Development & Technology

Three pitfalls in software-driven transformation

How to tackle the growing challenges

INSIGHTS

//01

New automotive players and mounting digitalization are leading to an increase in the product complexity and pace of change in automotive products, forcing the traditional automotive industry to follow suit.

//02

Software-based product solutions have to be approached with a holistic mindset: at the strategic level of software capabilities, at the tactical level of product delivery, and at the operational level of software development.

//03

Strategic collaborations with software and technology pioneers are crucial to swiftly build up necessary capabilities. The key to utilizing these capabilities is a decoupling of hardware and software in the product as well as in the product development process.

Understanding software-driven transformation

Software transformation is the answer to rising product complexity

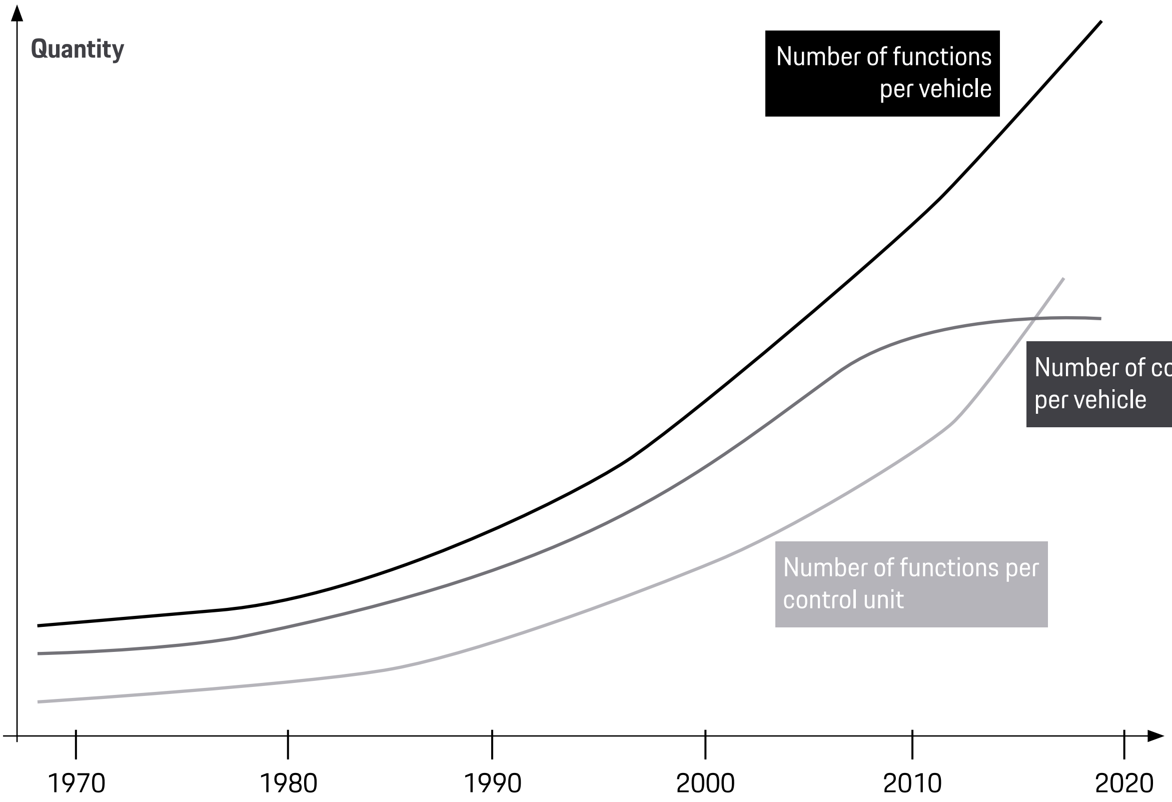
Digital transformation in the automotive sector is accelerating and is increasingly tying up the capacities of traditional vehicle manufacturers. This acceleration is generated by the reciprocal reinforcement of market-push and market-pull factors, among other influences. Together, these factors result in a disproportionate rise in product complexity, especially in the digital sector.[1,2]

Market-push and market-pull factors reinforce digitalization in the automotive sector

The main market-push factors that are currently tying up the capacities of traditional vehicle manufacturers are the electrification of the powertrain, the digitalization of the vehicle, and automated driving. These themes are the focus of intense activity, and are being driven not least by new players on the market such as Tesla, Google, and Apple. At the same time, the automotive industry continues to be challenged by the digital transformation of society as a market-pull factor. The digitization of everyday life, for example through the use of digital services and platforms, is gaining ground in all areas of life and thus also heightening customers' demands on their mobility, the associated vehicles, and the infrastructure.[3,4]

Increased product complexity and pace of change

The number of functions per vehicle has been increasing steadily since the 1970s. The increase in functions has been accompanied by an increase in the number of control units to perform the functions. Since the mid-2000s, the number of functions per control unit has been increasing at a faster rate than the number of functions, resulting in a trend toward consolidation or even a decline in the number of control units.



