providing our clients with valuable insights at city level.

— **Define strategic premises.** We jointly define the strategic macro and micro criteria to meet your ambitions. At a macro level these can include financial attractiveness, talent access, and ramp-up speed, and at a micro level a capabilities spectrum, future readiness, and scope extension options.

— **Select potential countries.** We conduct a country-selection process, starting with a global perspective and refining down to a manageable set of countries that fulfill the defined strategic macro criteria.

— **Evaluate options.** We evaluate the selected countries against the defined strategic micro criteria.

— **Investigate in detail.** We investigate the resulting locations in detail, assessing recent political developments, employee attractiveness, software company footprint, investments, and domain expertise.

— **Plan entry.** Lastly, we evaluate different entry options such as acqui-hiring, partnerships, investments, and acquisitions.

Improving software delivery capability is key to meeting essential strategic targets such as growth, higher make-ratio, and increased cost-effectiveness.

Features and costs: how to calculate the automotive software budget for your ideal product portfolio

Investing to meet customer expectations

A key area of today's transformation in the automotive industry is the transition of original equipment manufacturers (OEMs) into software-driven companies. Leading OEMs are forced into this transition due to the crucial role software plays in meeting the latest customer expectations. In the eyes of most customers, vehicles with cutting-edge features retailing at competitive prices are now the standard. You do not have to look far to see examples of OEMs making major investments in this space.

Volkswagen recently announced an investment of €89 billion into electric vehicles and software development over the next five years. Mercedes-Benz said it would invest €40 billion in electric cars, vans, and light commercial vehicles by 2030, and a further €5 billion into MB.OS.



Quantifying and prioritizing the right opportunities

While most OEMs are clear about the need for high software budgets enabling the continuous development and regular release of new features, they are less clear on what should be invested in and how to optimize the spend. For many, it's largely uncharted territory. The endless choice between different approaches as well as the often-limited software expertise in traditional automotive companies can make progress slow, or even send it in the wrong direction.

To fulfill customer expectations and start closing the gap between traditional OEMs and software-native challengers, the strategic planners of today must define a future-proof software strategy and portfolio. The approach must effectively quantify and prioritize the right opportunities, while also considering where resources are limited.

Kearney's comprehensive automotive software feature database

In response to more and more clients coming to us with this challenge, we worked with a range of suppliers and external experts to find an answer. We wanted to provide a concrete way of supporting our clients through a qualitative and quantitative assessment and prioritization of the software features in their product portfolios.

We developed a comprehensive database compiling more than 900 software features (everything from torque management to payment integration), covering in-vehicle and back-end domains on both an application and base operating system (OS) level (see figure 1). By adding different filters, the database compares features' characteristics, cost estimates, and potential optimization levers to identify the most efficient and effective solution. Our clients can now create an initial bill of materials and business case for feature development with moderate effort.

Figure 1

We compiled a comprehensive software feature set expanded by characteristics, costs, and optimization levers

KEARNEY — Automotive Sofware Feature Database

Domain	Comp. Cluster	Component	SW Stack	Feature	Characteristics						Cost estimate				
					Lifecycle	Technology stack	Automotive quality / confidence	Tech quality / confidence	QM/ASIL	Overall	Method	Average Player	MoB volume player	MoB premium player	Relevance
Body & Comfort															
		Body controller	Application	Passenger status (action, style, drowsiness, intention)	Differentiating	Automotive	Mid	High	QM	1,6	Benchmarking	100.000,00 €	Buy	Buy	High
				Middle console HMI	Commodity	Automotive	High	High	QM	1,2	Benchmarking	30.000,00 €	Buy	Make	High
				Roof console HMI	Commodity	Automotive	High	High	QM	1,2	Benchmarking	15.000,00 €	Buy	Make	High
				Garage door opener	Commodity	Automotive	High	High	QM	1,2	Benchmarking	10.000,00 €	Buy	Buy	High
				Signal horn	Commodity	Automotive	High	High	QM	1,2	Benchmarking	- €	Buy	Buy	High
				Drive mode select	Commodity	Automotive	High	High	QM	1,2	Benchmarking	75.000,00 €	Buy	Buy	High
				Anti theft system	Commodity	Automotive	High	High	QM	1,2	Benchmarking	incl.	Buy	Buy	High
				In-vehicle surveillance	Commodity	Automotive	High	High	QM	1,2	Benchmarking	incl.	Buy	Buy	High
				Keyless entry and start system	Commodity	Automotive	High	High	QM	1,2	Benchmarking	350.000,00 €	Buy	Buy	High
				Short range park assist	Commodity	Automotive	High	High	QM	1,2	Benchmarking	30.000,00 €	Buy	Buy	High
				UWB start system	Differentiating	Automotive	High	High	QM	1,4	Should costing	1.500.000,00 €	Buy	Buy	High
				Vehicle immobilizer	Commodity	Automotive	High	High	QM	1,2	Should costing	100.000,00 €	Buy	Buy	High
			Platform	AUTOSAR high	Differentiating	Automotive	Mid	High	QM	1,6	Benchmarking	4.150.000,00 €	Buy	Buy	High
				Bootloader	Commodity	Automotive	High	High	QM	1,2	Benchmarking	100.000,00 €	Buy	Buy	High
		Air conditioning	Application	Multi zone support (1/2/3/4-zones)	Commodity	Automotive	High	High	QM	1,2	Benchmarking	500.000,00 €	Buy	Buy	High
				Temperature management	Commodity	Automotive	High	High	QM	1,2	Benchmarking		Buy	Buy	High
				Air flow management	Commodity	Automotive	High	High	QM	1,2	Benchmarking		Buy	Buy	High
				Infotainment HMI integration	Commodity	Automotive	High	High	QM	1,2	Benchmarking	2.500.000,00 €	Buy	Buy	High
				Air quality management (Filter, ionisation)	Commodity	Automotive	High	High	QM	1,2	Benchmarking	50.000,00 €	Buy	Buy	High
				Remote control/start	Commodity	Automotive	High	High	QM	1,2	Benchmarking	200.000,00 €	Buy	Buy	High
				Independent vehicle heater	Commodity	Automotive	High	High	QM	1,2	Benchmarking	incl.	Buy	Buy	High
				Rear window heating	Commodity	Automotive	High	High	QM	1,2	Benchmarking	incl.	Buy	Buy	High
				Air sensor (temperature, quality)	Commodity	Automotive	High	High	QM	1,2	Benchmarking	incl.	Buy	Buy	High
				Windshield heating	Commodity	Automotive	High	High	QM	1,2	Benchmarking	incl.	Buy	Buy	High
			Platform	AUTOSAR low	Commodity	Automotive	High	High	QM	1,2	Benchmarking	1.500.000,00 €	Buy	Buy	High
				Bootloader	Commodity	Automotive	High	High	QM	1,2	Benchmarking	100.000,00 €	Buy	Buy	High
		Chassis & suspension	Application	Magnetic ride control	Commodity	Automotive	High	High	B	1,2	Benchmarking	2.500.000,00 €	Buy	Buy	High
				Air ride control	Commodity	Automotive	High	High	B	1,2	Benchmarking		Buy	Buy	High
				Driving modes	Commodity	Automotive	High	High	A	1,2	Benchmarking		Buy	Buy	High
				Chassis level control	Commodity	Automotive	High	High	A	1,2	Benchmarking		Buy	Buy	High
				Predictive damping	Differentiating	Automotive	High	High	A	1,4	Should costing		Buy	Make	High
				Active vertical dynamic functions	Differentiating	Automotive	High	High	A	1,4	Should costing		Buy	Make	High

Sources: Industry benchmarks, expert interviews; Kearney analysis

The database can be used in the following ways:

White spot identification. It's quick and easy to compare your current software feature portfolio with the database and uncover key features you may be missing in your software plan.

