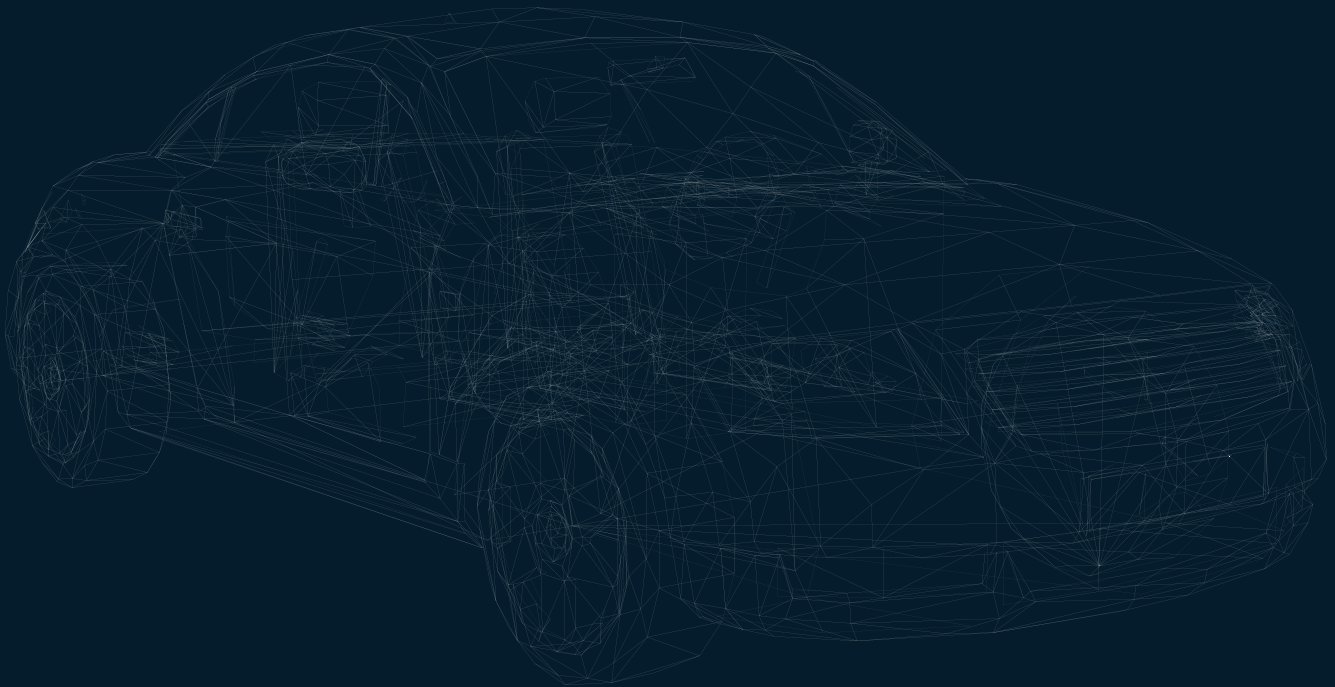


McKinsey Center for Future Mobility

# Mastering automotive software-launch excellence

Automotive players can crack the code on superior launch performance by reducing complexity and increasing robustness in embedded software development.

*by Ondrej Burkacky, Georg Doll, Dominik Hepp, and Rupert Stuetzle*



**Mastering a successful** product launch with respect to time, cost, and quality is a core capability for every organization. However, product-launch delays have multiplied rapidly in recent years in the automotive industry—often with hundreds of millions of euros or dollars at stake for an OEM or tier-one supplier. And that doesn't include the damage to a company's brand and reputation.

The failure to launch smoothly can jeopardize an entire company's competitiveness and brand trust, while disrupting its financial performance. This problem inflicts new industry entrants and established OEMs alike. Based on our research, the central reasons for automotive product-launch delays are the increased complexity of software and electronics and an approach to software development that fails to keep up with growing system-level complexity. The following article outlines best practices and tools to overcome these shortcomings and ensure successful product launches.

