The software-defined vehicle and how to transform the established hardware-centric development process

Andrea Denger, Program Manager Information Management, Software Products, Emission & Service, AVL List GmbH
Peter Ebner, Chief Engineer Hybrid Systems, Systems Engineering & Powertrain Electrification, AVL List GmbH
Katrin Moser, Marketing Manager, Integrated and Open Development Platform, AVL List GmbH
Josef Zehetner, Chief Engineer System Architecture, Integrated and Open Development Platform, AVL List GmbH

What do a photo album and a vehicle have in common? In the past, everything evolved around hardware, the physical objects you can touch. In recent decades, functions have expanded, and the importance and influence of software have increased. However, software alone is not the solution. Just as you can't view your digital photos without a smartphone or tablet, you can't drive software on its own to get to your desired destination.

The driving experience of mobility consumers will continue to be dominated by hardware and its functionality. It will be determined by what they can haptically grasp, physically feel, and how they can engage with the vehicle functions. Mobility does not happen in the metaverse – a real person interacts with the car and physically moves to their destination. Consumers are not interested in using a piece of software, they are interested in a product that provides customized functions. For a satisfying customer experience, a needs-based end-to-end approach is essential.

There is a new buzzword in the automotive industry: software-defined vehicle. For example, the distinction between vehicle variants is shifting from hardware to pure software and calibration data (i.e., using the same engine hardware but different power levels). The vehicle could become a software-defined IoT (Internet of Things) object. And there is no doubt that this transformation offers many more opportunities – from digital business models to digital services, connectivity and cloud services, and new types of collaboration or partnerships.

But we must not forget that it is still the hardware that brings the software functions or customer experience into our world. This hardware is developed, tested, delivered, and maintained at the OEM's site. In addition, the hardware remains largely unchanged over its lifetime and must be reliable, durable, and secure. Consequently, the software that operates this hardware must not alter the hardware's main attributes and must use the available customer and environment interfaces.

What would this require in terms of software-defined vehicles? How can we adapt the traditional development process, or do we need a new development approach? In this article, we will discuss two areas that have great potential:

- Virtual development with Software-in-the-Loop (SiL) systems
- Continuous verification and validation with virtual, mixed and real prototypes

Let's take a closer look at each of them.

Virtual Software Development

The challenge with physical testing (i.e., testing hardware and software together) is not only that it costs a lot of time and money. The available hardware prototypes are expensive and limited in the number of variants. Any bug identified could originate either from the software or from hardware that is not working properly.

Modern development processes and toolchains try to reduce verification with real hardware to a limited number of loops (e.g., to prove the achievement of quality gates) for one variant. All further testing loops and calibration tasks for software or functional variants are performed using a SiL approach that considers a working virtual hardware in combination with the control system software. This drastically reduces the development effort, the time and cost required, and the resources needed on the hardware side (Figure 1).