© Porsche Consulting

Fig. 1. Increasing number of functions and control units per vehicle[5]

The new functions are generally highly interconnected and must be integrated in the vehicle and in the surrounding systems. This results in higher complexity of the product, vehicle, and peripheral systems. The complexity of the products on the requirements side is primarily answered by new software functions, which also increases the complexity of the network and the software architecture.[6]

In addition to increasing product complexity, the digitalization-related pace of change in the products is also on the rise. Flexible IT solutions and software services, for example in the consumer goods industry, are changing at a significantly higher rate than the automotive industry's mostly hardware-based solutions, with their multi-year development cycles.[8,9]

> "[The] increase [in product complexity] is not only due to the fact that there are more and more software functions in vehicles, but also to the simultaneously increasing concentration of functions on fewer and fewer electronic control units."[7]

The consumer goods industry and the digital pioneers in the automotive industry, such as Tesla and NIO, are also fueling increased expectations among automotive customers with regard to the car as a product. The vehicle software and its functions should always be up to date and errors must be rectified without delay.[10]

Digitalization challenges the conventional automotive industry

The transformation that digitalization is bringing about is presenting traditional vehicle manufacturers with new challenges. Among other indicators, this becomes visible in the statistics of recent years:

- Forecasts predict that software development costs will increase by more than 40 percent between 2022 and 2030.[11]
- Approximately 60 percent of vehicle recalls can be attributed to the increased complexity of the products, carryover parts strategies and the speed of development.[12]
- Between 2011 and 2019, the number of recalls in the German automotive sector more than tripled to record levels.[13]

The need for conventional vehicle manufacturers to digitalize their organizations and products is thus clearly evident. Topics such as the aforementioned increase in complexity, the increased pace of change in the product, and the simultaneous need to address challenges in the development and provision of hardware and software require a high level of attention in order to remain competitive. Without the organizational, procedural, and operational focus on the provision and mastery of software-driven product solutions, traditional vehicle manufacturers will be left behind in the digital realm by the likes of new automotive players such as Tesla and NIO.

Change on all levels of the organization

To implement the ongoing software-driven transformation, the necessary changes must be embedded in an overall strategy. The levers for these changes are forcing traditional vehicle manufacturers to take action on three different organizational levels:

- At the strategic level of the organization, the capabilities that a vehicle manufacturer must have as part of the software-driven transformation need to be defined. Both the objectives and the necessary capabilities are derived from the digitalization strategy for the organization and the different approaches can be taken to make this capability available and usable for the organization: acquisitions of companies, entering into collaborations with partners, and open-source solutions are oft-cited approaches here. The chapter "Debugging the organization" provides information on how traditional vehicle manufacturers can identify and make available the necessary capabilities to maintain their competitiveness.
- At the tactical level, the challenge is to address the increased pace of delivery. Resolutely decoupling processes along the different delivery paces enables increased flexibility in product deployment. A precondition for harnessing decoupled processes taking place at different speeds is a decoupled product architecture geared to the delivery pace. The resulting interfaces must be managed through a

dedicated interface management. The chapter "Unleash the product change" addresses the necessary changes in the product and process and presents recommendations for traditional vehicle manufacturers.

- At the operational level, software development processes form the procedural backbone of a software-driven organization. In addition to creating efficient software pipelines, they also lay the groundwork for end-to-end complexity management and a flexible and decoupled product architecture. The strict derivation of the product architecture on the basis of strategic corporate concepts as well as market and customer requirements are the key factors for the implementation of a software-driven transformation at the higher organizational levels. The chapter "Hacking complexity in software product creation" provides valuable information on the focus points to make complexity manageable in automotive software development as well.

Debugging the organization

Collaborations offer an opportunity to quickly expand the capability portfolio

Market-related push and pull factors are forcing traditional automotive companies to acquire software-related capabilities in order to meet the trends of electrification, digitalization, and increased autonomy of vehicles and to remain competitive. The shift to "software centricity" in the automotive environment expands the target capabilities of the organization. Software can only be developed successfully and integrated into existing products and processes if its complexity and connectedness in the overall product are understood and mastered. However, barriers such as resource bottlenecks and rising costs are making it increasingly difficult to build such capabilities. More and more traditional companies in the automotive industry are therefore turning to internal and external collaborations with partners in order to share complementary resources in terms of software know-how and capabilities. Three success factors are relevant here: awareness of the company's capabilities, a willingness to proactively shape and manage the transformation, and the decoupling of the hardware and software process landscape.

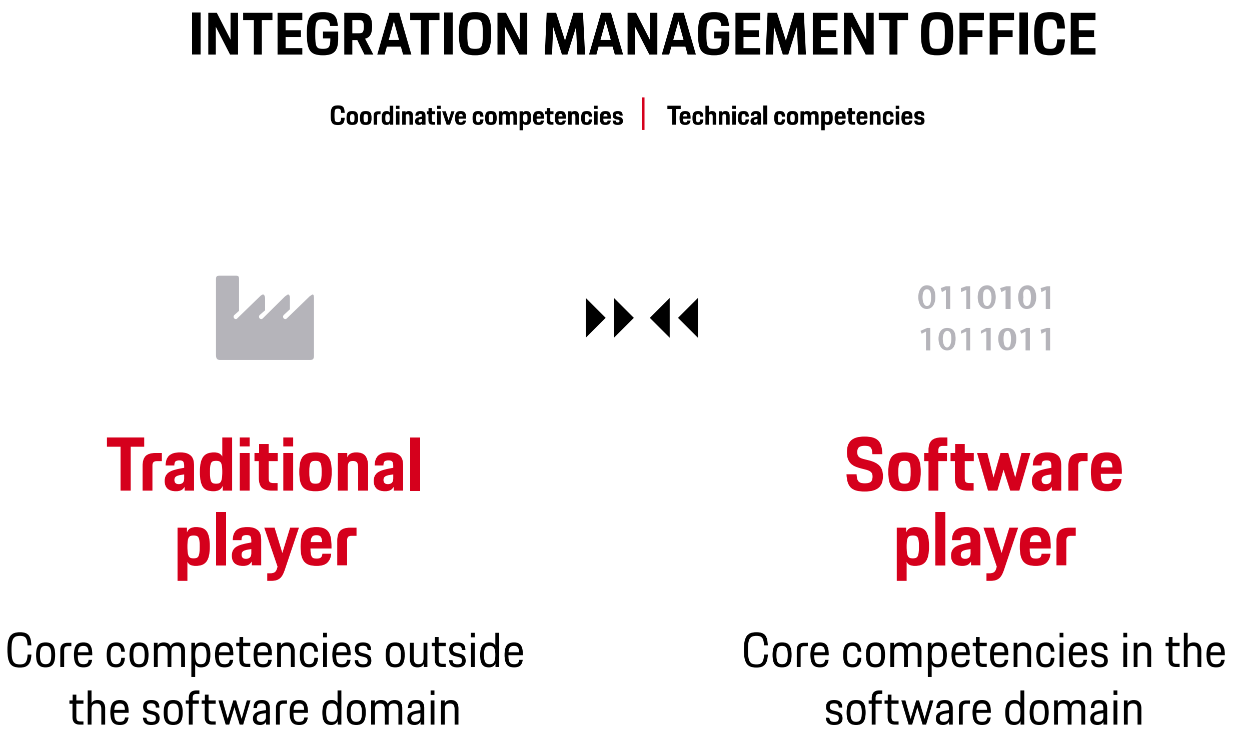
> **A competitive advantage depends not on the resources that are controlled, but on those that can be accessed.**

