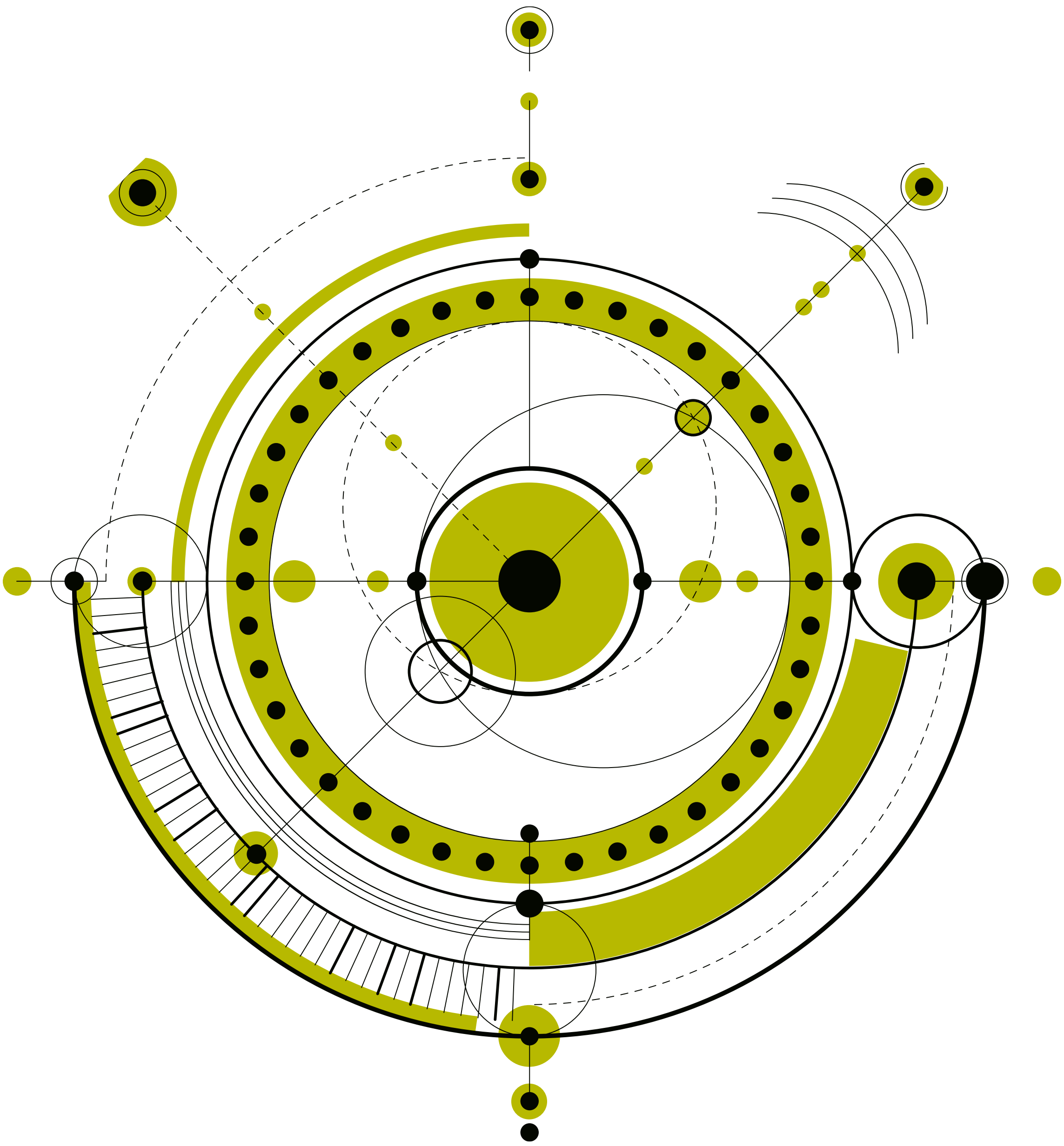
Deloitte.



# Systems Engineering and Agility

## How “Antipodes” Become Best Friends

MAKING AN IMPACT THAT MATTERS since 1845

# Introduction

Across all industries and operating models, organizations have invested billions of dollars into coaching, transformation, and supporting tools to increase their adoption of the agile methodology to improve key metrics such as time to market, flexible development, and brand value. However, these investments have led to marginal improvement in the metrics and have had minimal impact on product development and engineering efficiencies. The push for agile adoption has fallen well short of its potential because it was not paired with the critical best practices of modern Systems Engineering. Systems Engineering and its fundamental goals are the key to unlocking the untapped value of these previous investments and realizing the next level of organizational efficiency, operation, quality, and bottom-line profits.

Increasing digitalization and the associated complexity of products and their interdisciplinary product development pose ever-greater challenges for companies. Today's rigid and firmly defined value chains are increasingly being changed by the trend towards cyber-physical systems (CPS), which are making their way into the industry across all sectors.

This progressive transition to cyber-physical products, which is characterized by data-driven business models and highly networked systems (e.g., IoT), increases the demands on a product and intensifies the associated pressure to innovate. At the same time, increasing customer demand leads to a rising number of different functions and variants of products, while product development cycles are becoming steadily shorter. In addition, especially in the automotive sector, demanding certification requirements and verification obligations for safety and reliability (ISO26262, UNECE CSMS, UNECE SUMS, Automotive SPICE) must be continuously considered while the projected product delivery must be met. These drivers require a shift in thinking away from traditional Systems Engineering to a more agile approach that can better address the dynamic market environment and customer requirements that are constantly changing or changing at short notice. Agile Systems Engineering is the only way to meet the need to develop the systems of tomorrow, as the introduction of agile methods shortens the time-to-market and offers the possibility to react to unpredictability.