### **Growing launch problems highlight the increased importance of software**

The automotive industry faces a multitude of technology-driven disruptions. Software is becoming progressively more important as it increasingly determines the value of a car. The technologies driving this transformation include autonomous vehicles, connectivity, electrification, and shared mobility (ACES), which offer new opportunities for growth and disruption.

In other words, the digital car is finally arriving—over-the-air updates replace auto-shop visits and suggest new business models while software features replace formerly differentiating factors such as engine characteristics or suspension tuning. Therefore, the capability to manage the development of embedded software systems to deliver the right functionality on time and within budget becomes a differentiating asset.

In this environment, launch delays will become increasingly important vis-à-vis the value at stake, which can add up to hundreds of millions of dollars

for an OEM. And that's not counting the damage to a company's brand and reputation from delays. Additional opportunity costs can also arise, such as when a delayed launch leads to additional homologation efforts from increased emission standards.

McKinsey research suggests that the global automotive-related software market will roughly double between 2020 and 2030, outgrowing the automotive market in general (Exhibit 1). This dynamic will further increase the risk of software-driven launch problems.

### **Handling complexity**

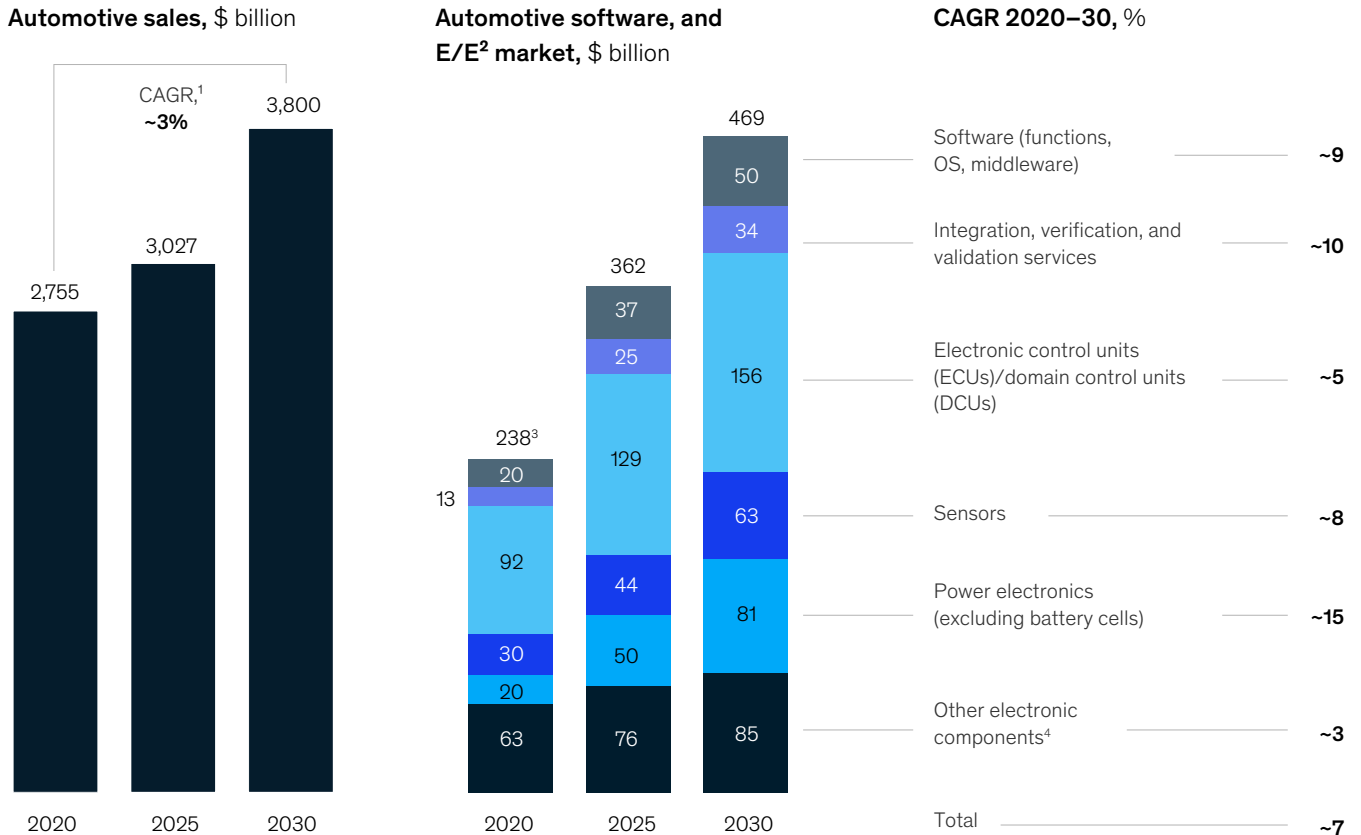
Product complexity has significantly increased during the past ten years, a development that will likely accelerate through 2030. New technological trends like ACES require not only the development of new features but also changes in the underlying electric and electronic architecture. Such shifts call for changes in the industry's collaboration models, with suppliers significantly widening the OEM–supplier interface. In addition, the introduction of new platforms for electric and autonomous cars further complicates the product portfolios of automotive OEMs and tier-one suppliers. Other confounding issues include required updates and enhancements to established internal-combustion-engine (ICE) platforms, including more stringent emission limits in the European Union, such as Euro 7, and the new regulation on cybersecurity and software-updates from the UN Economic Commission for Europe (UNECE).