Feature analysis. Each feature included in the database is characterized by a set of criteria assessing the difficulty of implementation. These criteria are:

1. **Life cycle.** There are three defined life cycle stages: common, differentiation, and innovation. These denote the different impacts each feature will have on the value-add for customers.
2. **Technology stack.** There are three ways of assigning whether an organization already has the expertise required for new features or whether it will need to be leveraged from other industries: common, automotive, and high tech.
3. **Implementation confidence.** There is a scale from low to high rating the quality and confidence of automotive and tech players in feature implementation.
4. **Automatic Automotive Safety Integrity Level (ASIL).** The ASIL considers functional safety implementation and testing efforts.

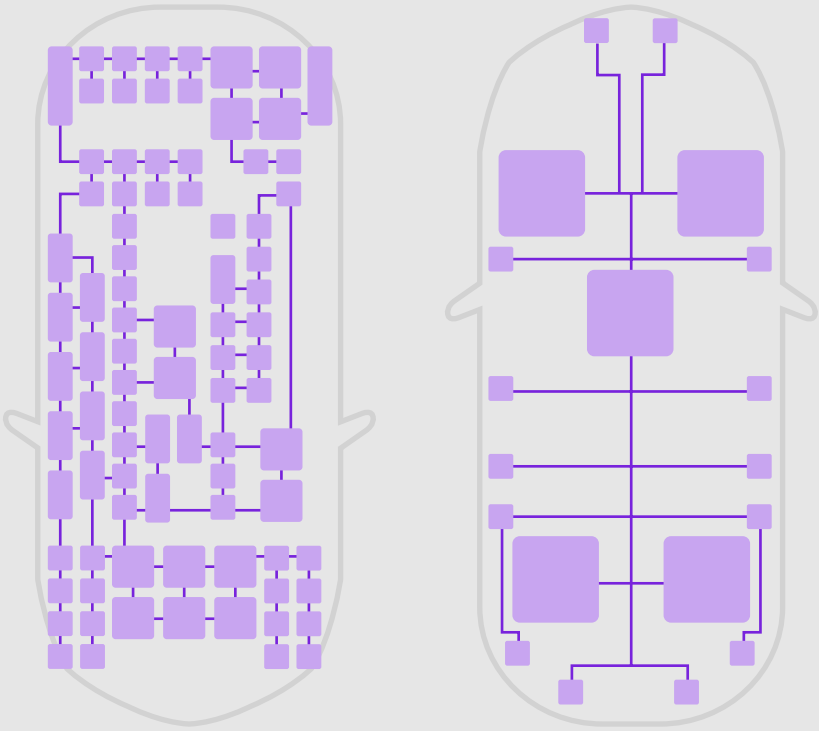
Cost estimate and validation methodology. For budget planning and prioritization, each feature must have a clear cost estimate. We provide an initial estimate based on expert interviews and our best fit should costing methodology (should costing is an analytical method that benchmarks expertise to compare developer and practitioner estimates for features that have not yet been developed).

Strategic make-or-buy decisions. Once you've assessed the features' life cycle, costs, and customer value-add, you're in a good position to make strategic make-or-buy decisions. The choice of sourcing partner (for example, tier 1 suppliers or tech players) will have a significant impact on revenue shares in a buy case.

Optimization levers. Optimization levers can be used as hypotheses to further improve feature planning and strategy. As examples, these might include adjusting shoring models to deliver commodity features offshore, or the reprioritization of features based on revenue estimates.

Strategic planners of today must define a future-proof software strategy and portfolio.

Figure 2
Example of how the database helps quantify the cost and savings of moving to a zonal architecture with fewer ECUs and a shared operating system



Source: Kearney analysis

How to use the software feature database for a platform cost comparison between two architectures

Due to decentral or domain-based E/E architectures, some OEMs are easily running on 60 to 120 ECUs, sensors, and actuators in one car. Below we outline how the database helps quantify the cost and savings of moving to a zonal architecture with fewer ECUs and a shared operating system (see figure 2).

An OEM wants to compare the base OS costs for a domain-based architecture versus a zonal architecture. They can set a filter in the database to electric vehicle, in-vehicle domains, and exclude sensors/actuators, leading to around 40 electronic control units (ECUs) at €115 million base OS costs for a domain-based architecture.

For the zonal architecture an additional filter can be applied on high-performance compute module readiness, which then reduces the required ECUs to five high-performance ECUs at either €5 million cost for one common OS or €25 million in the case of individual versions.

This clearly illustrates the potential and serves as an initial indicator in the business case of a zonal architecture for platform cost reductions. In a second step the cost increase in software management and project management office can be quantified to sharpen the calculation. In general, the reduction in platform costs opens budget for further improvements at the application layer, ultimately enhancing the customer experience.

Excellence in end-to-end automotive software development

Over the next 5 to 10 years, challenger original equipment manufacturers (OEMs) are increasingly likely to derive more recurring revenue from software and connected services than from hardware unit sales.

This transition is accelerated by establishing new business models such as features-on-demand (the capability to remotely add improved options such as seat heating, better headlights, and increased engine horsepower), which by itself could provide 30+ percent additional revenue potential by 2025, according to our market perspectives on car software service monetization gathered during our automotive software benchmarking exercise.

To meet these targets and remain competitive in a sector crowded with challengers and new entrants, traditional OEMs must strengthen their software capabilities and enable their architecture toward the software-defined vehicle. The goal is to have a high-performance continuous integration/continuous deployment (CI/CD) software pipeline that can automate builds, regression testing, and global over-the-air (OTA) fleet deployment within 24 hours (see figure 1 on page 7).