### Systems Engineering

Systems engineering is a problem-solving approach to the realization of complex systems. The goal is to create a high-quality product that meets the customer's needs within the framework of diverse technical and economic requirements over the product life cycle. As an interdisciplinary and holistic approach, systems engineering promotes the earliest possible definition of customer requirements and functions of the overall system through stringent processes of requirements and architecture management, which hardly allow any process-related variability.

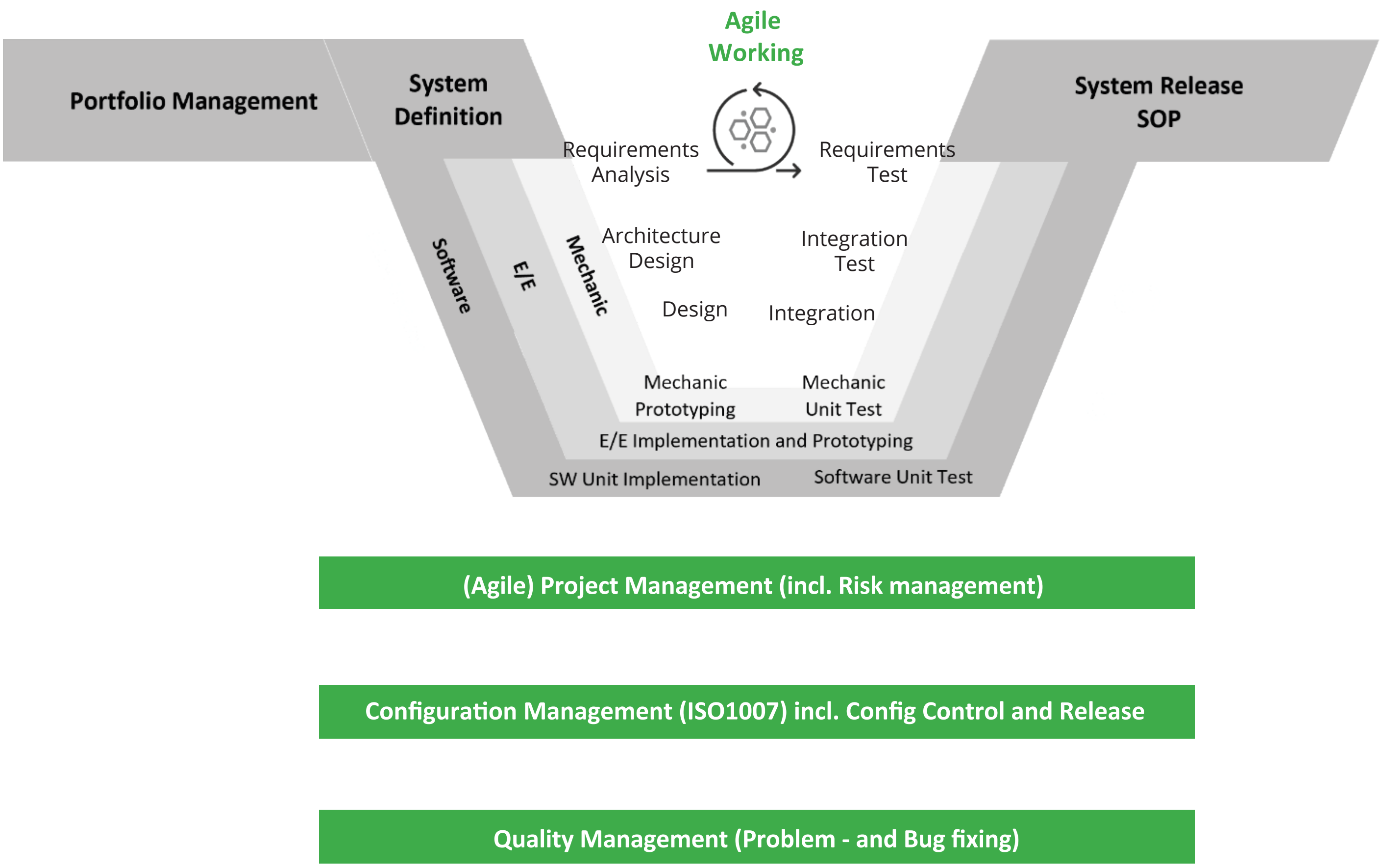
### Agility

Agility describes the ability of a development process to respond individually, flexibly and effectively to unpredictable and dynamic changes in the market environment. The use of Scaled Agile Framework (SAFe) provides organizations with proven agile practices and principles that can be scaled and implemented across business units to increase adaptability along the value chain.

In the environment of new product development and product enhancement beyond the SOP (e.g., over-the-air updates), Systems Engineering is often positioned as the clear antipode to agility in the automotive industry, which is supposedly challenging to reconcile. This is based on the erroneous assumption that Systems Engineering only follows a purely phased waterfall approach, where progress is measured and controlled against a set of specific, one-off milestones, with no continuous and iterative verification and validation.

This Point of View is intended to extend the previous view and demonstrate that Systems Engineering practices are compatible with agile processes and methodologies such as SAFe. Furthermore, no practical solutions exist to integrate both conceptions into a coherent, usable approach to agile Systems Engineering. Therefore, this brief study highlights commonalities and identifies valuable synergies by combining the frameworks to answer recurring questions in anchoring Systems Engineering and agile.

Moreover, it shows that Systems Engineering already provides a suitable framework for agile and faster development in a volatile and complex environment by iteratively and recursively going through the processes of the V-model. Based on our Deloitte experience, we demonstrate a concrete approach for the successful practical implementation of agile Systems Engineering for innovative, intelligent products. Here, we critically contrast challenges and success factors and demonstrate the benefits of a joint synthesis.