Capabilities: First reflect, then act

Before entering into a collaboration model with other companies, a dedicated analysis to identify core competencies should be carried out. For example, the strengths of traditional automobile manufacturers are often found in the conventional areas of body, electrics, chassis, and powertrain as well as organizational, coordinative areas, which have been elaborated through the modular assembly of vehicles and the management of that process. Once the current core competencies are known, they are compared with the corporate objective, the corporate vision and the desired, strategic positioning. Comparing the targets with the actual situation makes it possible to derive the skills and capabilities needed to achieve the targets that are not yet in place. Looking at traditional vehicle manufacturers, software-related capabilities are often insufficiently developed. However, not all capabilities can be built up independently and in a timely manner due to factors such as investment risks and market barriers. To overcome these obstacles, collaborations offer traditional companies the opportunity both to acquire capabilities and to gain deeper insights into the market and technology from progressive technology companies. This makes it possible to tap into market potential more quickly and react to changes in technology. Compared to building up these capabilities independently within the company, this makes it possible to overcome the barriers to acquiring operational and strategic software know-how and dealing with resource scarcity more flexibly. Horizontal collaboration between traditional and technology companies is therefore better suited to flexible capability development in the automotive environment than vertical integration of skills via the supply of services or the acquisition of parts of companies.

Organization: The best of both worlds

Collaboration models, both internally within an organization and externally when two independent organizations work together, pose specific transformational challenges. These challenges arise from the prevailing circumstances of the individual organizations and can be of a formal nature (e.g., incompatible process landscapes) or an informal nature (e.g., incompatible corporate cultures). Incompatibilities can occur, for example, where traditional companies or business divisions collaborate with progressive, agile companies or business divisions, as it can currently be seen in many cases in the automotive industry. To ensure that the potential of collaboration can be exploited through mutual sharing of core competencies and is not disrupted by formal and informal barriers, organizational or project-based collaboration should be coordinated centrally. This function can be implemented by installing an integration management office (IMO). The IMO acts as a control instrument that connects all activities related to the transformation. The activities of the IMO are focused primarily on creating overarching framework conditions, e.g., for the development of a common process landscape. In the course of elaborating processes, factors such as the definition of communication structures, conflict resolution procedures, interface functions, and the exchange of information are considered. For informal structures, the IMO should have expertise in the field of change management. Identifying change drivers and activating management to support and represent them are key factors in successfully bringing together two collaboration partners and facilitate the implementation of formal compatibilities. Located directly with management, the IMO thus fulfills an overarching governance function. The task of the IMO is to bring together and make accessible the core competencies of both collaboration partners in the best possible way in the context of both organizational worlds.



INTEGRATION MANAGEMENT OFFICE

Coordinative competencies | Technical competencies

Traditional player

Core competencies outside the software domain

Software player

Core competencies in the software domain

© Porsche Consulting

Fig. 2. The integration management office is the central control function for the optimal utilization of complementary capabilities

Processes: Starting over again

The combination of core competencies in the automotive and software industries should lead to a software-oriented, horizontal expansion of the range of capabilities for traditional vehicle manufacturers. To achieve this, the process worlds of two collaborating companies must be brought together. The content of both process landscapes should be aligned for the respective business units, while simultaneously decoupling hardware and software-related processes. Aligning the content of the process landscape requires the coordination of tasks, capabilities, and areas of responsibility for processes, methods, and tools within both collaborating companies. The decoupling of hardware and software processes promotes flexibility in product deployment. To address the factor of product complexity in the process design, e.g. in network design in the overall vehicle system and the corresponding requirements for hardware and software, the aim should be to decouple their processes. In this context, support processes such as project management and configuration management must be a particular focus for the joint process design. They exert influence on the development processes and play a guiding role. Only a holistic view of the process landscape can enable the interaction of software and hardware development.

In summary a transformation of a of traditional industrial company in the automotive industry into a software-driven company requires an analysis of the company's current strengths and weaknesses. This should be used as a basis for complementing and expanding the capability portfolio. Collaborations offer a suitable form for the purpose. Central coordination of the transformation process should be provided in the form of an integration management office. This ensures that the processes are not only coordinated with each other, but also integrated into a common process landscape in focus areas. To accommodate the different development cycles for hardware and software, the associated development processes should be abstracted from each other.