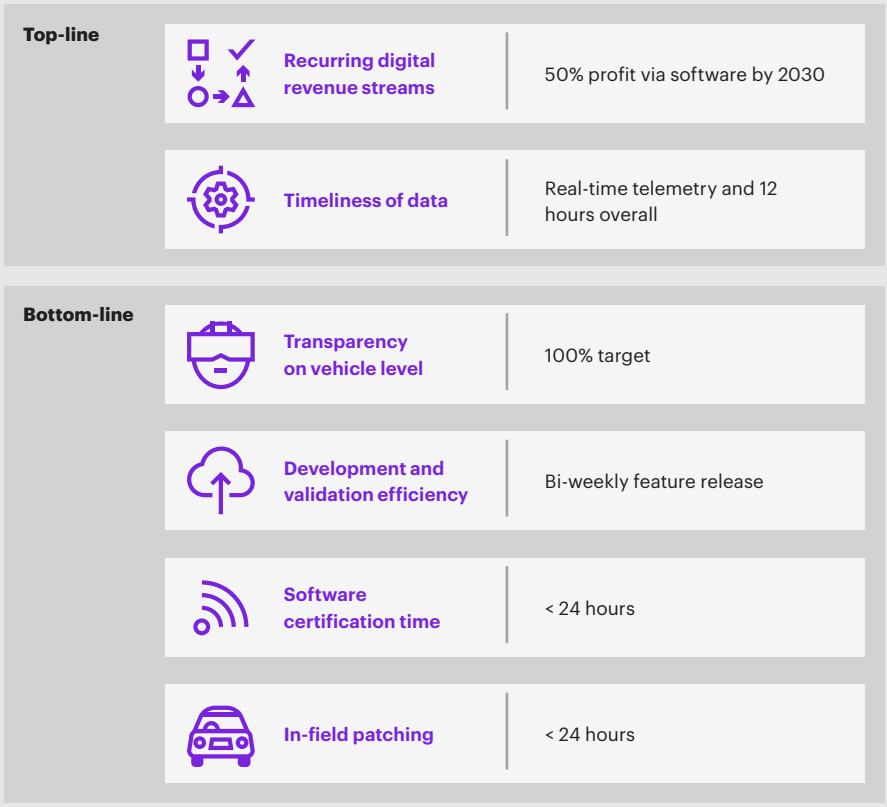


Fueled by our software experience, insights from various challenger OEMs with a deep dive on Tesla, and software company benchmarks, we designed an end-to-end software development approach that addresses the top and bottom lines. It provides the required baseline to enable new software-driven business models while saving costs. The example of a leading electric vehicle company clearly demonstrates the advantages of this structure. When they had to recall 53,000 vehicles due to concerns over the rolling-stop function, they were able to deploy an over-the-air software fix in 24 hours, resulting in zero physical recall costs.

The shift from process-focused to continuous delivery models

To build these capabilities, traditional OEMs need to shift from the typical "V" model to a continuously evolving delivery model synced to in-field operations. Our end-to-end software development approach includes 15 software excellence enablers that empower traditional OEMs to transform and remain competitive (see figure 2 on page 8).

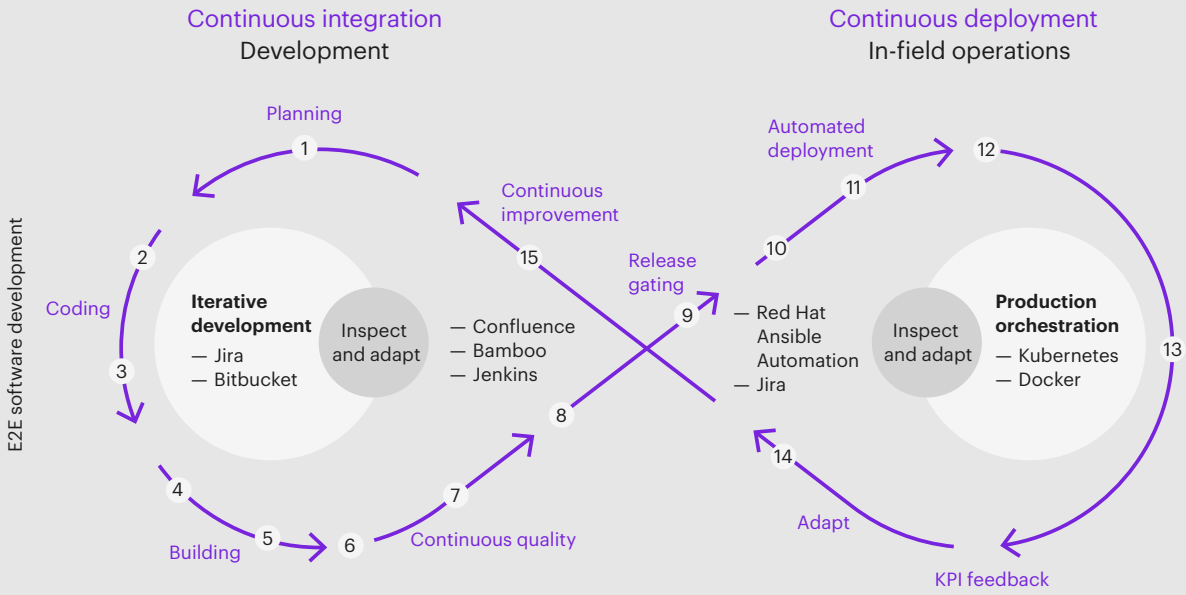
Figure 1
Certain metrics can help challenger OEMs master vehicle software development excellence



Source: Kearney analysis

Figure 2

15 software excellence enablers can empower traditional OEMs to transform and remain competitive



Source: Kearney analysis

15 automotive software excellence enablers

Development

1. End-to-end integrated requirement engineering

2. Feature-driven agile software development

3. Feature-driven software design and architecture

4. People-empowered software development

5. Integrated ticket-based toolchain collaboration

6. Monolithic in-field branching structure

7. Release-ready end-to-end testing and integration

Delivery

8. Compliancy of regular in-field software updates

9. Standards and software safety

10. In-field continuous integration pipeline

11. Bi-weekly feature releases

12. Global scalable over-the-air architecture

Continuous improvement

13. Software- and feature-driven monetization

14. Value-adding data collection

15. Real-time digital twin representation

Embedding this approach has a huge impact on the speed, quality, and cost of software development. Referencing our Automotive Software Benchmark Index, we also see that it enables challenger OEMs to build software organizations with 1,500 full-time equivalents (FTEs) developing operating systems and connected services on an annual budget of less than €500 million with an in-house make ratio of more than 80 percent.

Three key enablers of end-to-end automotive software excellence

Three specific enablers stand out for helping OEMs achieve major differentiation. First, feature-driven software design and architecture enables continuous software updates and reduces risk of delayed feature releases. With the mentality that everyone can contribute, an efficient resource pool, and the elevated role of the integration engineering team, the development, integration, and validation of the feature can all go hand-in-hand. This is why highly qualified integration engineers are the most in-demand talent for challenger OEMs.

Next, a people-empowered software development approach means that 1,500 FTEs are sufficient for all software-related functions. The choice of developers, lean teams set up by the best engineers, and the ability of a single committee to make technical and financial decisions are essential to success.

Lastly, release-ready end-to-end testing and integration allows more than 80 percent of the in-field fleet to receive any kind of software update within 24 hours. In urgent cases and with more restrictive variant prioritization, this can come down to as little as eight hours. The five stages of testing—functional, integration, system, regression, and acceptance—are all steered automatically by tools such as Jenkins and require no manual coordination.

On your way to end-to-end automotive software development excellence?

We understand the complexity of such a transition and its interlinkage with hardware development. We also know that a simple copy and paste of existing models from one to another company will not succeed. Our transformation approach enables you to establish the software development capabilities your company needs, making software development both ready to scale and a competitive advantage. To talk more about your ambitions and how we can help, please contact the authors.