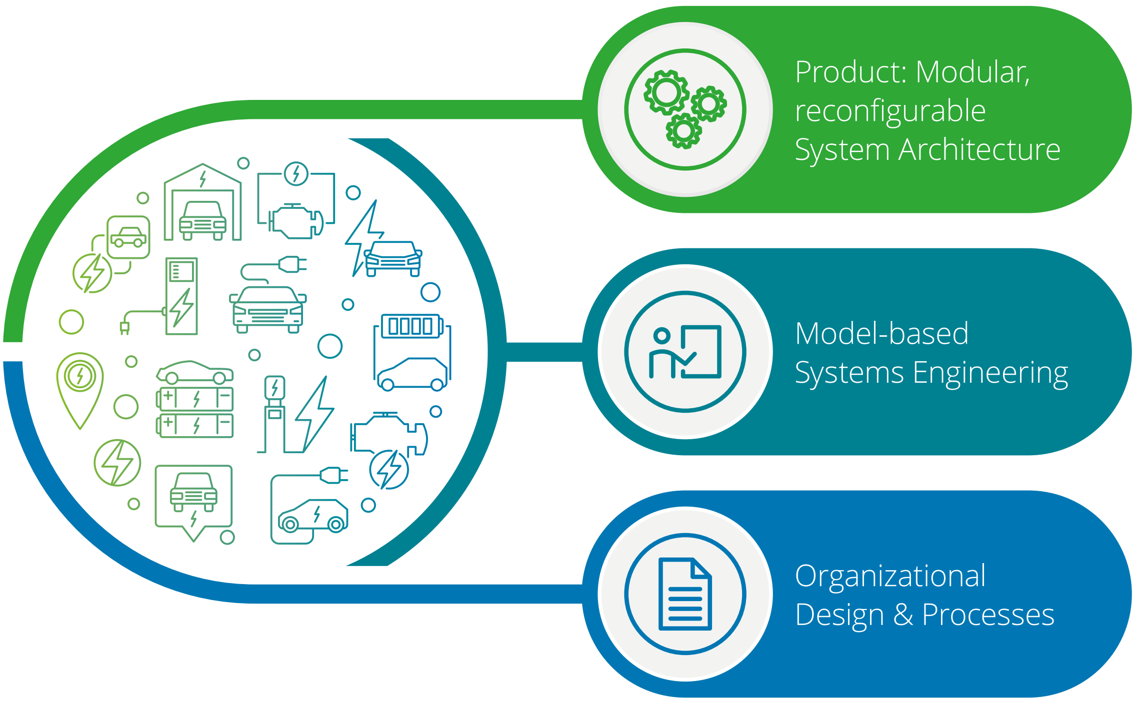


# Systems Engineering is already agile!

The International Council on Systems Engineering (INCOSE) recommends the application of agility in Systems Engineering whenever product development is characterized by unpredictability, uncertainty, risk, and variability.

An agile approach to product development is therefore essential, as the development of cyber-physical systems always takes place in a dynamic environment due to increasing functional complexity and certification requirements. Furthermore, many years of consulting with development managers show that agility is a central building block in Systems Engineering and is already anchored in the development methodological foundation via various manifestations. To this end, we will present a total of three perspectives on how agility is implemented in Systems Engineering. Moreover, through agile practices such as the Scaled Agile Framework, Systems Engineering can be extended to have a comprehensive vision and overall agile understanding of the system with its dependencies.

## Agile Systems Engineering




### Product: Modular, reconfigurable system architecture

Change and dynamic operating environments require the design of sustainable, agile systems. In the literature, agility is often defined as a sustainable system and process capability. Our Deloitte experience in various industries shows that the basic prerequisite for the use of agile action guidelines in Systems Engineering is the establishment of a coherent system architecture, as this is what makes efficient agile work possible in the first place. The system architecture is characterized by function-oriented, hierarchical units that can interact with each other via clearly defined interfaces. The clear assignment of requirements to the individual functions that are independent of each other facilitates the formation of modules and responsibilities. Thus, each module is self-sufficient in its functionality, which favors a fast as well as defined re-configuration of the considered system at any point in time to react to new and short-term changes of the considered requirements.

The clear definition and management of the interfaces of system components, as well as the constant consistency check of the functional scope at each change, allows the system to respond efficiently to new and immediate situational requirements since changes to the system architecture are made according to the 'plug-and-play' principle and new requirements can also be incorporated through an iterative-incremental approach. At the same time, interaction and interface standards reduce complexity and enable individual system configurations.