In addition to the strategic level, it is also important to address the challenges of the increased delivery speed at the tactical level. The following chapter presents recommendations for changes in product and process.

Unleashing the product change

Decoupling increases flexibility in product deployment

Software capabilities alone do not automatically yield the desired benefits in the organic existing structures of automotive companies. Adequate product and process architectures are necessary to achieve flexibility and speed. The demand for a high degree of flexibility and speed in product adaptation combined with the increasing complexity of software-based products poses a particular challenge to traditional vehicle manufacturers. Product complexity is often addressed with waterfall-type development sequences. Strictly hierarchical product and process structures with coupled hardware and software processes are generally an integral part of the company. At the same time, however, this coupling leads to low responsiveness in terms of product deployment, compromising the ability to react to rapid changes in customer requirements during product development and after the launch.[14]

> **"Despite the increasing differentiation, individualization and personalization of the products, [...] the demand for shorter delivery times [...] and higher flexibility with regard to product changes [...] will continue to rise."[15]**

Coupling inhibits change flexibility

Increasing the agility of product development processes (e.g., via agile frameworks such as SAFe, or LeSS) in automotive software development represents a valuable approach to increasing responsiveness. The full potential of this approach cannot be realized, however, if the product and process structures have internal coupling that inhibits flexible product development and deployment of the products.[16,17]

Coupling in the product can occur, for example, if complex functions are distributed over multiple software components and are thus highly mutually dependent. In this case, the adaptation of one function can result in changes to a high number of software components at the same time and thus necessitates major integration efforts.

Coupling in the process, like coupling in the product, includes, for example, the compulsory coupling of software releases together with hardware releases in a shared cycle. The separation of software and hardware releases is often inhibited due to waterfall-type processes, as there is no separation of the processes, and thus no independent development and integration, of the individual product components.[18]

Without decoupling within the product and process, it is not a single product component that determines the development time, but rather the entirety of the hardware and software components affected by the changes. The component with the longest development time in the overall product thus also determines the deployment speed for new product functions and product changes. This makes short-cycle deployment of rapidly variable product components and functions impossible.

The three steps of decoupling for flexible deployment of functions

Coupling in the product or process components of traditional vehicle manufacturers often developed organically over time and can be resolved through restructuring of the product architecture and the development processes over the entire product life cycle of the vehicles. This approach enables vehicle manufacturers to make full use of the advantages of rapidly adaptable software solutions:

01
Identify the blockers: analysis of coupling in the product architecture

Dependencies in the product can be visualized and categorized by functional analysis methods such as sequence analyses, activity diagrams, or state machines. Through this process, instances of coupling are identified and can then be dealt with.

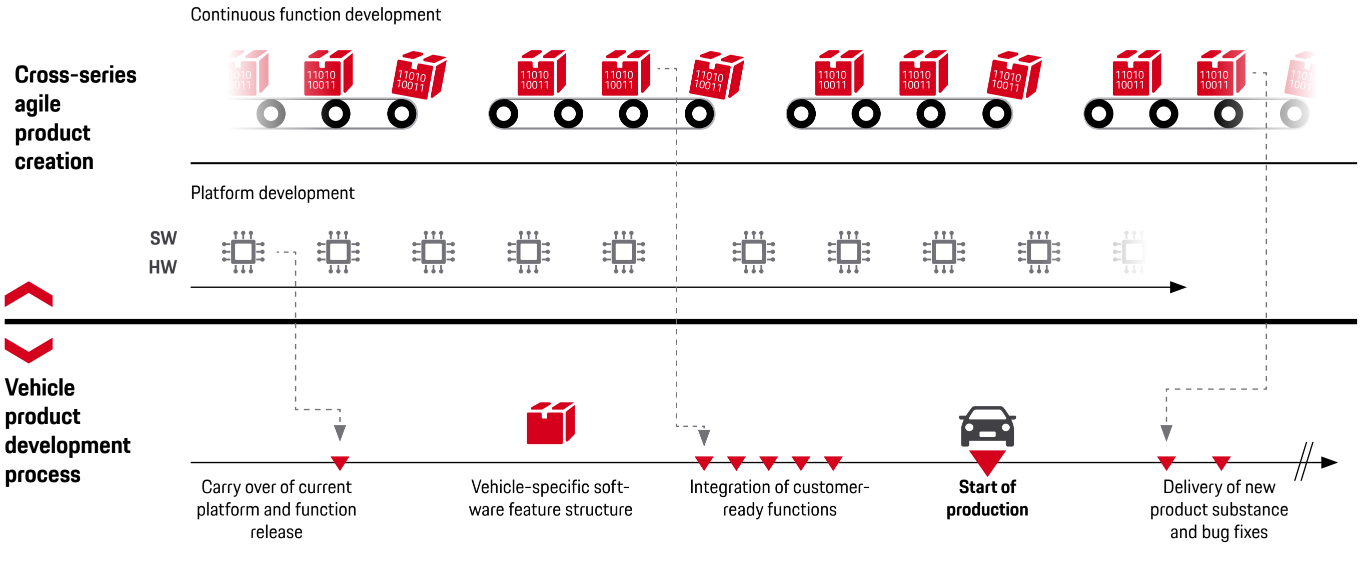
02
Form follows function: product structure and interfaces based on the product strategy

Coupling in the product can be eliminated through the modularization of product components. The modules should be designed as functionally independent of one another as possible and with strictly managed interfaces in order to enable the independent change, replacement, or omission of individual modules. Design decisions should incorporate the required pace of change for the modules, i.e., a “pace-layered architecture,” in order to achieve the highest possible degree of change flexibility in relation to market and customer requirements.