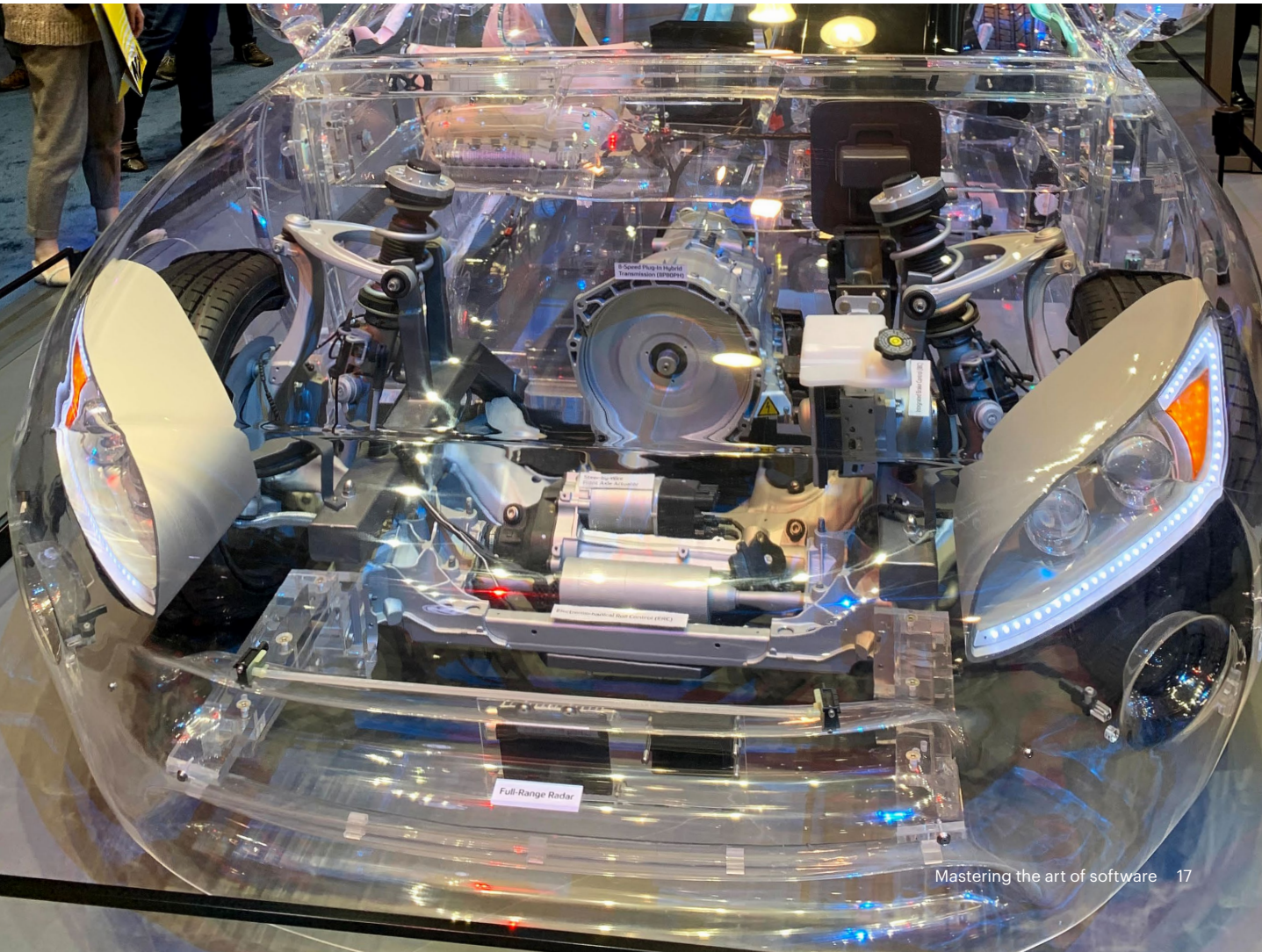
Feature-driven software design and architecture enables continuous software updates and reduces risk of delayed feature releases.

A guide to model-based systems engineering

Why now?

The automotive industry is changing fast and OEMs are under intense pressure to meet evolving customer expectations and stay ahead of the competition. However, current deficiencies in product development are posing a real obstacle to their profitability, innovation, and growth. The extreme complexity of the process and the associated data and information management present a need for higher levels of transparency, agility, and compliance. As development cycles across the automotive landscape become shorter, all players are moving to innovate faster.

To tackle these challenges, highly integrated tools and automated processes are key. Organizations must shift from document-based to data-driven and model-based systems engineering (MBSE). This approach includes connected data layers for requirements, functions, logical, and physical (RFLP) to manage traceability throughout the development process, developed on data-driven, model-based innovation platforms with advanced design, engineering, and manufacturing capabilities as well as enabling third-party tool integration, supplier authentication, and authorization management on a need-to-know principle.



The shift from document-based to model-based development

The move from document-based to MBSE is a move from decentral data pieces to a central interactive model, a scattered application landscape to a fully integrated environment, and limited engineering tool integration to full product life cycle management (PLM), virtual validation, and simulation orchestration. Switching to MBSE is not merely a technical or organizational transformation, but requires a comprehensive mindset change across all teams. Organizations only reap the rewards of the approach when they embrace customer-centricity over technology-centricity and deploy function-oriented rather than hardware-oriented system thinking.

According to research for Kearney's Automotive Software Benchmark Index, MBSE can reduce prototyping hardware costs by 50 percent, shifting from real-world to stimulated testing. This major benefit sits alongside a 20 to 30 percent reduction in time to market by enabling continuous integration and testing, and a 10 to 20 percent reduction in development costs through collaborative teamwork and model reuse.

Assess your current progress to define the steps ahead on the MBSE transformation

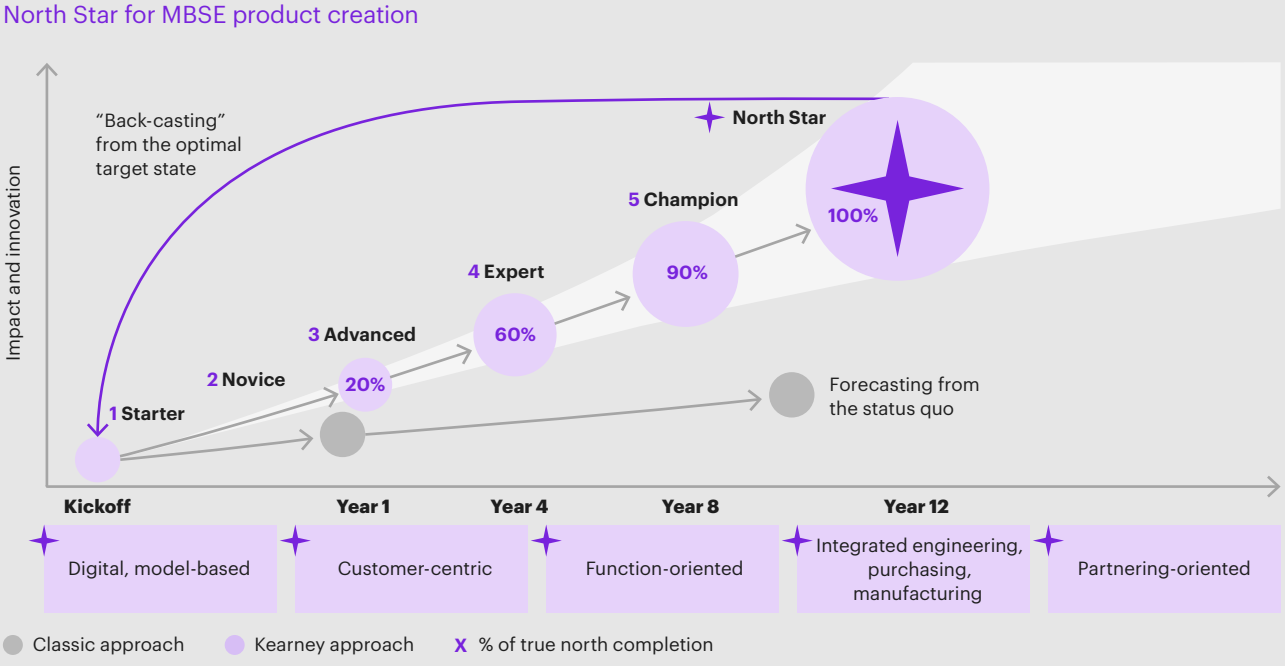
The five stages of excellence below describe organizations at different stages of the journey toward an optimal target state, from "Starter" at the kick-off phase to "Champion" from year 8 onward (see figure).