Furthermore, it remains unclear which platform will be the most successful in the coming years. Will hybrid electric vehicles (HEVs), battery electric vehicles (BEVs), or ICEs win? Amplified via modern regulation requirements such as the Worldwide Harmonised Light Vehicle Test Procedure (WLTP) and Euro 7, the rising complexity levels of products and platform architectures demand rigorous decisions and thoughtful management.

Today, many OEMs must contend with the greatest variety of levels ever in their product portfolios, not only regarding the diversity of models and variants

Exhibit 1

**The automotive electronic and software market will see strong growth through 2030, driven by power electronics, software, ECUs, and DCUs.**



<sup>1</sup>Compound annual growth rate.  
<sup>2</sup>Electrical and electronic components.  
<sup>3</sup>Figures may not sum to 100%, because of rounding.  
<sup>4</sup>For example, harnesses, controls, switches, displays.

Source: Revenue forecasts based on vehicle volumes from IHS Markit (Automotive), Light Vehicle Production Forecast, Oct 2018, pull completed on Nov 6, 2018; McKinsey analysis

but also in the underlying software platforms. OEMs have begun to cut down complexity, but they remain at the start of this journey.

**Handling the workload**

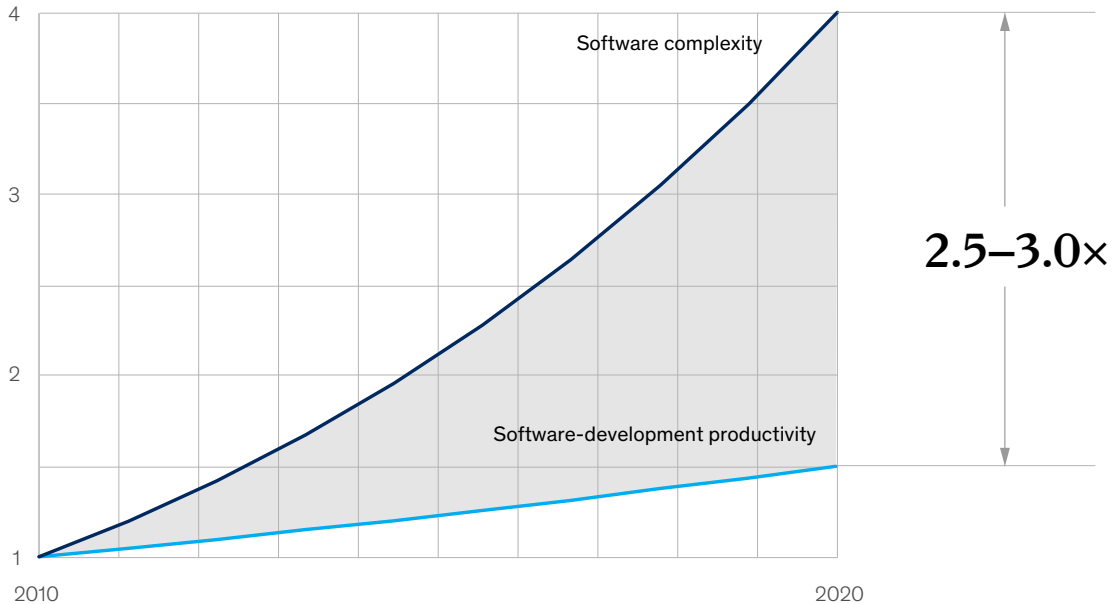
McKinsey research shows that the complexity for mission-critical software features such as those for autonomous-driving functions is currently growing at double to triple the speed of software-development productivity (Exhibit 2).

