ptc

July 2024

Software-defined Vehicles

Impact of Agile Product Development

Foreword

It wasn't subtle. 'Everything about carmaking is changing at once,' wrote *The Economist* in April 2023 in its special report on the industry. Whether it's keeping up with customer demands or complex regulations, automotive OEMs are finding that software is as important as hardware in making it first to market with the highest quality products. The new king of the road is the software-defined vehicle (SDV).

In this world it is the software developer, not the hardware engineer, who rules. Toyota, the world's largest automobile manufacturer, expects to have 18,000 software engineers by 2025. Honda plans to have 10,000 software developers working for it by 2030. More than one publication has asserted that software is as important as hardware in cars.

Software is involved in almost all the functions of modern cars: engine control, heating and cooling, power steering, transmission, parking assisting, locks, lighting, navigation, and the communication and infotainment options customers expect in their 'third space' between work and home. With up to 100 million lines of code some cars have become as complex as aircraft. Car makers are now buying one another – for their software.

What's Changed

Cars have become 4-wheeled computers that can see, sense and act more and more autonomously. The hardware subsystems that sense and act must be closely integrated with the software.

Consider a radar system in a vehicle's side mirrors that indicates a car following closely behind. In the past this mirror had a separate controller, but it was not truly connected to the central car computer. The Tier1-supplier provided both the hardware and the software for this feature. In the case of a lane-changing assistant, however, the necessary connections multiply. This radar is connected to steering and braking, and thus must be connected to a central car computer and then across sensors and actors from more and more subsystems.

Car manufacturers must take collaboration between their mechanical, electrical, electronics, and software engineers to a higher level and do so deliberately and thoughtfully. Both the expectation for innovation and the penalties for failure are high.

When those systems are not integrated, **delays pile up**, **software-related recalls** increase, and embarrassed manufacturers find themselves having to tell **customers** that promised features will not be available when owners take delivery of their vehicles.

This article explains how automotive OEMs can improve collaboration among mechanical, electrical, software, and production by using agile product development. Agile allows physical product developers to create and refine new products with reduced time to value, greater transparency, and less risk. Well established in software, agile development methods are finding their way into the world of hardware as automakers discover their need for speed and responsiveness has never been greater.

At The Starting Line

Automakers have to prepare. Management and functional experts must plan for fast-changing vehicle electrical/electronic (E/E) architectures and increased product feature variability. They need to integrate vehicle software in the platform concepts with separate lifecycles and Over-the-Air (OTA) updates, ensure cybersecurity and safety, coordinate with suppliers, and speed up manufacturing development to meet faster delivery schedules.

To achieve these product and production enhancements, engineering must be enhanced too. Below, McKinsey & Company highlights capability needs (on the right) emerging because of the changing development environment.

The same old ways won't do.

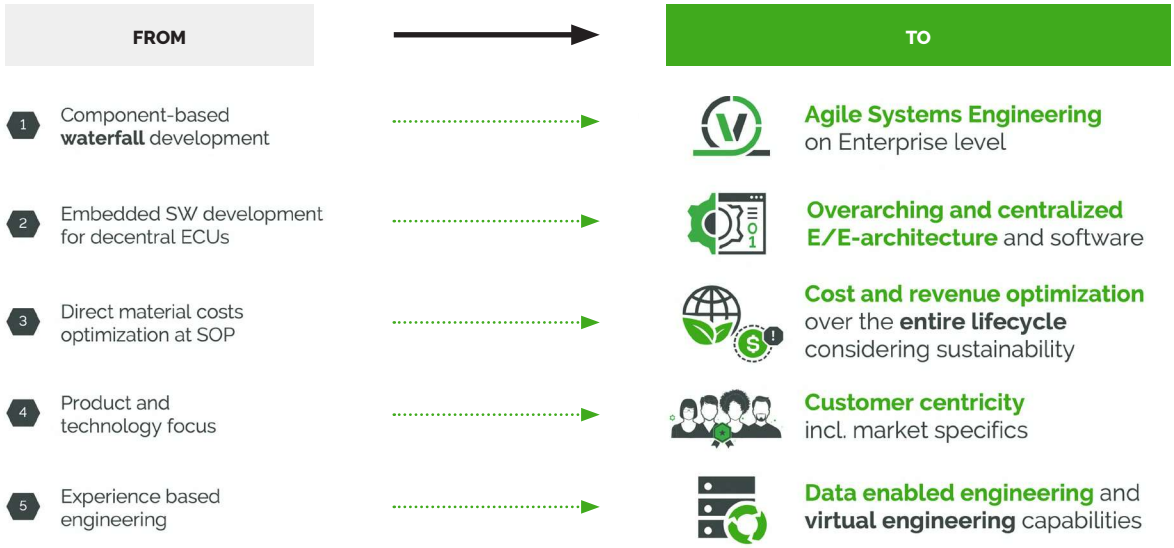


Figure 1: Five emerging game-changing capabilities in R&D
Source: McKinsey & Company / GenAI in A&D / October 2023

Picking Up Speed: Applying Software Development Methods to Hardware

Hardware development has long relied on the waterfall method, a linear process well-adapted to the consideration of lead times, committed costs, and production planning timing.The goal was to produce a product correctly the first time. Paper waterfall charts could run the width of office walls.

