BERYLLS STRATEGY ADVISORS

DRIVING SUCCESS: STRATEGIC MANAGEMENT OF AUTOMOTIVE SOFTWARE PROJECTS

AGENDA

BERYLLS STRATEGY ADVISORS

INSIGHT

The software supply chain is becoming increasingly critical to automotive success – yet OEMs and suppliers often struggle to manage it effectively. Now is the time to identify known and emerging problems in software supplier management, and re-think supply processes end-to-end.

1 INTRODUCTION: DRIVING SUCCESS

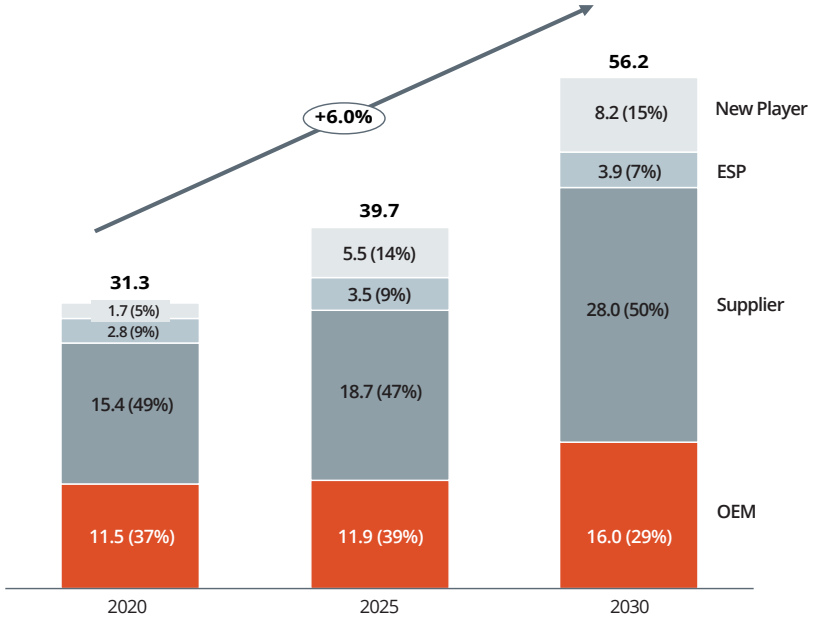
In the ever-evolving landscape of the automotive industry, software-defined vehicles are rising from being a mere futuristic scenario to becoming the dominant technological force. As digital-native Millennials and Gen Z consumers claim a larger stake in the market, car manufacturers are racing to meet their heightened expectations. The significance of the Chinese market amplifies the demand for a seamless digital automotive experience, compelling automakers to swiftly adapt to this technological paradigm shift.

Yet as OEMs and their key suppliers have already found out, the software development and supply chain is a minefield of challenges and potential traps. In just one recent example, a US OEM had to stop production of one of its highest volume models for a period of weeks due to software problems affecting the vehicle's instrument cluster. Such issues do not just impact carmakers' reputations for reliability and customer service - we estimate that for a premium model, the cost of a single week of lost production close to the launch date could range from 34M€ to more than 100M€.

LOSS OF CONTRIBUTION MARGIN PER LOST PRODUCTION WEEK FOR PREMIUM VEHICLES

2 THE FOCUS HAS CHANGED

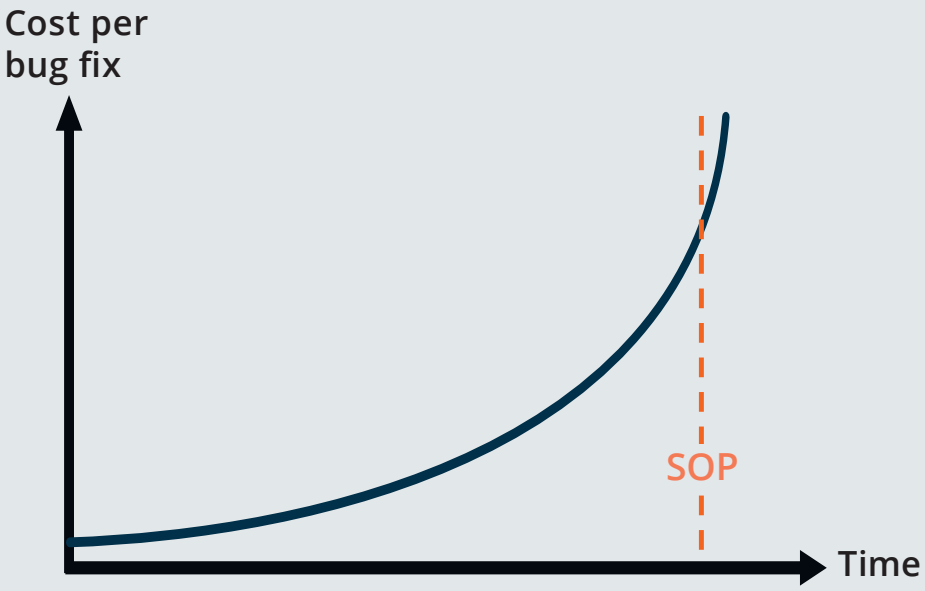
TOTAL SOFTWARE R&D BUDGETS BY COMPANY TYPE, 2020-2030, IN BN EUR, %