Stage 1: Starter

At this stage, organizations are at the beginning of their MBSE journey. They have investigated and identified the need for new ways of working and are willing to transform. The initial assessment is conducted along the V-model process covering requirements engineering, development, testing, and integration of the product including the underlying toolchain.

Based on these insights, an initial road map and target KPIs are defined for the key research and development (R&D) domains. This stage is especially relevant for traditional players with matured development processes looking for change. It will not just provide an overview of the status quo, but also a first indication of whether a green- or brown-field transformation is the better option.

Figure
An aspirational North Star is defined to push the boundaries and avoid incremental thinking



Source: Kearney analysis

Stage 2: Novice

The case for change is set and the transformation road map needs to be defined. To gain know-how and verify a company-specific hypothesis, a proof of concept (PoC) is initiated. Ideally, this involves the implementation of a system or sub-system with MBSE methodology and tools. These systems provide sufficient modeling complexity while not interfering too much with the overall vehicle development.

Again, this stage is highly relevant for traditional players looking into the approach before committing to the full transformation journey. For start-ups, the PoC is framed as the development of their first minimum viable product (MVP). It's crucial to gain hands-on experience with MBSE to embed new ways of working within the company.

As development cycles across the automotive landscape become shorter, all players are moving to innovate faster.

Stage 3: Advanced

Advanced users have successfully analyzed the status quo and gained the first insights from one or multiple PoCs. They have used best practices and learnings to define their systems engineering methodology regarding requirements engineering (RE) and trace-ability throughout the development process. Based on the newly defined methods and process the R&D toolchain simplification and transformation is approached.

This level of implementation is typically found within traditional OEMs and tier 1 suppliers. It is driven by recent compliance challenges with a strong focus on the systems engineering process.

Stage 4: Expert

Expert players have successfully implemented MBSE methods and tools for parts of the R&D business (for example, hardware development). They already realize substantial savings in the region of a 20 to 30 percent reduction in development costs. The digital representation of the product, as in the example of hardware, is modeled in a central CAD database as an interactive digital twin. Internal and external engineers have access to the collaboration platform, enabling real-time simulation and analysis of committed changes.

We observe the entry into this level at large-volume OEMs. These changes are driven by pricing chal-lenges and realize savings by extending simulation capabilities, reducing physical prototyping, and fostering early conflict detection within the develop-ment process.

Stage 5: Champion

The champions and forerunners in this space have fully adopted the method and harmonized the toolchain over all R&D domains. They have imple-mented a fully fledged MBSE suite covering the complete PLM process. The organization itself has transformed its mindset to become customer-centric and function-oriented, alongside the defined systems engineering methodology.

Organizations at this stage are typically battery electric vehicle (BEV) start-ups with a limited budget and enormous pressure regarding time to market and innovation. As an example, NIO managed to launch the ES8 SUV within three years of development, rather than the common automotive 5-to-6-year phases. This was thanks to simulation, seamless collaboration, and virtual production planning.

Three key areas define your road to success

1. Processes and methods

When it comes to processes, taking an end-to-end, function-oriented perspective is key. Continuous interaction with customers requires new forms of prototyping, while the model-based approach needs new, digital ways of developing the product.

What are the benefits?

— Decreased risk of not meeting customers' expectations

— Improved quality and maturity of increasingly complex products

— Improved compliance with internal and external regulations

— Decreased vulnerability of data and unauthorized access to IP

— Increased product value through innovation

— Accelerated development time

2. Governance

This approach requires new steering methods that enable cross-functional collaboration. Defining the standards of governance across the organization and RFLP data model dramatically increases transparency.

What are the benefits?

— Improved transparency regarding maturity through automated reporting

— Fewer interfaces and complexity through harmonized tools

— Higher level of collaboration and resource efficiency

— Higher organizational efficiency and improved product compliance

3. IT and toolchain

To create a seamless toolchain across all disciplines a simplified IT infrastructure, built on highly integrable solutions, is needed.

What are the benefits?

— Reduced cost compared to current complex and heterogenous IT landscapes

— Reduced lock-in risk through standardization of solutions

— High level of automation and real-time data availability

— Improved collaborative capabilities within and outside the organization

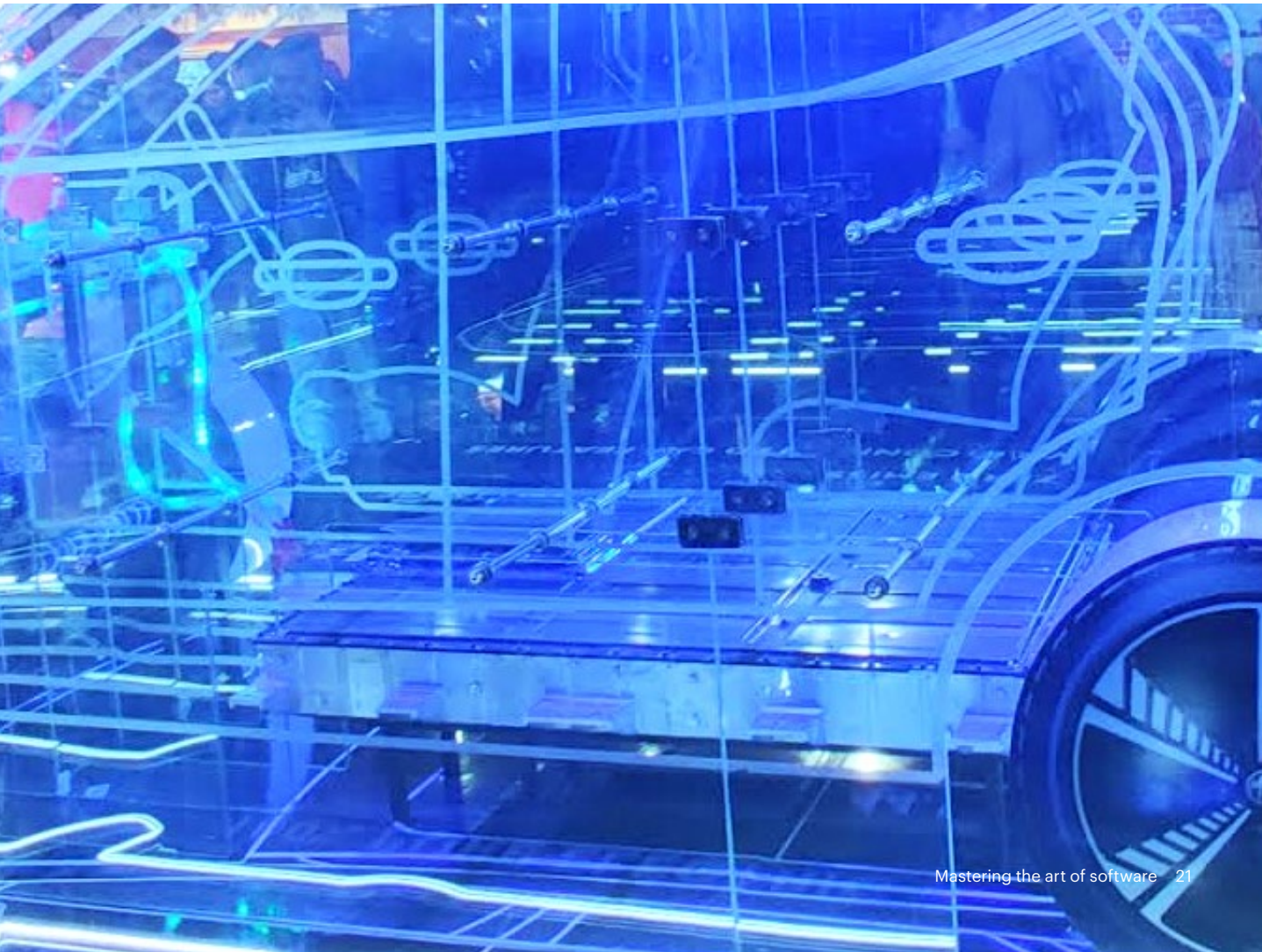
To create a seamless toolchain across all disciplines a simplified IT infrastructure, built on highly integrable solutions, is needed.