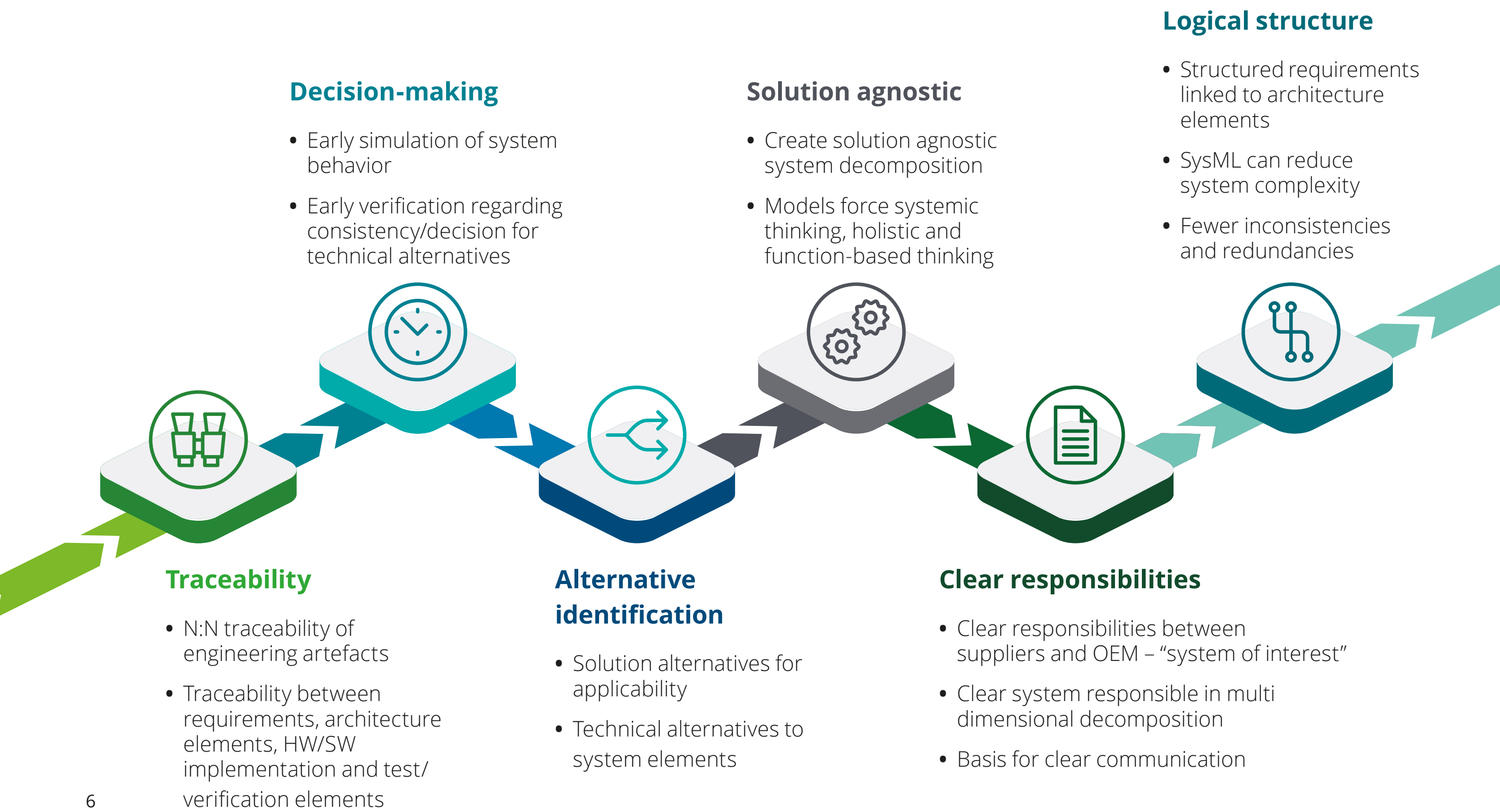
## Model-based Systems Engineering

Another perspective in which agility is already being practiced is the operation of IT systems and development operations (DevOps). In the future, an interdisciplinary and holistic system model will increasingly become the focus of companies and take a central position in addressing technical issues. The starting point for the creation of integrative system models is Model-Based Systems Engineering (MBSE), which creates the framework for system development with standardized interfaces, a uniform modeling language, and suitable processes and methods.

A model-based and function-oriented approach to software development has long since become the standard.

At the same time, increasing progress in computer, memory, and networking technologies have decisively advanced the state of MBSE practice and have led domain-specific areas such as mechanics and electronics to move to a model-based approach as well. This advanced transition to increasingly model-based system development is critical to surviving in an unpredictable and dynamic market environment, as model-based system development with a clear interface definition favors an iterative and recursive approach. This can reduce inconsistencies and enable early, virtual validation of the product as well as a shortened time-to-market. For this purpose, MBSE provides a comprehensive overview of all proposed functions of a system and how they are realized by the system design. It has the advantage that the product is correctly modeled after the first run so that in subsequent release cycles it is already known and adaptations can be implemented more easily and quickly.

A model-centric development approach using the solution agnostic Systems Modeling Language (SysML), in which all dimensions, behaviors,



dependencies, and components of a system are holistically integrated and represented in a model, promotes the ability to meet individual customer needs and rapid changes. Holistic digital modeling, simulation, development, and integration, as well as the iterative-incremental approach to architecture creation, enable continuous testing and validation of requirements, which fosters innovation during the development and implementation phases, as well as after the fact. This allows changing requirements to be effectively incorporated into continuous system improvements until the SOP is achieved. This leads to strong pacing in the development process as well as the ability to evaluate achievable intermediate results and communicate progress transparently

## Organizational design & processes

Traditional hierarchical organizational forms according to functions offer proven structures, which no longer do justice to the digital age. It is falsely assumed that phase-specific product development according to the waterfall principle does not permit an agile approach.