Source: Future Markets Insights, Berylls Strategy Advisors

Increasingly, production outages are attributed not to hardware failures, but rather to issues stemming from the vehicle's software. The software-defined vehicle demands a radically different automotive design-and-development model. Manufacturers have to shift from an operating model based on complex hardware layouts (which is where their traditional skills lie) with relatively simple software modules, to simpler hardware designs with complex software architectures. Customer expectations are relentlessly shortening development cycles, which translate into a much greater software development and testing workload.

LATE ERROR DETECTION LEADS TO UNRECOVERABLE COSTS



- Costs to fix bugs increase exponentially over time
- Failing to fix issues early leads to significant costs and quality issues that often cannot completely be resolved until SOP

Source: Berylls Strategy Advisors

Avoiding such costs is entirely feasible. We identified four levers to effectively tackle the complexities of supplier management in the era of software-defined vehicles: harnessing best-practice requirements and systems engineering methodologies, instituting a robust governance framework, integrating processes to enforce standards compliance, and forming a proactive software excellence task force. What unifies these levers is their emphasis on early detection of emerging issues and the imperative to reconfigure the development process accordingly.

3 FOUR LEVERS TO TACKLE THE COMPLEXITIES OF SUPPLIER MANAGEMENT

1. DETERMINE REQUIREMENTS AND SYSTEMS ENGINEERING
2. MANAGE COMPLEXITY THROUGH CLEAR GOVERNANCE
3. BUILD STANDARDS AND COMPLIANCE PROCESSES
4. CREATE A PREVENTATIVE TASK FORCE

1. DETERMINE REQUIREMENTS AND SYSTEMS ENGINEERING

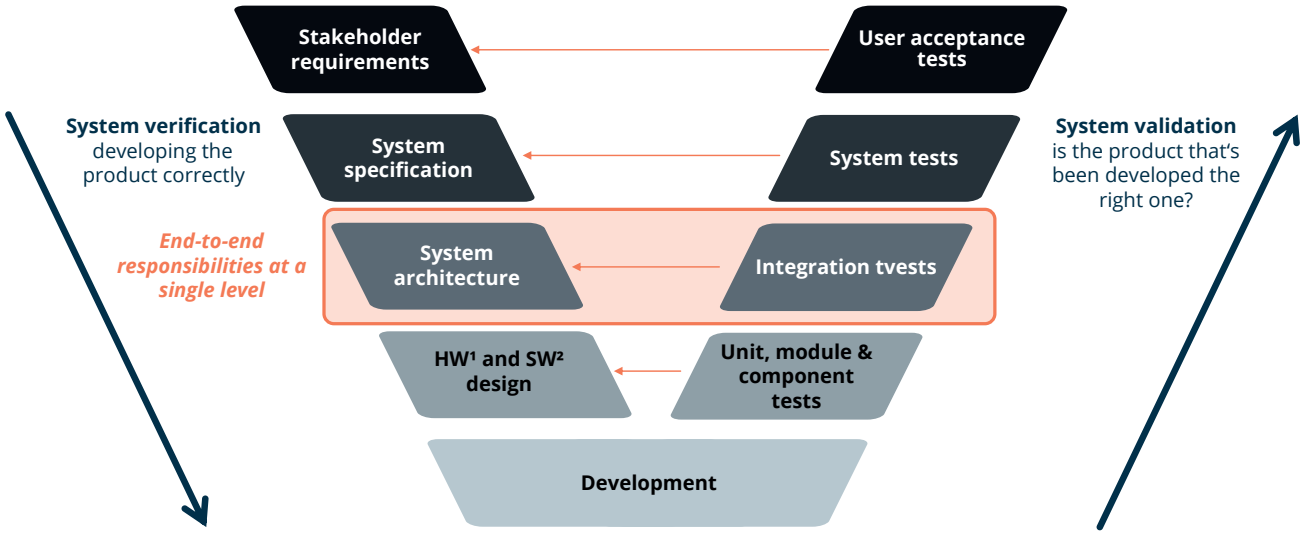
A requirement analysis demands sessions with cross-functional teams to understand which software requirements are truly mission-critical, and where the OEM's maturity level in terms of product specifications and the ability to communicate and align with suppliers on these remains insufficient.