Unlocking speed, quality, and cost: how to optimize hardware development in an embedded software environment

So far in this series, we've covered many things around automotive software, from the prioritization of features to end-to-end software development. Now we turn our attention to the role of hardware within an embedded software environment and how challenger OEMs are rapidly deploying software to ECUs (electric control units) as in-vehicle hardware. In this article, we look at the challenges this presents to established OEMs, the fully automated delivery model disruptors use to stay ahead, and the key levers enabling excellence in end-to-end hardware development.

Challenger OEMs outpace traditional players with automated deployment and synchronized updates

For many OEMs, ECU provisioning is still a manual process. It's a common occurrence to see engineers physically connecting to test stands for flashing ECUs, staging test sets, and playback reports. This time-consuming method of deploying software can take days and sometimes even weeks if the right hardware isn't on hand. In contrast, challenger OEMs use the advantage of OTA (over-the-air) update infrastructure already in testing, making it a background process.



Leaders in this space ensure the latest software version always fits the current hardware-in-the-loop (HIL) by syncing hardware and software updates in good configuration management and fixed update cycles. If a hardware update has taken place, the software developers must know to push the corresponding software. In the same way, if new software has been developed for the base system but the new hardware has not been tested, the lack of compatibility can lead to major issues. To prevent this from happening, challenger OEMs operate on a tight schedule with strict deadlines, making software updates consistently every two weeks within a six-week hardware update cycle.

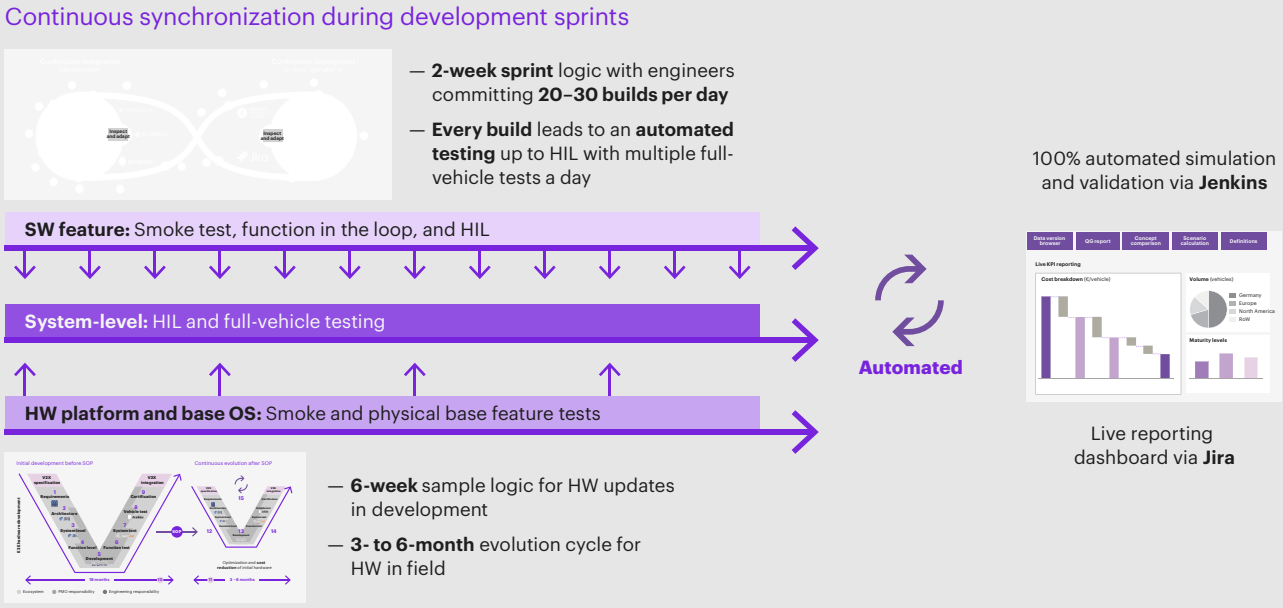
In the absence of a carefully regulated system and supplier dependencies, traditional OEMs may only sync hardware and software systems every three months. Often, they are left waiting on suppliers operating on separate toolchains, limiting the speed at which updates can be made. While OEMs don't need to develop everything themselves, by taking on the role of software integrator they enjoy the benefits of building the overall software package and testing it with the hardware in-house. This integration massively shortens the test frequency gap between competitors, avoids delays and potential crashes, and allows OEMs to develop in a truly agile manner.

Tech players and the advantages of a two-speed delivery model

Tech players in the sector are operating in a two-speed delivery model to operate like software companies. For the slower hardware platform and base OS development, we observe a traditional waterfall approach along the V-model with a six-week sample logic. Having a slower pace for hardware platform development and a faster pace for software feature development creates a stable platform and provides the foundation for greater innovation at the application layer. Tesla is known for optimized use of this model, where their two pipelines integrate into the central system more frequently for software features than for hardware features (see figure 1).

Figure 1

A slower pace for hardware platform development and a faster pace for software feature development creates a stable platform while providing the foundation for greater innovation at the application layer



Notes: HIL is hardware-in-the-loop. HW is hardware. OS is operating system. SOP is start of production. V2X is vehicle-to-everything.