Developed in the past 20 years, the Agile approach to software development rejects linearity. In the words of the **Manifesto for Agile Software Development**, Agile prioritizes "responding to change over following a plan." Businesses using agile techniques may rely on daily huddles, focused time-boxed sprints, frequent customer feedback, and regular reflection on what worked well and what didn't. Agile puts a premium on communication because as the direction of product development changes course, so do the project dependencies.

Waterfall and Agile development methodologies and their adherents can clash, leaving the business saddled with delays, lower development velocity, quality issues in design, and problems with efficient procurement and integration. This needn't be the case. The same results can be achieved with Agile methods (as with waterfall) with good process discipline supported by software solutions that embody best practices (e.g. via project templates).

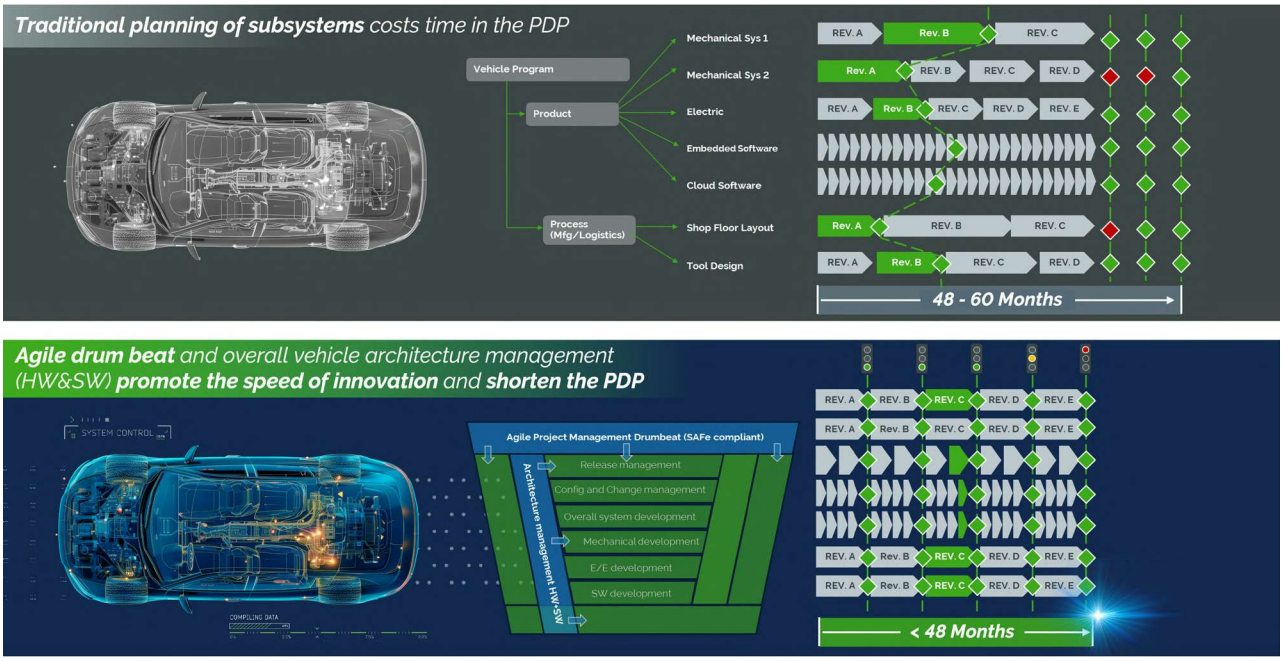
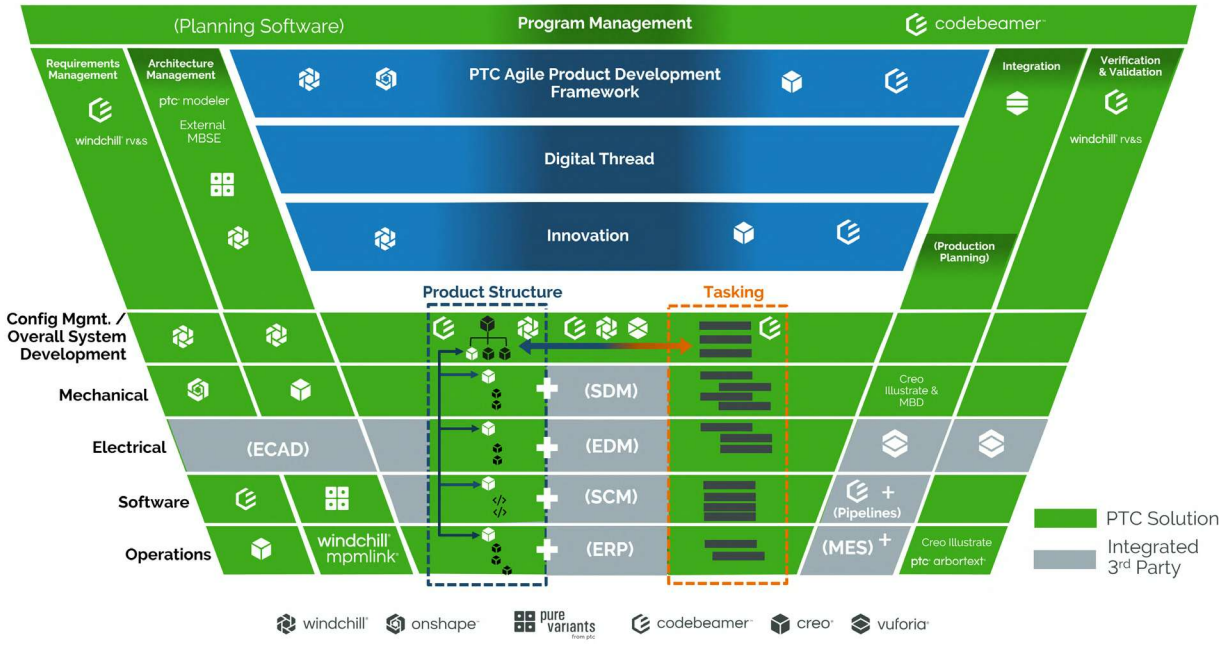