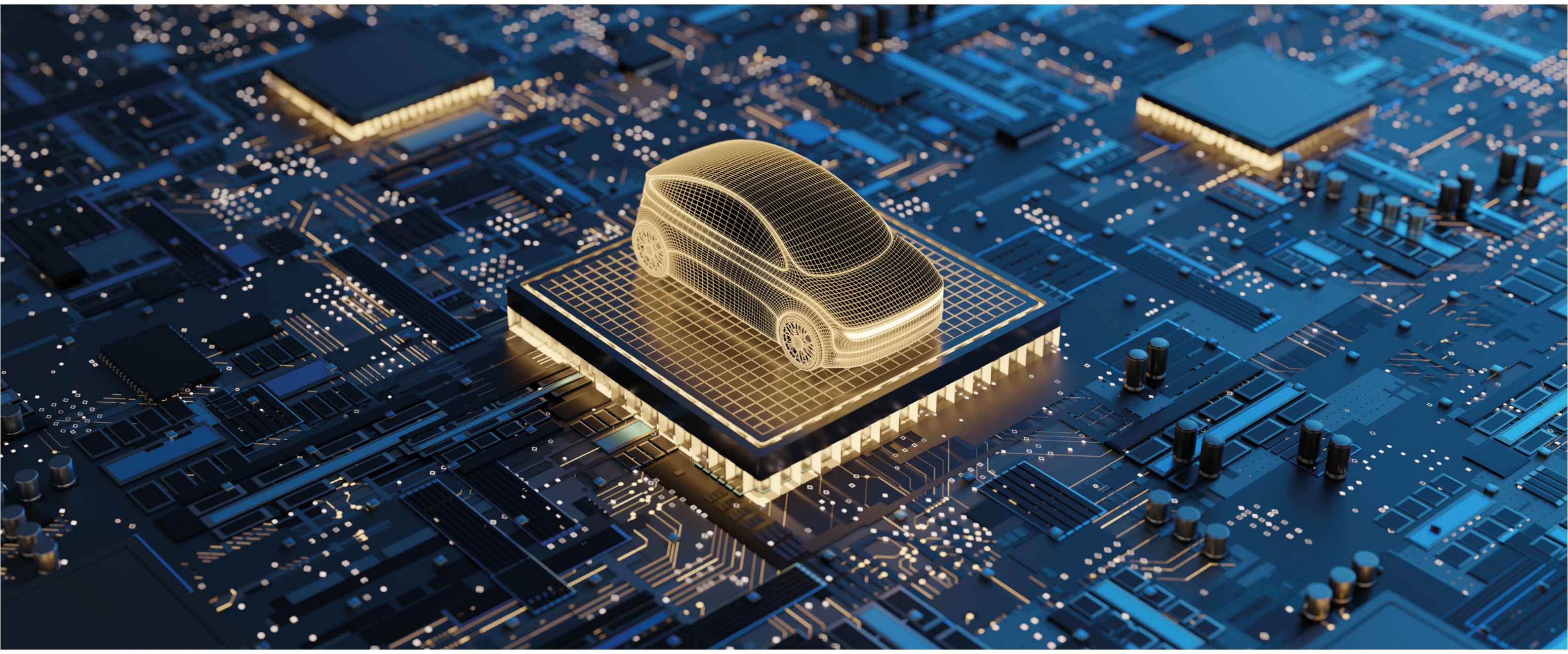
At the earliest possible point in time, the best decision known up to that point is made in an isolated function, which is then implemented seemingly according to plan, only to discover later that the product solution cannot be implemented as such under new conditions. The inherent variability of requirements and architecture over development time is ignored as cyber-physical products are developed to tight costs and delivery schedules.

However, the reality is that Systems Engineering has also evolved over the years. Intended product solutions from the past must be able to adapt to current conditions over time. Thus, new organizational approaches and processes have found their way into Systems Engineering to be able to react flexibly to internal and external variances. In fact, some synergies to the agile approach according to the SAFe framework can be identified here.

At the same time, both approaches can complement each other and use the advantages of the other.

While Systems Engineering focuses on the architecture and interdisciplinary development of technical systems and solutions, SAFe promotes agile collaboration at the organizational and team level. Both concepts pursue similar goals and values in terms

| Roles Systems Engineering | Roles Safe | Description |
|---|---|---|
| Facilitator | Scrum Master | Scrum Master are servant leaders and coaches for an agile team by removing obstacles and creating a high-performance environment |
| Project Lead | Product Owner | Product management is responsible for, defines and prioritizes the program backlog |
| Systems Engineer/Architect | Product Management | Systems Engineer/Architect provides architectural guidance and technical support to teams on the Agile Release Train |
| Function Owner | Systems Engineer/Architect | A product owner defines stories and prioritizies the team backlog |



of strong product and customer orientation, innovation, and systematic value creation. These values are implemented through the collaboration of autonomous, cross-functional, self-organized development teams. This results in a customer-centric decomposition of requirements, similar of the decomposition to a modular system architecture, which is divided into subsystems. SAFe combines several agile teams into an Agile Release Train (ART), which iteratively develops product increments. Lean agile methods and tools are suitable for this purpose, as shown by the way of working in capabilities, epics, features, and stories.

Agile teams are composed of specific roles that combine different views: customer, system, and process. In addition to those responsible for the content of the product/system, technical architects and engineers must implement these customer requirements in a coordinated approach. It is noticeable that many roles from SAFe can be adapted to Systems Engineering and are defined similarly in terms of content. This is also the case for some lean-agile artifacts from SAFe. Thus, the methodological and process-related foundations of Systems Engineering in the form of the V-Model can be embedded relatively easily in an agile organization.

For example, the business agility value stream makes it possible to respond flexibly to business opportunities by ensuring that operational processes interact with product development in the best possible way. Here, an incremental recursive Systems Engineering approach in the sense of "Continuous Integration" and "Continuous Development" is indispensable. Instead of deciding on a single requirement and product variant at an early stage, several requirements and design options are considered while the solution is developed incrementally in short time windows. Each increment represents an integration point and thus its own validated and verified system, demonstrating the feasibility of the solution in the process. Unlike the one-time run of the V-model, each milestone includes a part of each process step – requirements, design, development, testing – which together deliver customer value. Systems Engineering must be understood here as an iterative and recursive approach to product development.

When developing complex systems, local integration points are used so that each system element or capability contributes to the overall solution. These local points must then also be continuously integrated at the next higher system level. Integration points test technical feasibility but can also serve as prototypes to test the market, gather initial customer feedback, and validate the "business hypotheses" from Lean Portfolio Management. In accordance with an agile "test-first" approach, the traditional V-model is thus folded into a very pointed V in the agile procedure and run through several times in short V-cycles by performing testing very early and iteratively in the process. This enables a rapid learning process in the development teams. For many systems, early verification & validation as well as feedback on product conformance is critical to decide whether to manufacture or release products.