Rigorous attention to specifications and associated documentation is needed (this is where focus is often lost), and there should be an end-to-end line of responsibility so that individuals involved in the initial assessing and signing-off of requirements carry responsibility through to delivery. A comprehensive requirements process is therefore necessary:

- Build **technical and operational expertise** by investing in training and development and if necessary, consult with subject matter experts to augment the capability of teams.
- Develop **common understanding** of requirements through alignment sessions between OEMs, Tier 1 and Tier 2 suppliers, using standardized templates for documentation to minimize interpretation discrepancies.
- **Optimize value creation** by delineating core competencies and identifying modules for outsourcing. Specify interfaces for input/output exchange and evaluation criteria for those outsourced components. Moreover, comprehensive test cases must be formulated and communicated to software suppliers to facilitate seamless integration and robust performance across all software artifacts.
- Implement **full lifecycle management**, ensuring every requirement is actionable, verifiable, and linked to a specific development task or goal – and is documented.
- Adopt **granular planning and ownership**. Break down complex systems using systems engineering best practices such as the Requirement, Functional, Logical, and Physical (RFLP) framework to establish a clear and structured definition of what is needed and how the system and subsystems will fulfil product requirements. Accountability must be established end-to-end at all levels for requirements, implementation and testing.
- **Ensure holistic testing of requirements** not only under ideal conditions (Hardware-in-Loop (HiL)/ Software-in-Loop (SiL) testing) but also under extreme or unexpected conditions (early-on robustness testing) to confirm that all understood requirements are accurately delivered and fulfilled. This can prevent potential failures that might not be evident during routine testing but could manifest in real-world scenarios.

SYSTEMS ENGINEERING – END-TO-END RESPONSIBILITY IN THE V-MODEL

The V-model is applied in systems engineering and ASPICE processes and allows the complexity to be dealt with through a stringent development methodology and end-to-end responsibilities.



Source: Berylls Strategy Advisors [1]Hardware [2]Software

2. MANAGE COMPLEXITY THROUGH CLEAR GOVERNANCE

Governance frameworks are intended to manage quality and security requirements in software. Companies must develop a robust framework for overseeing external software suppliers, ensuring their deliverables align with their respective quality and security standards – and this will mean structural changes in the way OEMs and Tier 1 suppliers manage software-intensive projects.

- Companies should consider a shift to a **system-oriented project structure**. The aim is to reduce interfaces by creating end-to-end responsibilities, and to implement a performance monitoring system for external partners to track progress and adherence to requirements.
- Companies should **rationalize responsibilities** by bundling end-to-end responsibility for each deliverable to a single local team and defining clear handover points. This approach minimizes the number of sites involved, reducing handovers and delay risks, and thereby enhances productivity.

To support these changes, it is advisable to have external functional experts from OEMs or Tier 1s onsite with the software supplier (or vice versa), speaking the same language and sharing the same culture to ensure requirements from the product owner are correctly implemented. This allows OEMs, Tier 1s and possible additional software providers to communicate and align in a transparent way, preferably through a standardized meeting structure with daily check-ins and senior management steering. Experts should be able to discuss sprints and artifacts directly with third parties and the OEM, and collaboratively identify and mitigate emerging errors. Furthermore, it is crucial for Tier 1 suppliers to proactively seek assistance from lower-tier suppliers and OEMs, especially in critical areas such as software testing. This support is essential not only for meeting project deadlines but also for ensuring quality. For example, if there are testing delays during the validation phase prior to the software release and the supplying partner fails to meet the confirmed deadlines, the situation may be managed through collaborative problem-solving sessions rather than escalation. This underscores the importance of collaboration and shared objectives in the automotive software development ecosystem.