Traditionally, automakers have managed their interactions with suppliers along well-defined system limits such as physical electronic-control-unit (ECU) boundaries, the functions an ECU provides, and protocol definitions. However, with software arriving in the automotive supply chain, these interfaces are changing and, as a result, their complexity is increasing significantly. Furthermore, new methodologies like agile development call into question traditional development approaches. For example, many OEMs have traditionally specified

Exhibit 2

**Growth in software complexity more than doubles the growth in software-development productivity.**

Relative growth over time, for automotive features, indexed, 1 = 2010



Source: Numetrics

their requirements at the control-unit level and now must do the same for software components. As a result, their attempts to integrate software systems with suppliers often lack depth and rigorous management. What's more, the OEMs' development tools often don't include features needed to monitor supplier progress effectively, such as defect tracking. Likewise, cross-supplier dependencies usually require a holistic understanding at the component-architecture level—insights that current OEM systems do not support and teams often lack.

To set up an effective, shared launch-management approach for embedded-software-system solutions between manufacturers and suppliers, companies need to understand the requirements for launch readiness. Consequently, many OEMs often proactively search for ways to involve first- and second-tier suppliers to share the complexity load.

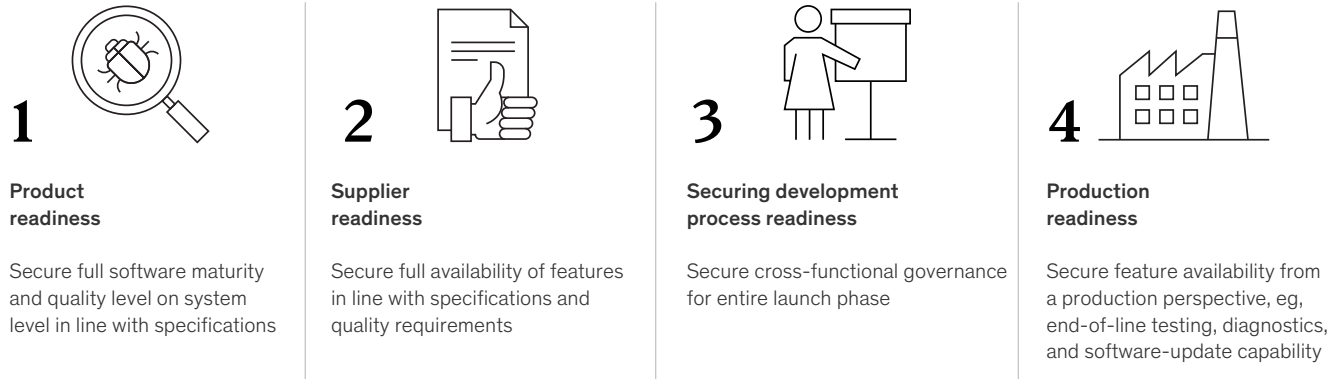
**Mastering software-launch excellence**