# How do we deal with the challenges?

Despite certain existing synergies in the organizational design of Systems Engineering and SAFe, the greatest challenges probably lie in the successful transformation, which requires greater cultural change and a major rethinking on the part of the teams and, above all, the managers.

We at Deloitte have taken a close look at this transformation process and want to show how agile Systems Engineering can nevertheless be successfully implemented in the company and what the potential economic benefits can look like in concrete terms.

### Portfolio Management – Requirements-based Engineering

**Challenge:** The intention of SAFe is to create flexibility and responsiveness early on through small increments, whereas in Systems Engineering, requirements are specified and frozen early on using a classic PO approach and are subject to change management.

**Solution:** The Architectural Runway with its so-called enablers provides the basis for business features to be developed later and enables roadmaps to be synchronized with PI planning in order to take into account both early and later requirements or refinements

### Organization – Communication & Collaboration

**Challenge:** Different terminologies and methodologies lead to difficulties in the collaboration between developers and the agile team.

**Solution:** During the course of the project launch, a common development methodology should be agreed upon, as well as a project definition of terms. In addition, governance of the interfaces should be defined.Process organizations must be linked right from the start.

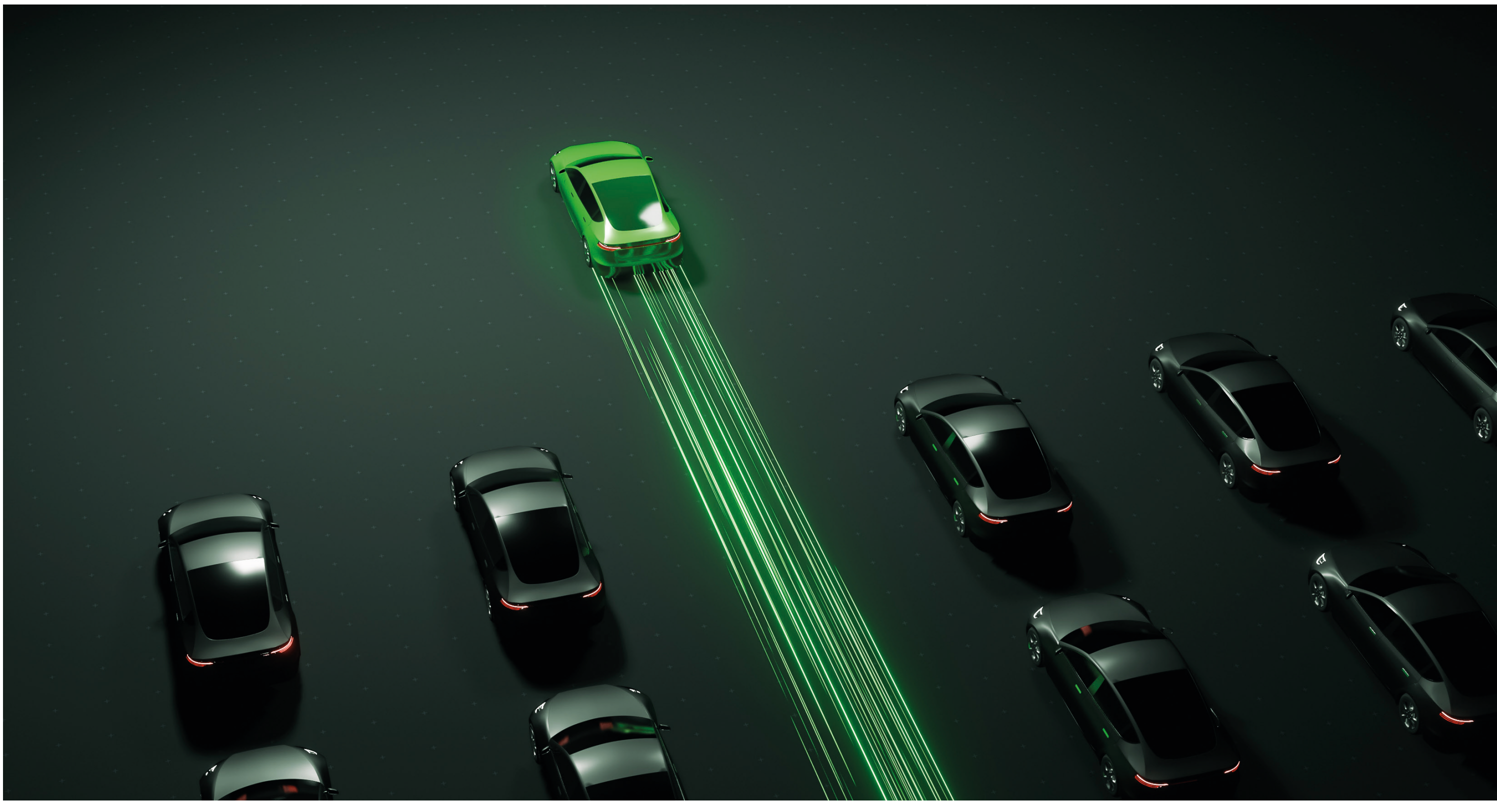
### Testing/quality assurance Safeguarding (Sub)-systems and overall systems

**Challenge:** During joint testing of software and hardware, certain challenges arise in terms of complexity and coordination between the various organizational units.

**Solution:** The complexity of the overall system must be broken down and it must be demonstrated that portfolio and overall system requirements are met. Testing as early as possible and on a regular basis at different test levels makes it possible to uncover errors and reduce costs and development times. Innovation and update cycles simplify coordination and communication.