Source: Kearney analysis

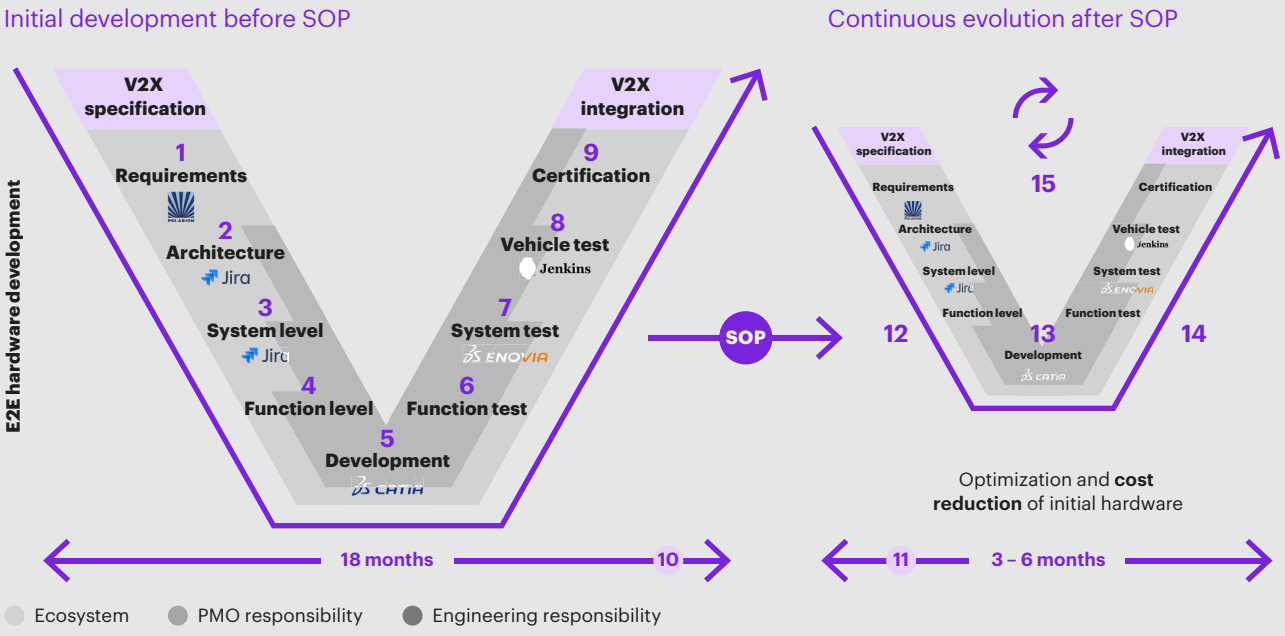
One of the major advantages of this model is that it unlocks huge efficiencies in time and cost. Because the system integration level is entirely automated regarding provisioning and regression testing, it allows for a lean maintenance team of a few people per model servicing around 50 test stands. With this setup, application developers can regularly commit software into the repository, triggering around 20 to 30 full vehicle software builds per day. To put this in context, traditional OEMs might have a same-size team servicing only one test stand and deploying software builds somewhere between every two weeks and three months.

Software builds using this model are mainly tested on HIL, where ECUs are connected to the central computer which simulates all the data without having to use the full vehicle. Additionally, challengers are able to perform up to two full-vehicle regression tests every day. Such high test frequency combined with the fixed regression test and resulting transparency, connected to the Jira requirements, allows for near real-time transparency on maturity grade and progress reporting. Without this overview and insight, traditional OEMs are more likely to approve every new software that comes in, often spending huge amounts of money without being able to measure the quality or qualify the impact.

Best-practice end-to-end hardware development

Referencing our extensive Automotive Software Benchmark Index, we identified two optimal V-cycle time lines for end-to-end hardware development and 15 key levers along the way. In this process, the initial ECU development before the start of production (SOP) takes up to 18 months, compared to a traditional setup of 24 to 30 months (see figure 2).

Figure 2
Best-practice E2E hardware development



Notes: E2E is end-to-end. SOP is start of production. V2X is vehicle-to-everything.
Source: Kearney analysis

15 key levers

Pre-SOP specification

1. End-to-end integrated requirement engineering
2. Zonal ECU architecture
3. Early expert validation within product teams
4. Hardware over specification on parts and power overhead

Pre-SOP development

5. Tooling and central CAD model storage

Pre-SOP integration and testing

6. Hardware and software synchronization
7. In-house testing and rapid prototyping
8. Automated full-vehicle testing
9. Risk-based shortening of industrialization and homologation
10. 18-month development cycle in sample logic

Post-SOP optimization and updates

11. Continuous evolution and optimization based on usage data
12. Re-specification of systems in vehicle behavior board
13. Minor and major product changes committed to the central model
14. Integration, testing, and self-certification
15. In-field hardware upgrades and central configuration management

Challenger OEMs and the continuous optimization

Of these 15 levers, some of the most crucial fall within post-SOP optimization. At the start of production, ECUs and mechanical hardware are often over-specified in power and durability, leading to part redundancy. Once released, actual tester and customer field data is used to optimize these parts and derive updates. These updates are then released in a three- or six-month cycle, depending on the scale of the change.

The front trunk closing system is a great example of this continuous optimization. When initially launched, the original setup was for 10,000 openings. However, Tesla soon saw from user data that the function was only being used in the first three months of ownership when customers were showing it to friends. Based on this insight, the durability of the closing system was reduced for future models by a factor of 10, to around 1,000 openings and closings. As well as significantly reducing costs, this optimization models true customer centricity, innovating around the customer by understanding exactly what they do and don't need.

The same principle holds for ECU optimization and part reduction. At the start of production, challenger OEMs will double vital parts on the ECUs just in case one of them breaks. In the event of a fault, the backup part can be activated via software update. Similarly, if the parts prove good enough, they will be stripped down to the most efficient configuration. Where more traditional OEMs tend to optimize before the start of production, at times spending more than a year optimizing components, challenger OEMs are free to add as many features as they like over time, without being forced into a trade-off. This approach is a game-changer in the production of hardware and the reduction of time-to-market.

A new era for automotive business models

Our analysis shows that by 2030, leading OEMs aim for up to 50 percent of their profits via recurring digital revenue streams and connected services. To unlock these potentials, traditional OEMs need to first reimagine their role and relationships within the automotive ecosystem.

Currently, their primary role is to assemble the vehicles, sourcing up to 80 percent of the materials from suppliers. Conversely, disruptors such as Tesla have a make-ratio of around 80 percent. For traditional OEMs to create more customer-focused products and services, the answer is not to build everything themselves, but to work much more closely with those they buy from. In this way they establish their place in the value chain as deep tech integrators and develop win-win collaborations with suppliers, ushering in a new era of automotive revenue.