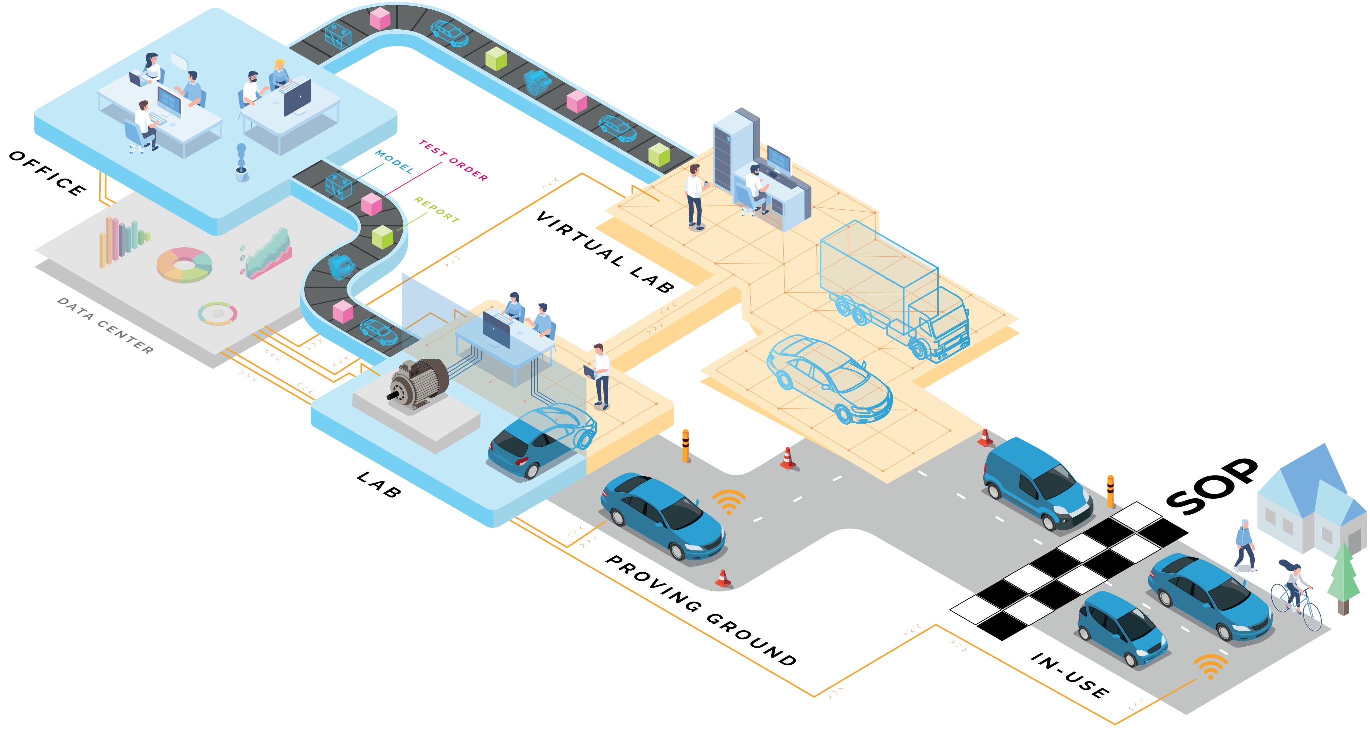

Figure 1: *Virtualization opens up a wide range of possibilities in the modern development process*

We just have to decide:
- Which tests should we run for which variant?
- Which tests can we reuse without adaptation?
- Which tests do we not need to repeat for each variant?

In addition, we can also take into account:
- Variations in the product due to aging or production tolerances
- Unusual usage and misuse

This specific type of frontloading also allows integration tests to be performed for two or more control systems within a single SiL system to verify the interaction of these – usually independently developed – software modules. For example, we can run a test with an engine ECU, a transmission ECU and an emission model - testing an entire global validation matrix in one night on the SiL system to identify critical cases or issues related to emissions with the interaction between ECUs.

Complexity will continue to increase due to topics such as over-the-air (OTA) software updates in the in-use phase. In addition to simple enhancements of functionality (with no impact on the hardware), drastic changes to the overall system behavior are also conceivable (or have already been implemented). Take, for example, increasing the charging power of a battery in the field. What kind of validation would that require? In the field, there are both new batteries and batteries that have already been driven a significant amount of their lifetime mileage. What needs to be done to ensure that this new function in the vehicle meets all requirements – both in terms of homologation and in terms of safety, reliability, robustness, warranty and so on?

Model-based systems engineering (MBSE) is one of the most important methods for structuring complexity and establishing context between hardware and software or the associated interfaces. MBSE also makes it possible to generate the specific functional view (end-to-end) of a product, including the corresponding hardware and software parts.

Continuous Verification & Validation (V&V)

The transformation from a hardware-centric and product-structure driven development towards a software-defined, vehicle-centric development has several implications:
- The number of functions in the vehicle realized by means of software will increase continuously, even after the start of production (SOP) or during the in-use phase
- The development effort for software functions will increase drastically, e.g., due to ADAS/AD functions
- Software functions must be reused across vehicle variants and types (e.g., by modularization)
- Clearly defined interfaces between hardware and software will be crucial

The goal is to develop software and hardware decoupled from each other: Hardware following the traditional V-process, software using an agile DevOps approach. This allows each one to operate at its own speed, while also allowing to validate the software against all vehicle variants in Continuous Integration/Continuous Deployment (CI/CD) cycles by bringing hardware and software together (Figure 2). For this we need virtual, mixed and real prototypes of the vehicle hardware.

With the help of functional prototypes, we can assess the state of a product throughout the development project: we compare the achieved test results of virtual, mixed or real prototypes with the defined target requirements.

Summary

The transformation of the car to a software-defined vehicle is already underway. To get the most out of your development processes, you need to rethink the way you work:

- Intelligently distribute the development tasks between the hardware and the virtual environment
- Define clear interfaces between hardware and software
- Use continuous verification and validation to test software and hardware together
- MBSE provides the required context and information

The DevOps process and software development will enable significantly faster release cycles of new or improved functions to consumers. The increased frequency will lead to a paradigm shift in validation: Pure simulation in product development and continuous validation and implementation up to the in-use phase will massively gain in importance.