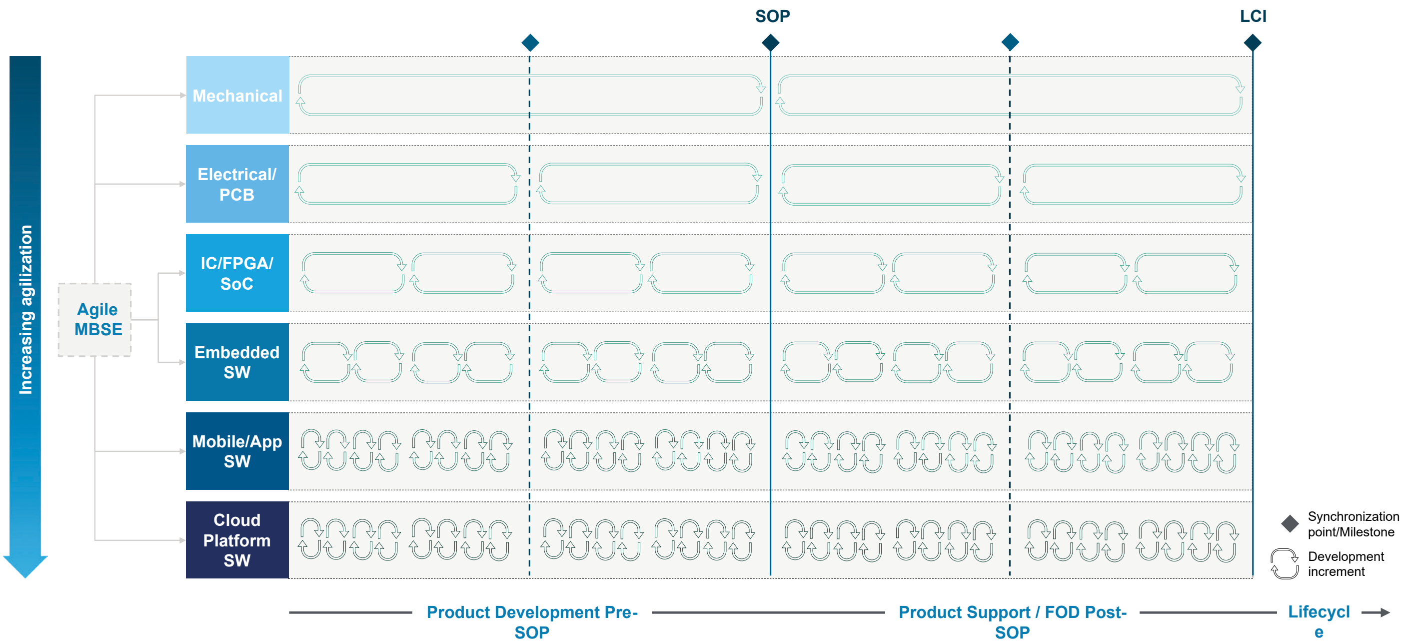
## Portfolio Management (Requirements-based Engineering)

| | |
|---|---|
| **Problem** | The consistent implementation of SAFe and Systems Engineering often fails at the program and portfolio level because traditional product/portfolio management conflicts with an agile approach. Within the development organization, the question often arises as to when specific decisions need to be made or whether they can be delayed. This is because SAFe’s backlogs contain the specifications and the granularity of requirements within agile development comes from so-called refinements in later phases. The intention of SAFe is to create flexibility and responsiveness early on through small increments, whereas, in Systems Engineering, requirements are specified and frozen early on under a traditional PO approach and are subject to change management. At the same time, a problem in the operational application of SAFe is the lack of maintenance of requirements, since in automotive development a consistent system description based on clear requirements is essential for homologation. |
| **Solution** | In our experience, the so-called Architectural Runway from the SAFe context has proven its worth in countering this problem area. This – consistently maintained and kept up to date – helps to avoid or mitigate many classic problems of product development. The Architectural Runway represents the basis for business features to be developed later and can be, for example, a hardware component, code, or defined guidelines. So-called enablers form the runway and thus help to avoid inconsistencies. This is because enablers allow unknown knowledge and decisions to be incorporated into the solution being developed. It also helps to synchronize roadmaps with PI planning to account for both early and later requirements or refinements. In this context, changes to requirements must be flanked by a lean and effective CCB (Change Control Board) at the requirements level and are thus part of the Architectural Runway. A consistent Systems Engineering environment can help here by creating and managing traces from requirements to functions to components or software clusters. |



## Testing/quality assurance – safeguarding (sub)-systems and overall systems

| | |
|---|---|
| **Problem** | In addition, our experience shows that certain complexity challenges arise when testing software and hardware together. Coordination between different organizational units is difficult, but for the continuity of a combination of SAFe and Systems Engineering, it is essential to synchronize the time windows of agile, hybrid-agile, and non-agile systems. |
| **Solution** | To counteract this, the complexity of the overall system must be broken down and it must be demonstrated that portfolio and overall system requirements are met. Testing as early as possible and on a regular basis at different test levels make it possible to uncover errors and reduce costs as well as development times. For example, based on virtualized simulation environments, the software can be tested independently of the hardware. This requires joint planning and coordination of all stakeholders of both organizational units. Together, they decide which systems will be built in an agile or hybrid-agile way and which systems will provide a stable basis around which to develop. To this end, innovation or update cycles must be established to ensure effective coordination and a continuous flow of communication. |