03
The dual operating model: pace-dependent development and application processes

Starting from the independent modules of the product, in a further step it is possible to identify the linkages in the process that impede the independent development and deployment of the architecturally independent modules. For each module, the development and application process is analyzed and the interdependencies identified. In order to resolve these dependencies and to implement a pace-dependent process architecture, the processes can be separated and then, for example, reunited in a targeted manner via fixed synchronization points. The synchronization points re-link the now separated cycles of slower and faster process components only at planned points in time. At the same time, a decoupling of the processes is achieved outside the synchronization points, increasing the deployment flexibility of the faster process components. The independence of slower and faster development and application processes is at the core of the sought-after dual-operating model.

Cross-series portfolio


Continuous function development
Cross-series agile product creation
11010 10011
Platform development
SW
HW
Vehicle product development process
Carry over of current platform and function release
Vehicle-specific software feature structure
Integration of customer-ready functions
Start of production
Delivery of new product substance and bug fixes

© Porsche Consulting

Fig. 3. The dual-operating model combines short- and long-cycle development and deployment processes

In summary, the elimination of coupling in the product and process makes it possible to deploy product components that are not dependent on the release cycle of slower product components, but rather on the pace of delivery of individual product components. Short- and long-cycle product components can thus be independently adapted, integrated, and released. The resulting dual-operating model increases change flexibility and responsiveness to market and customer needs. The basis for achieving change flexibility is a strategic modularization of the product structure as well as strict interface management. The adaptation and mastery of the complex product structure and its interfaces require clear determination of the requirements from the overall product to the software level as well as comprehensive management of the interfaces. Approaches to this are discussed in more detail in the chapter "Hacking complexity in software product creation."

Hacking complexity in software product creation

Focus on software product creation

As seen in the previous chapter "Unleash the product change," decoupling slow and fast paced product components enables a flexible and rapid reaction to change by making use of a dual-operating model. The software product creation process contributes to the flexibility of reacting to customer and market needs as well and must also be optimized to deal with the rapidly growing complexity of software in the automotive industry. Ambitious customer expectations and flexibly deployable software, for example, can drive the software complexity higher. An advanced process design can help to structure and control the complexity in an organization becoming more software-driven.

Software product creation—the complexity challenge

In an ever more digital world, the improvement of software capabilities in consumer electronics leads automotive customers to have more ambitious demands on automotive software as discussed in the introductory chapter. This, combined with the increasing popularity of mobility solutions over ownership models, places further expectations on the software feature and function capabilities. This increase in capability drives an increase in the number and complexity of requirements, which the development must satisfy. A linear increase in requirements logically forces the specification to contain exponentially more interdependencies between requirements.[19]

In addition to the increasing complexity in requirements specification, in practice, the journey from customer need to an accepted technical requirement is significantly longer than desired. Requirements and software expectations also change during the development cycle due to, for example, a disruptive new technology. The implementation of requirements and software changes especially during the development cycle can cause disorder in the rollout of features and functions, due to complex dependencies in the software. The rapid deployment of software and software changes in combination with a more complex software product poses a conflict that must be mitigated to successfully develop, test, and deliver the software product customers want.

Furthermore, in the pursuit of faster, more agile, and more flexible code development and subsequent deployment, there is an expectation that all software variants must be easily and continuously deployed across a variety of environments. These environments include not only the series hardware (ECU) platforms but also virtual testing simulation platforms for speeding up development and integration. This multitude of hardware and software platforms forces software to be platform-independent and modular. This type of implementation presents a challenge, especially when incorporating legacy code with strong hardware dependencies. The incorporation of legacy code cannot always be avoided if the code performs a central and time-sensitive function, or if it cannot be replaced without significant effort.