You also need to rethink the way you work with others: In the context of digital transformation and model-based thinking, processes become closed-loop approaches – from requirements to design and validation, with unambiguous interfaces between hardware and software. This is followed by production in a connected smart factory, knowing every delivery state of a car, as basis for downstream OTA services in the in-use phase, for example. Clearly defined standards and guidelines as well as a general openness of systems will enable companies to overcome their own limits of interoperability and achieve common benefits.

Contact

Andrea Denger
Program Manager Information Management, Software Products, Emission & Service
AVL List GmbH
+43 316 787 8756
andrea.denger@avl.com

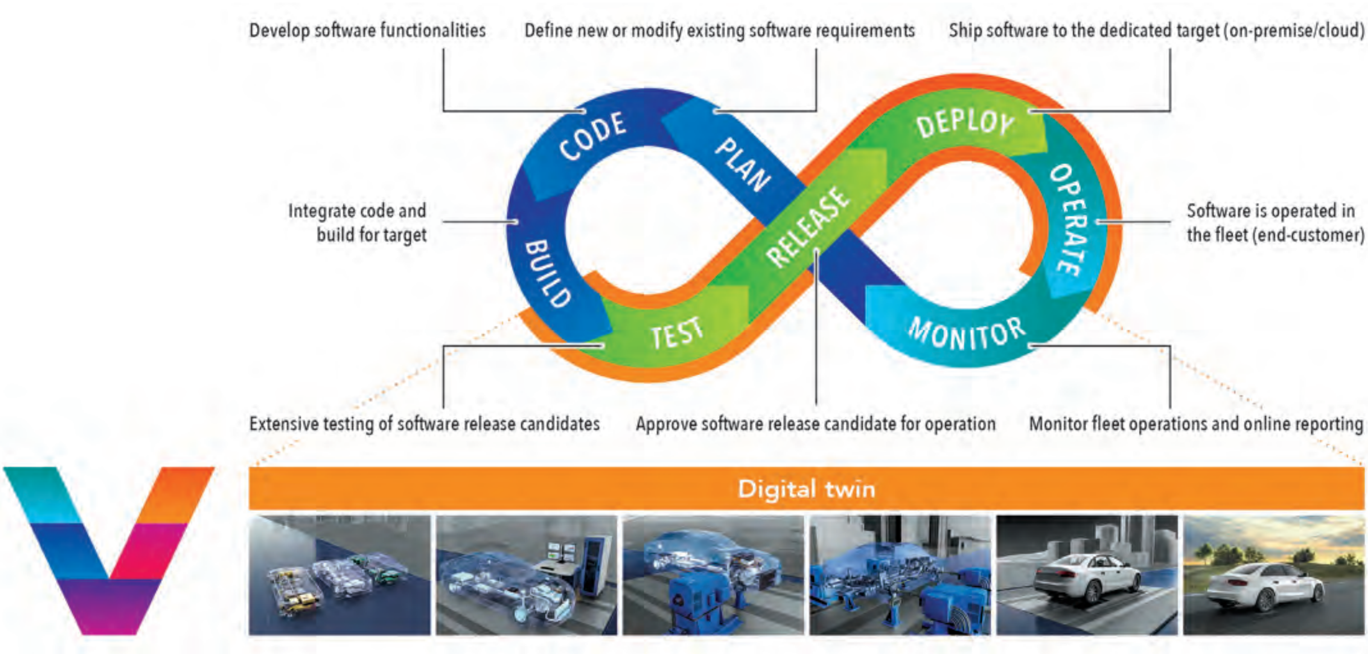

Figure 2: *DevOps and V-Process connected in a symbiotic overall process*

Contact

Peter Ebner
Chief Engineer Hybrid Systems, Systems Engineering & Powertrain Electrification
AVL List GmbH
+43 316 787 3299
peter.ebner@avl.com

Contact

Katrin Moser
Marketing Manager, Integrated and Open Development Platform
AVL List GmbH
+43 316 787 7345
katrin.moser@avl.com

Contact

Josef Zehetner
Chief Engineer System Architecture, Integrated and Open Development Platform
AVL List GmbH
+43 316 787 1765
josef.zehetner@avl.com

prostep ivip web seminars

Exchange – Information – Inspiration

Another service that the prostep ivip Association provides to its members is participation in a monthly web seminar free of charge. This series of events usually takes place on the last Friday of a given month and comprises a half-hour presentation followed by a discussion among the international group of participants.

You can find out more about the web seminars and how to register by contacting Sarah.Giese@prostep.com or online at www.prostep.org.

**Only some of the web seminars are open to non-members*

WEB SEMINAR

The Dates for 2022:

30. September 2022 DDP
Members only

28. October 2022 New Project Proposals
Open to non-members

25. November 2022 ReqIF
Open to non-members