## Organizational challenge and process organization

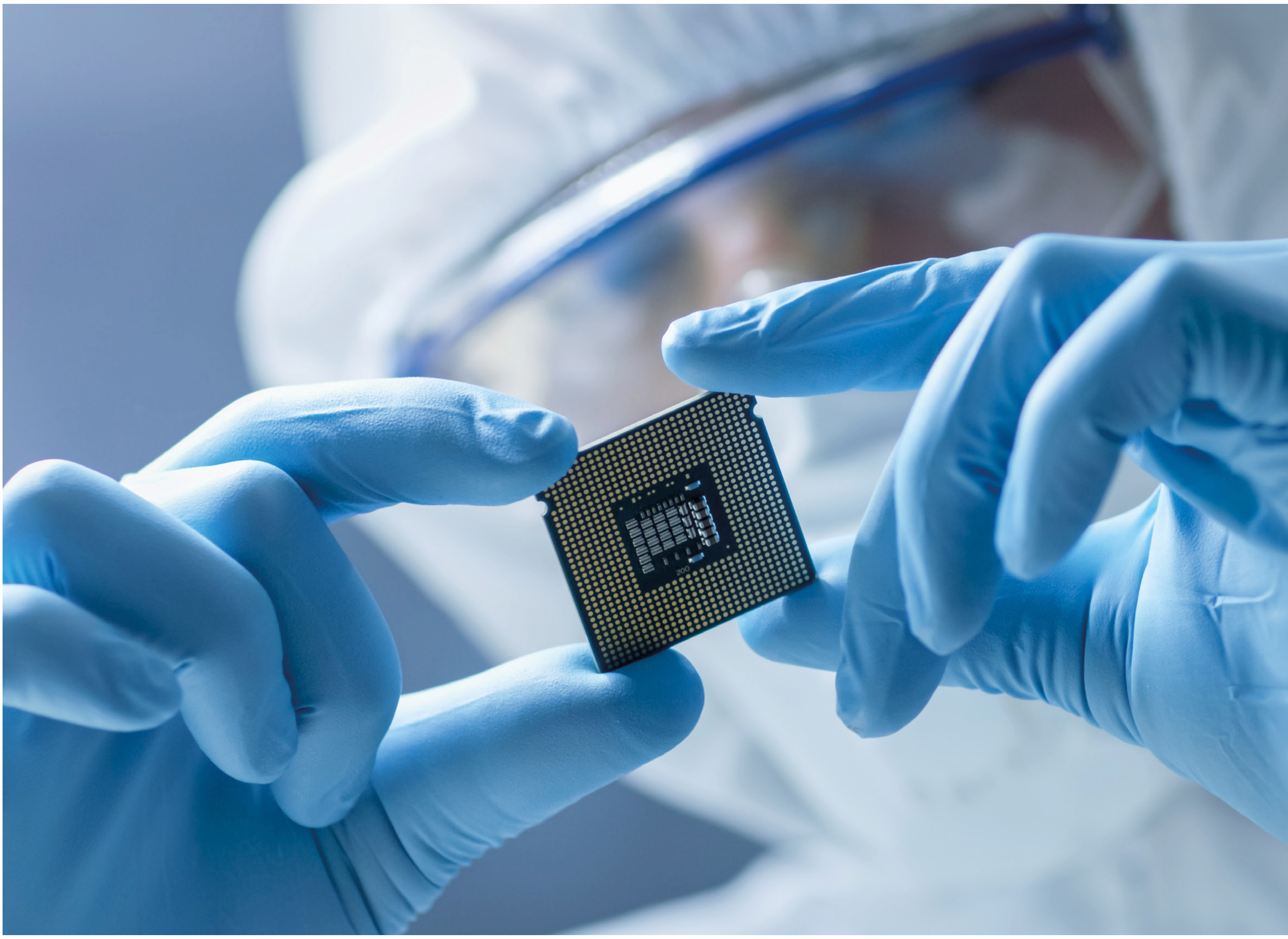
| | |
|---|---|
| **Problem** | Another challenge is the communication between engineers and the agile team. While the engineers and developers fall back on the terminology of Systems Engineering, the agile team is guided by the SAFe vocabulary. Thus, language barriers can occur which permanently hinder communication, collaboration, and ultimately development progress. |
| **Solution** | One of our lessons learned is therefore to agree on a common development methodology and a project definition of terms right from the start of the project. An understanding of both methods is essential, and employees should be sufficiently trained in advance for both SAFe and Systems Engineering to be able to work efficiently in this context. Furthermore, governance of the interfaces should be defined. From the very beginning, process organizations must be linked and the SE milestone plan must be merged with PI planning. An overarching solution train can be used to control when integration points are reached. In addition, dependencies can be identified and decoupled at an early stage, e.g., in a program backlog. This simplifies coordination and communication between the development teams. |

# Outlook

At Deloitte, we rarely see companies fully master the complexity that arises in interdisciplinary product development. Often, the benefits of linking Systems Engineering and agile practices such as SAFe are not holistically captured, which in our experience leaves a lot of potential untapped in development departments. We have many years of expertise and the necessary skills to make agile Systems Engineering tangible in companies and to create a comprehensive vision and an agile overall understanding of the system with its dependencies on an organizational, process related as well as architectural level.

We therefore actively support the synergy between Systems Engineering and agile practices such as SAFe in development organizations across industries to fully exploit the value creation within product development and to meet the increasing functional complexity and certification requirements in a dynamic environment.

We offer custom-fit solutions to the challenges of regaining control of product development with our numerous best practices that have emerged from many years of consulting with customers in a wide range of industries. Only in this way can we meet the need to develop the systems of tomorrow with an agile system engineering approach.



# Contact person

**Sassan Mottaghian**
Director
Systems Engineering
smottaghian@deloitte.de
+49 89 290365978

**Lena Holzner**
Manager
Systems Engineering
lholzner@deloitte.de
+49 89 290366748

**Henning Ohde**

**Christoph Wolfbeisz**

**Nico Wenzel**

**Abdelaziz Daroui**

Issue 07/2023