3. BUILD STANDARDS AND COMPLIANCE PROCESSES

In the software era, OEMs have to develop new sets of rules and standards that apply to all suppliers in order to ensure adherence to development and production plans and avoid additional and unplanned software iterations.

- Software demands new **development standards**. As OEMs source components with integrated software from different suppliers that must be brought together into one system, the software components must follow the same development and industrialization schedules. To ensure the delivery of stable software within a structured process, all suppliers must follow the same development standards and common Automotive Software Performance Improvement and Capability Determination (ASPICE) scoring levels.

- The software-defined vehicle demands new **safety and security standards**. In the case of ADAS implementations and battery management, or other components connected to the functional safety of the vehicle, companies should establish clear guidelines and checkpoints to ensure that these standards are met consistently throughout the development lifecycle. They should create a playbook outlining expectations and methods and achieve formal agreement from all parties involved to guarantee delivery of customer requirements in line with standards and regulations.

- **Quality standards** are critical. The potential cost of development delays or failures has made quality standards in the software development process critical. Quality standards include sufficient software testing and validation and proof of adherence to the set requirements.

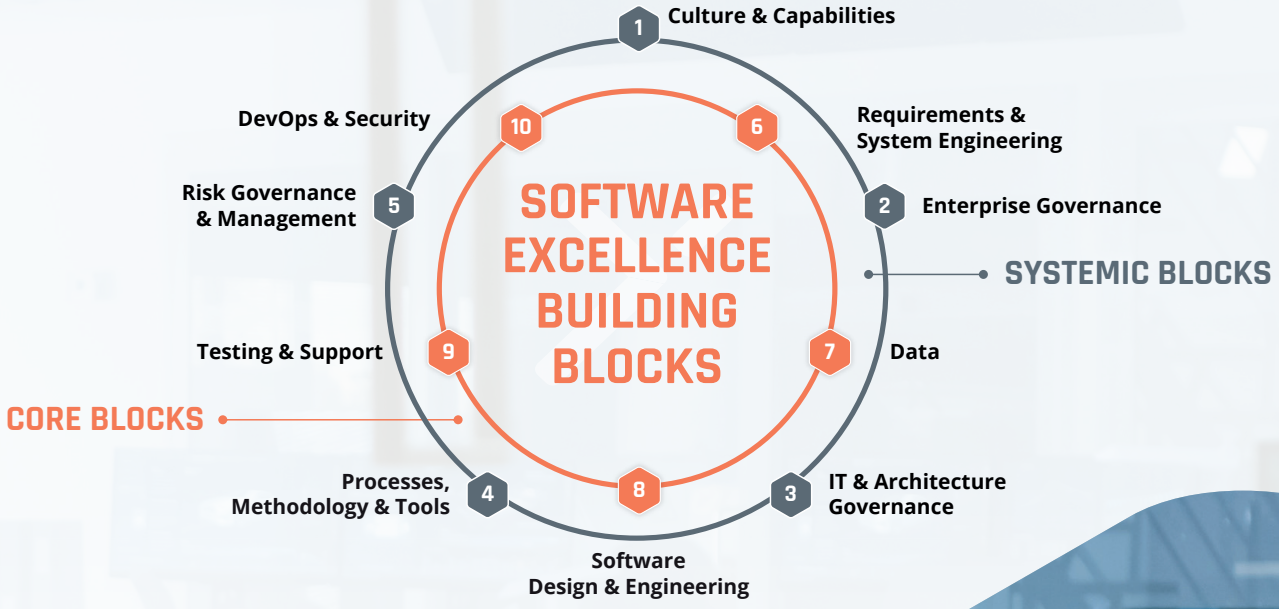
4. CREATE A PREVENTATIVE TASK FORCE

As many OEMs or Tier 1 suppliers have limited experience in software development, and task new and inexperienced teams with developing new products, it is advisable to set up a preventative software task force at the start of new programmes. At Berylls we have developed a comprehensive framework to assess a company's software capabilities and identify the most critical improvement measures to be rolled out in a preventative task force mode. These include:

- **Best-in-class testing**. A test integration model capable of handling anything from a single software module to an entire development chain including specification, code, testing and release. The test process is characterized by a high degree of automation, with virtualized testing environments, SiL, and HiL concepts used for verification.
- **Processes, methods and tools.** The task force concentrates on KPI-based reporting and success measurement, with reporting ideally through live dashboards and focusing on problem-solving rather than problem-restatement in meetings.
- **Risk governance and management.** Identification, assessment and tracking of risks (such as time, quality, quantity, cost and functionality) with clear assignment of risk management responsibilities.
- **Software design and engineering.** The task force enforces software best practices to increase efficiency and quality. It establishes a regular and well-managed release cycle as the key for high quality software deliveries to customers.

BERYLLS SOFTWARE EXCELLENCE BUILDING BLOCKS – AREAS FOR ACTION

We established a comprehensive framework for evaluating the state of a software project or organization and derive work packages.



Source: Berylls Strategy Advisors

4

PUTTING A SOFTWARE PROCESS BACK ON TRACK

A Tier 1 automotive supplier encountered challenges in managing a critical Level 3+ autonomous driving software development project. The engineering team had overspent its budget by more than four times and start-of-production was at risk due to a high amount of defects. Moreover, there were significant delays in the implementation of both safety and non-safety features. The root causes were an excessive number of contributing suppliers, a lack of coordination, and a failure to prioritize resources effectively. In response, an action plan was crafted to stabilize the start-of-production schedule by focusing solely on must-have requirements.

To address these challenges, a Change Control Board was formed between the OEM and Tier 1, tasked with narrowing down ‚must-have' requirements to only essential features. These were then segmented into manageable deliverables. This segmentation facilitated more focused and efficient development efforts, as the team could tackle each deliverable with clear objectives and deadlines.

Furthermore, an onsite task force was established, implementing a structured meeting landscape, and conducting efficient problem-solving workshops to determine necessary actions and appropriate measures to guarantee timely software delivery. The progress and challenges were monitored in twice-daily ‚stand-up' meetings to ensure accountability and enable swift responses to any emerging issues. This structured approach significantly enhanced our capacity for prompt issue resolution.

Additionally, an expanded testing program was launched, significantly intensifying in-vehicle testing and overall testing efforts. This proactive approach not only improved the robustness of each software update but also enhanced the reliability of the software. It streamlined the development cycle by enabling earlier detection and correction of defects, which reduced the frequency and extent of revisions needed to meet customer requirements. This strategy effectively reclaimed three months of development time, allowing the Tier 1 to deliver a high-quality product to the OEM on schedule.

5 CONCLUSION

In conclusion, the shift to software-defined vehicles presents both challenges and opportunities. To prevent budget overruns and ensure timely delivery of software projects, our experience is that early intervention and strategic project management are the keys to success.

Clear governance, the importance of standards and compliance from day 1, and the benefits of a preventative task force are all critical aspects to consider. The cost of not addressing these issues can be significant, as seen in the potential loss of millions due to production delays. However, with strategic management and a proactive approach, these costs can be avoided. Now is the time to act, to not let the challenges hinder the progress and instead achieve greater innovation and success in the new software-defined era of mobility.

Reach out to Berylls today, to explore how our bespoke solutions can cater to your project requirements and lay the groundwork for your succes.

MEET BERYLLS

Berylls Strategy Advisors – The expertise of our top management consultants extends across the complete value chain of automobility – from long-term strategic planning to operational performance improvements. Based on our automobility thought leadership Berylls Strategy Advisors stand out with their broad experience, their profound industry knowledge, their innovative problem-solving competence and, last but not least, their entrepreneurial thinking.

YOUR CONTACT PERSONS

Christian Grimmelt
Partner
christian.grimmelt@berylls.com

Sebastian Böswald
Associate Partner
sebastian.boeswald@berylls.com

Fabian Dinescu
Senior Consultant
fabian.dinescu@berylls.com

Julius Gaupp
Consultant
julius.gaupp@berylls.com

STRATEGY
ADVISORS

Maximilianstraße 34 | 80539 München | +49 89 710 410 40-0
info@berylls.com | www.berylls.com

Munich, April 2024