OEMs and suppliers must master challenges across four dimensions to achieve software-launch excellence (Exhibit 3):

- *Improved product readiness.* Ensure full maturity and quality when integrating software-enabled functionality from different suppliers and when integrating the full system. This requires the definition of launch-critical features, testing procedures, and product-release effectiveness: debugging, acceptance testing, version control and management, and over-the-air updates. Also, secure integrated product-development processes are needed for both hardware and software.
- *Enhanced supplier readiness.* Focus on timely availability and alignment with quality requirements for software features and

## The launch excellence framework entails four main components.

### Securing software functionality



Source: McKinsey analysis

establish key performance indicators (KPIs) along the entire software-development process, with a concentration on early indicators for launch risks. Focus areas include the ramp-up of capabilities, the guarantee of resources, the conformity of features to specifications, the determination of release processes, and the definition of selected logistics concepts like test execution planning.

- *Robust software-development processes and project management.* Guarantee the robustness of embedded and integrated product-development processes for both hardware and software across OEMs and suppliers. To do this, establish a cross-functional governance system for the entire launch phase jointly with suppliers. To drive the impact across all phases of a launch, it makes sense to introduce a clear governance structure with responsibility matrices, change-control boards, and KPI systems, for example.
- *Focused production readiness.* Ensure feature planning from a production-readiness perspective. The essentials for a successful project launch include production-enabling features such as end-of-line testing, diagnostics,

and software-update capability functions. Another critical element involves exactly aligning milestones along the different production launch steps.

### Solution approaches

Two approaches for avoiding launch delays can be explored. One, the proactive solution, involves front-loading the setup in early project stages, shortly after the start of a development project. The other, the reactive solution, focuses on debottlenecking, feature prioritization, and setup recovery in later project stages, mostly within the last 12–24 months before a targeted start of production. Our research has identified and detailed a set of actions for each approach.

- *Proactive approach* OEMs and suppliers get ahead of the challenge, proactively laying the foundation for software-launch excellence. Building capabilities in managing complexity through full-stack transparency will pay off.
- *Reactive approach* OEMs and suppliers need to turn around an already-delayed running development project that is unlikely to keep

its launch date. Companies must implement mitigation measures under increased time and financial pressure.

### **Proactive approach: Front-loading at early-phase delays**

To ensure product readiness, automakers must sharpen their requirement specifications and integration capabilities. That means developing a policy to optimize and manage requirements, which should include a transparent, cross-functional collaboration in requirement management. They should pursue the early, iterative development of functional and value requirements, and make a neutral and systematic evaluation of the requirements based on cost and value. Furthermore, OEMs need to adhere to a minimum feature-set plan for integration, development, and sourcing. Following a software-industry maxim, they should “build enablers first, then functions on top” to avoid redundancy and gaps among enablers. Other suggestions include enhancing frequent automated testing by introducing and establishing a new software-development tool chain. The tool chain should support automation that could reach from artificial-intelligence-based ticket assignment to automated regression tests on target hardware. They should also start to build a forecast model to monitor task progress based on input from task-driver trees, dependencies between activities and milestones, early indications of tracking, and resource configurations.

Ensuring supplier readiness requires companies to establish a collaboration model between internal and external developers in a continuous-integration mode. Companies need to align supplier interfaces, clearly defining and establishing them at all levels of supplier interaction, detailing milestones, and synchronizing flows along the phases of development. In many situations, development and testing logistics should take the form of reviews, to ensure the timely availability of test vehicles, prototypes, or in-the-loop systems with the right software version installed.