Figure 2: Aligned Agile development methodology provides an enterprise-wide drumbeat to synchronize all deliverables.

With traditional waterfall methods, different subsystems were developed independently and integrated late in the process - if ever. This sequence is shown in the upper part of Figure 2. Compare this to the lower section of Figure 2. With the time-boxing of Agile, these systems and subsystems now automatically have common release points, giving the effort a high-frequency rhythm. This way of working allows for earlier and more frequent integration cross-checks and overall status validation.

PTC Portfolio for Software-defined Vehicles & Agile Project Execution

Figure 3: PTC Solution Portfolio Across the Systems V



The V Diagram

The development cycle for vehicles follows the V-Model to design systematically for quality. The classic V diagram of systems engineering consists of three sequential parts read left to right. The left section shows requirements and architecture management, the bottom section is the implementation phase in different domains, while the right side shows the final steps of integration, verification and validation.

The PTC Solution

Starting in the upper left, PTC's solutions cover:

- Requirements management
- Model-based systems engineering (MBSE)
- Platform architecture
- Variant management
- Mechanical design, simulation and test management
- Configuration and change management
- Production planning

Much of the V diagram is green because PTC offers integrated (not stand-alone) solutions across every part of the SDV development lifecycle with an open architecture strategy.

Such a strategy allows customers to sidestep the trap of vendor lock-in. Vendor lock-in occurs when software buyers discover they are locked into a vendor because that vendor's products, often by design, do not integrate easily with other systems and components.

The items shown in grey are best-in-class outside toolsets that integrate closely with PTC's. Such broad reach, and the attendant flexibility offered by open standards, makes possible agile and software-integrated methods and project execution.

"The worst decisions in business are the ones you don't realize you're making."

– Jon Hirschtick, co-founder SolidWorks and Onshape.

Squandered Opportunities: A technical note for non-technical professionals

When highly skilled employees can't find the data they need, they don't say that data is siloed – they say it's lost.

To the degree that systems are integrated, these issues can be avoided, and people can get what they need to speed the completion of a high quality SDV.

PTC's solution provides the depth and range needed for SDV development. It's designed as a holistic solution for systems, software and physical platforms engineering – the opposite of a corporate maze of people and disciplines with their own languages.

- ALM for requirements management (what the product will be and its variants and test cases) via **Codebeamer** and **pure::variants**
- CAD & PLM for managing the product definition (design and manufacture the product and its variants as planned) via **Creo** and **Windchill**
- MBSE for designing how the product's subunits will work together via **PTC Modeler**
- PTC's solutions above come with built-in project templates and tools to get a quick start on best practices from agile methodologies. The templates can also be tweaked to allow for customer-specific process needs.
- 3rd party alternatives which work smoothly with PTC products through the well-documented APIs and the broad set of supported integration standards.

The SDV is Pulling Up. Are You Ready?

Buyers want the vehicles they want. Consumers are the easier ones to satisfy due to a relatively stable "base" mechanical and electronic configuration beneath the surface of most makes and models. Industrial customers present issues that are more confounding: innumerable hardware and software subsystems that must work together.

With PTC's integrated solution portfolio, automakers can keep pace and so do profitably. The outcome is a more productive and successful engineering organization that can deliver more compelling products at a faster pace and lower cost.

We call that winning.

Learn more about Agile Product Development

For deeper insights read *Agile Development: Develop Hardware Like Software* - the paper gives an overview on how to approach this methodology more effectively.

Speak with an automotive expert, to explore how PTC can support your software defined vehicle initiatives.

121 Seaport Blvd, Boston, MA 02210 : ptc.com