Requirements mastery

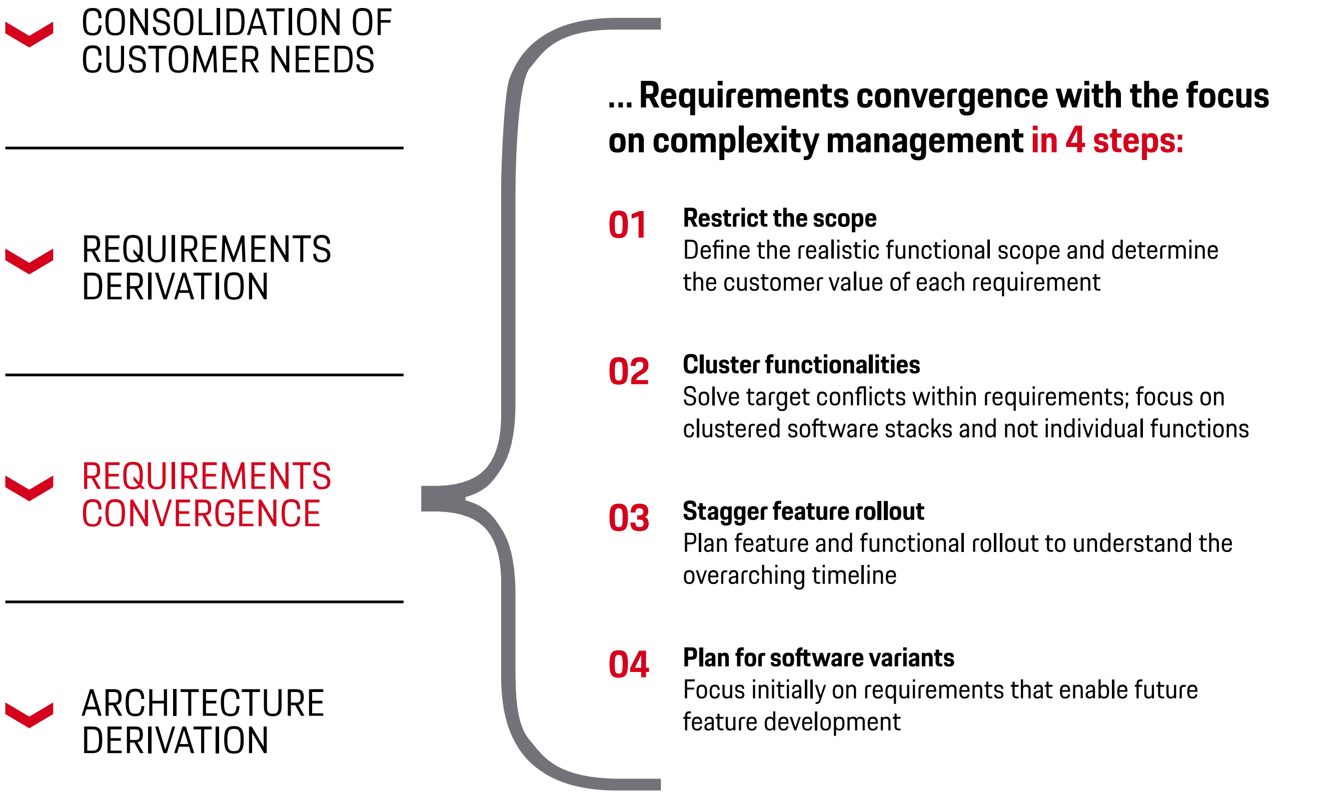
The successful definition of the future vehicle software must combine customer centricity and a requirements convergence process. This implies focusing technical requirements on customer needs and converging them with developers and suppliers to an achievable specification. This collaborative requirements convergence process structures the journey to an aligned software specification that can be more easily accepted into the further development. The convergence of requirements to manage the complexity can be broken into four steps:

- **Restrict the scope:** Define technical requirements in collaboration with both customer need experts and software development specialists to distill requirements to the core of the customer need while remaining obtainable.
- **Cluster functionalities:** Solve target conflicts in the requirements specification by clustering functionalities and benefit from synergies in software stacks rather than function silos.
- **Stagger the feature rollout:** Work with software development to plan the feature rollout early and develop an overarching timeline for software product deployment to projects.
- **Plan for software variants:** Focus initially on software development critical to building the base of the software architecture (cross variant) to allow for future variant-dependent software development on a standardized interface base.

The combination of customer centricity and requirements convergence manages the complexity by primarily focusing development on customer needs and identifying and aligning any compromises in requirements early in the development.

SOLUTIONS TO ADDRESS THE COMPLEXITY

Reactive and dynamic software development is necessary to respond to the challenges in the organization creating software products.



CONSOLIDATION OF CUSTOMER NEEDS

REQUIREMENTS DERIVATION

REQUIREMENTS CONVERGENCE

ARCHITECTURE DERIVATION

... Requirements convergence with the focus on complexity management in 4 steps:

01 **Restrict the scope**
Define the realistic functional scope and determine the customer value of each requirement

02 **Cluster functionalities**
Solve target conflicts within requirements; focus on clustered software stacks and not individual functions

03 **Stagger feature rollout**
Plan feature and functional rollout to understand the overarching timeline

04 **Plan for software variants**
Focus initially on requirements that enable future feature development