Software-development processes and project-management readiness require automakers to run

criticality assessments along the integration path, including risk assessments. This risk-assessment process should include cross-functional discussions via a dedicated workshop format, as well as a detailed checklist for action items. Developers can use visualization techniques to create action lists for high-risk parts and suppliers and to plot an overview of the risk-assessment process on a risk matrix.

Companies should establish a KPI-based early-warning system, applying a systematic approach early to pick up warnings of upcoming risks. One important aspect of this step involves choosing the right indicators in the KPI set, such as requirement-implementation rates, unit-test coverages, and defect-detection and -resolution rates. When using agile-development methodologies, burn-down velocity offers a good way to determine a project’s actual development status. Organizations should also establish a budget and development plan and use deviations as early indicators. Best-practice companies usually create a cockpit that can automatically track relevant KPIs from the current launch phase, enable abstraction, and ensure adequate coverage from development to the business-unit level.

They also employ predictive analytics to generate greater transparency as far as scheduling the required resources at the start of the project, and to assess plan risks resulting from unrealistic assumptions. Managers need to answer the questions of what risks exist in the current project plan, and what the optimum plan would look like. Output could include forecasts of the personnel needed by role and project phase, for example, or a quantification of the risks that arise due to unrealistic productivity assumptions.

### **Reactive approach: Debottlenecking delays at a later stage**

Product readiness in this case requires automakers to prioritize critical features and descope content along the development timeline. They should consider staged software releases, focusing on diagnostic and factory requirements first and then derisking the production start, and plan early software updates before delivering the first cars

to customers. They must also ensure pragmatic, frequent tests and an efficient planning of the availability of test assets, including test boards, hardware-in-the-loop systems, and test vehicles.

Supplier readiness results from rigorous defect tracking and rigorous quality control for prioritized features. Automakers should also ensure resource availability and that key suppliers have appropriate skill sets.

To promote project-management readiness in the context of embedded-software-development projects, OEMs should set up a digital project-control board, which can help them achieve two objectives. First, it can generate full transparency per cluster for all required software functions. It can also help plan and communicate upcoming release content and the start of production. It provides the testing status for all software functions, including a direct link to error-management systems. Second, the project-control board makes possible analytics-enabled error pre-analysis and management. This allows automakers to monitor the inflow and outflow of error tickets efficiently to ensure a deadline-based burn down. It allows users to monitor ticket transitions between different development teams and helps reduce inefficiencies during root-cause analysis of problems. Of course, companies must augment these KPIs depending on the project situation and data availability.

Automakers and suppliers can also establish a project war room, with a full-time dedicated launch manager and a robust governance structure for quick decision making. The war room thus becomes

the nexus of project governance, featuring daily launch check-in meetings for the escalation of problems and issues and rapid problem solving. All launch teams should meet at least once a week and engage in steering discussions to manage issues that extend beyond day-to-day problem solving. Another use for the war room involves holding progress reviews every two to three weeks where senior executives (above the plant-manager level) participate.

Another priority centers on establishing a standing decision-control board with a clearly defined escalation ladder, ensuring strong progress tracking and KPI-measurement standards. The system should feature standardized change requests that address war-room operations themselves as well as first- and second-level decision-board involvement as needed. For example, first-level decision-board participation might focus on timing, vendor shifts, or content changes, while second-level involvement would include top management and could concern highly risky or far-reaching changes.

---

To establish world-class software-development capabilities, OEMs need to take proactive steps to solve software issues and create a development engine with a holistic software-transformation program. The goal is to fix the basics and develop solid processes, methodologies, and tools that allow engineers and managers to focus on the right priorities to set themselves up for success, independent of any potentially problematic launch project at hand.

**Ondrej Burkacky** is a partner in McKinsey's Munich office, where **Georg Doll** is a senior expert and **Dominik Hepp** is an associate partner; **Rupert Stuetzle** is a partner in the Berlin office.

Designed by Global Editorial Services  
Copyright © 2019 McKinsey & Company. All rights reserved.