Why collaboration is essential

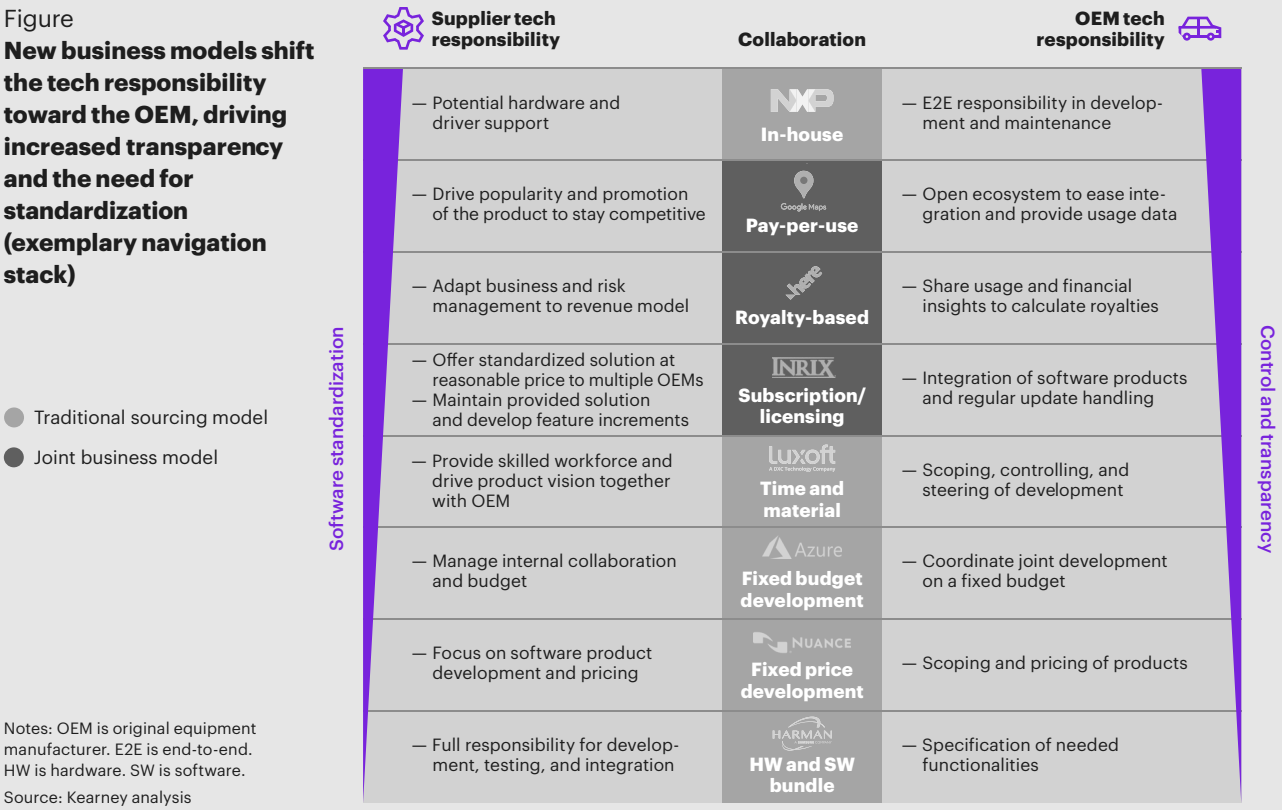
While challenger OEMs partially operate at a much higher make-ratio for software and hardware, they mostly focus their sourcing activities purely on non-differentiating features with potential cost savings directly from suppliers. To remain competitive, traditional OEMs must develop platforms that enable deep tech integration for software suppliers, providing them with direct access to as much as 80 percent of the car's systems and the plug-and-play connectivity of new features. Only by inviting closer collaboration with suppliers can these OEMs match the speed at which challengers continually meet customer demands, increase revenue, and reduce costs.

Looking beyond traditional sourcing to joint business models

With our extensive experience in this sector and insights from software company benchmarks, we see a significant opportunity presented by the adoption of new collaborative business models. By using the open vehicle platform ecosystem, OEMs and suppliers can achieve a revenue increase of 40 percent by 2030.

This new era will be shaped by OEMs and suppliers together, working toward mutual profitability increase. Shared success relies on the clear division of responsibilities, where OEMs have control over the right vehicle platform and suppliers are enabled to establish a certain degree of software standardization to build upon (see figure).

Figure
New business models shift the tech responsibility toward the OEM, driving increased transparency and the need for standardization (exemplary navigation stack)



Notes: OEM is original equipment manufacturer. E2E is end-to-end. HW is hardware. SW is software.
Source: Kearney analysis

Pay-per-use

Adopting a pay-per-use business model allows for an increasing scope of both commodity and differentiating vehicle features. The open ecosystem provided by the OEM enables easy feature integration as well as in-field data collection. The ownership and further usage of the data can lie with the OEM, supplier, or both, depending on specific features and the possibility of value creation for each company. Alongside this, suppliers have the chance to drive promotion of the product to stay competitive and decrease unit costs.

Royalty-based

Because the OEM must provide financial insight for the fair calculation of royalties, a trusting relationship between the OEM and supplier is a prerequisite for royalty-based business models. As a result, suppliers can better adapt their business and risk management models. A profitable business for the supplier will ensure its longevity and provide the necessary resources for further development. Meanwhile, the OEM maintains consistent revenue and profit flow.

Subscription/licensing

Lastly, a subscription/licensing model allows OEMs to achieve an even more efficient bottom line. Suppliers can scale commodity features beyond customized OEM solutions toward standardized solutions for multiple OEMs. The role of the OEM is reduced to the integration of software products and handling of regular updates while they benefit from a reduced cost per unit.

Interested in exploring new business models?

We understand the complexity involved in the adoption of these new business models. We believe success requires a cross-company approach with a deep understanding of what is possible for your organization. To talk more about your ambitions and how we can help, please contact the authors below.

By using the open vehicle platform ecosystem, OEMs and suppliers can achieve a revenue increase of 40 percent by 2030.

Authors



Michael Römer
Partner, Kearney and Founder, BinaryCore, Munich
michael.roemer@kearney.com



Marcus Weber
Partner, Munich
marcus.weber@kearney.com



Felix Kreichgauer
Partner, Munich
felix.kreichgauer@kearney.com



Sebastian Werner
Head of Software Engineering, BinaryCore, Munich
sebastian.werner@binarycore.com



Andreas Foitzik
Consultant, Munich
andreas.foitzik@kearney.com



Marcin Kurzal
Consultant, Munich
marcin.kurzal@kearney.com

About BinaryCore

BinaryCore was born from a need to tackle complex technology challenges businesses face in an era when realizing the full value potential of software engineering and cloud usage will clearly distinguish the leaders from the laggards. But addressing these challenges requires radical step change—a willingness to unlearn what went before, an openness to new possibilities and partnerships, and the confidence to make bold decisions and rewrite the rules.

With deep technology expertise, powerful proprietary software, a sharp focus on the numbers, and a clear set of rules, we'll guide you every step of the way to transform your technology and talent set and put in place the infrastructure and teams needed to set you up for long-term success.

www.binarycore.com

About Kearney

As a global consulting partnership in more than 40 countries, our people make us who we are. We're individuals who take as much joy from those we work with as the work itself. Driven to be the difference between a big idea and making it happen, we help our clients break through.

kearney.com

For more information, permission to reprint or translate this work, and all other correspondence, please email insight@kearney.com. A.T. Kearney Korea LLC is a separate and independent legal entity operating under the Kearney name in Korea. A.T. Kearney operates in India as A.T. Kearney Limited (Branch Office), a branch office of A.T. Kearney Limited, a company organized under the laws of England and Wales. © 2022, A.T. Kearney, Inc. All rights reserved.

BinaryCore | KEARNEY