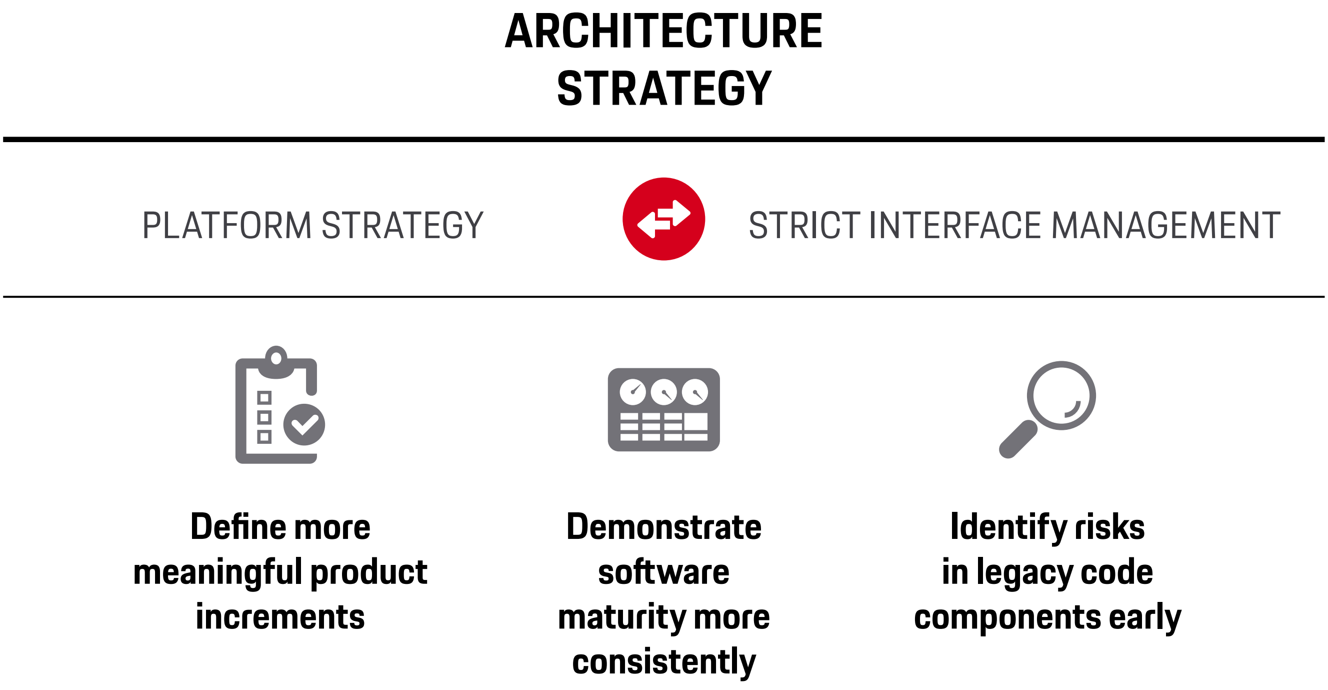
© Porsche Consulting

Fig. 4. Steps to mitigate complexity on the left half of the development V-model

Control the environment

The modelling of architecture structures must incorporate a platform strategy and strict interface definition and management. The platform strategy should clearly define all environments, virtual and real, in which the software products are to be deployed. The architecture strategy shall incorporate parts of integration management early in the software product creation process, also defining when software can be deployed in each environment. This allows agile teams to plan meaningful product increments with demonstrable outcomes per increment. The platform strategy shall give transparency when software environments are expected to be available to developers and coordinate the platform development.

Deployment to a variety of environments is also made significantly easier by defining clear interfaces in the software architecture, as well as ensuring that these defined interfaces are respected. As mentioned, legacy code with hardware dependencies can pose a significant risk to the modularity of code. The interface definition and management thereof work in combination with the platform strategy to define any risks due to legacy or hardware-dependent software early in the development. A platform strategy and interface management incorporated into the architecture process can significantly decrease integration complexity later in the development process.



ARCHITECTURE STRATEGY

PLATFORM STRATEGY

STRICT INTERFACE MANAGEMENT

Define more meaningful product increments

Demonstrate software maturity more consistently

Identify risks in legacy code components early

© Porsche Consulting

Fig. 5. Extract of architecture strategy and main benefits

In summary, due to the increasing complexity in software product creation, complexity management techniques are required in the software product creation process. These techniques have the best possibility of making a noticeable reduction in complexity if set as early as possible in the development. It is for this reason that the suggestions mentioned in this chapter aim at addressing the requirements and architecture phases of the development cycle. Comprehensive requirements convergence built into the requirements management process, as well as a clear platform strategy and strict interface management built into the architecture process can help to mitigate complexity before it becomes unmanageable.

Ready to transform

New players in the automotive market such as Tesla, Google, and Apple are driving digital market-push effects while the ever-expanding digitalization of everyday life is creating a market-pull effect. This is heightening the pressure on traditional vehicle manufacturers to keep pace with the new competition in the field of digitalization. Rising customer requirements lead to rising product complexity, while at the same time accelerating the speed of innovation and change in products on the automotive market. The requisite organizational and procedural changes must be analyzed and implemented on three levels:

Vehicle manufacturers must first carry out a comparison between the overall digitalization strategy and the necessary skills of the organization. Any gaps in the capability portfolio can be filled as needed through the development or restructuring of in-house resources, although the current shortage of skilled workers make this more difficult. A quick and flexible alternative to building up new capabilities is collaboration with competent partners with software expertise. The installation of an integration management office, which controls the collaboration and the process unification, serves to integrate the newly acquired competences.

In the next step, the available software capabilities must be used at the tactical level for decoupling hardware and software in the product and process in order to harness the advantages of a flexible software product and to accelerate the rate of delivery for the product. The key to this is analyzing the coupling both in the product architecture and in the product development process. This coupling can be specifically eliminated in individual product components that are intended to be rapidly adaptable.

Managing the individual product components in the development and application process presents a special challenge in the context of increased product complexity and pace of delivery. One of the keys to successful transformation for vehicle manufacturers is therefore consistent attention to customer requirements throughout the product architecture and the software based on a platform strategy, a controlled convergence process for the requirements as well as dedicated interface management between the individual product components.

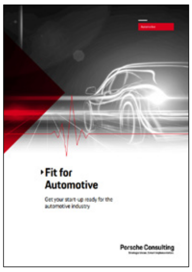
In Brief

01 Horizontal collaborations enable flexible acquisition of software-related capabilities. An integration management office can thereby manage barriers that arise and coordinate process integration for both collaboration partners in addition to change management-relevant aspects.

02 Significantly increasing the deployment flexibility and pace of change for a product requires a dedicated decoupling of rapidly changeable product components. The process architecture, for example in development and application processes, follows the product architecture in order to make the flexibility operationally usable.

03 Complexity in software product creation is growing due to a multitude of factors. A focus on converging requirements to form realistic and achievable specifications as well as a platform strategy and interface management can restrict the complexity early.

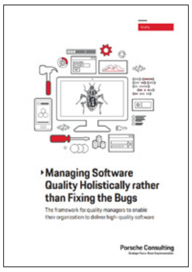
Further reading


Fit for Automotive

Get your start-up ready for the automotive industry

Porsche Consulting

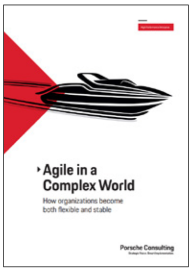
Fit for
Automotive


Managing Software Quality Holistically rather than Fixing the Bugs

The framework for quality managers to enable their organization to deliver high-quality software

Porsche Consulting

Managing
Software Quality


Agile in a Complex World

How organizations become both flexible and stable

Porsche Consulting

Agile in a
Complex World

Authors

Dr. Rodrigo
Biurrun
Partner

Matthias
Brandt
Manager

Ines
Burkhardtsmayer
Senior Consultant

Stuart
Church
Senior Consultant

Contact

Dr. Rodrigo Biurrun
☎ +49 170 911-5454
✉ rodrigo.biurrun@porsche-consulting.com

Porsche Consulting

Porsche Consulting GmbH is a leading German strategy and operations consultancy and employs 800 people worldwide. The company is a subsidiary of the sports car manufacturer Dr. Ing. h.c. F. Porsche AG, Stuttgart. Porsche Consulting has offices in Stuttgart, Hamburg, Munich, Berlin, Frankfurt am Main, Milan, Paris, São Paulo, Shanghai, Beijing, Atlanta, and Palo Alto. Following the principle of "Strategic vision. Smart implementation," its consultants advise industry leaders on strategy, innovation, performance improvement, and sustainability. Porsche Consulting's network of 12 offices worldwide serves clients in the mobility, industrial goods, life sciences, consumer goods, and financial services sector.

Strategic Vision. Smart Implementation.

As a leading consultancy for putting strategies into practice, we have a clear mission: we generate competitive advantage on the basis of measurable results. We think strategically and act pragmatically. We always focus on people—out of principle. This is because success comes from working together with our clients and their employees. We can only reach our aim if we trigger enthusiasm for necessary changes in everyone involved.

Appendix

References

(1) *Schäuffele (2016). Automotive Software Engineering. Wiesbaden: Springer Fachmedien*

(2) *Jooß (2022, 04). Komplexität von Softwaretests in der Fahrwerksentwicklung. ATZ - Automobiltechnische Zeitschrift*

(3) *Schäfer (2022). Moderne Fahrwerktechnik auf der chassis.tech plus. Fahrwerk*

(4) *Hill (2020, 10). Corona als Treiber der Digitalisierung. Innovative Verwaltung*

(5) *Schäuffele (2016). Automotive Software Engineering. Wiesbaden: Springer Fachmedien*

(6) *Jooß (2022, 04). Komplexität von Softwaretests in der Fahrwerksentwicklung. ATZ - Automobiltechnische Zeitschrift*

(7) *Schäuffele (2016). Automotive Software Engineering. Wiesbaden: Springer Fachmedien*

(8) Maier-Borst (2022). Hebel und Handlungsfelder für die digitale Transformation in der Automobilindustrie am Beispiel der Porsche AG. In T. S. Gerhard Oswald, Digitale Transformation - Fallbeispiele und Branchenanalysen. Wiesbaden: Springer Fachmedien

(9) *Bosler (2021). Das Digitale Dominante Servicedesign – Implikationen für digitale Innovationen im vernetzten Automobil. In P. D. Proff, Making Connected Mobility Work. Wiesbaden: Springer Fachmedien*

(10) Winter (2022). Smart Data, Smart Products, Smart Services – Innovationen und neue Leistungsversprechen in Industrie, Dienstleistung und Handel. In P. D. Prof.Dr.Dr.h.c.mult. Manfred Bruhn, Smart Services. Wiesbaden: Springer Fachmedien

(11) statista (2022, 08 31). Retrieved from statista: https://de.statista.com/statistik/daten/studie/1321740/umfrage/prognose-zu-software-ausgaben-von-automobilherstellern/

(12) Bratzel (2018). Die Rückruf-Trends der globalen Automobilhersteller im Jahr 2017 - Referenzmarkt USA. Bergisch Gladbach: Center of Automotive Management (CAM)

(13) statista (2022, 08 30). Retrieved from https://de.statista.com/statistik/daten/studie/1254342/umfrage/rueckrufaktionen-in-der-automobilindustrie-in-deutschland/

(14) *Kitsios (2021). Der Digital (3D-) Master-Prozess. Springer*

(15) *Kern (2021). Organisation, Planung und Steuerung der Montage im Automobilbau. Wiesbaden: Springer Fachmedien Wiesbaden*

(16) *eveling (2022). IT-Architektur für die Silicon Economy. In P. D. Prof. Dr. Dr. h. c. Michael ten Hompel, Silicon Economy. Berlin Heidelberg: Springer*

(17) *Hanschke (2021). Lösungsbausteine der Digitalisierung. In I. Hanschke, Digitaler Wandel – lean & systematisch. Wiesbaden: Springer Fachmedien*

(18) *Bratzel (2018). Die Rückruf-Trends der globalen Automobilhersteller im Jahr 2017 - Referenzmarkt USA. Bergisch Gladbach: Center of Automotive Management (CAM).*

(19) *Krauter (2022). Certificate-based Safety Concept for Future Dynamic Automotive Electric/Electronic Architectures. In P. D.-C. Prof. Dr. Michael Bargende, 22. Internationales Stuttgarter Symposium. Stuttgart, Germany : Springer Fachmedien Wiesbaden*

Porsche Consulting

Stuttgart | Hamburg | Munich | Berlin | Frankfurt | Milan | Paris | São Paulo | Atlanta | Palo Alto | Shanghai | Beijing

www.porsche-consulting.com