BinaryCore

1 June 2022

Unlocking speed, quality, and cost: how to optimize hardware development in an embedded software environment

So far in this series, we've covered many things around automotive software, from the prioritization of features to end-to-end software development. Now we turn our attention to the role of hardware within an embedded software environment and how challenger OEMs are rapidly deploying software to ECUs (electric control units) as in-vehicle hardware. In this article, we look at the challenges this presents to established OEMs, the fully automated delivery model disruptors use to stay ahead, and the key levers enabling excellence in end-to-end hardware development.

Challenger OEMs outpace traditional players with automated deployment and synchronized updates

For many OEMs, ECU provisioning is still a manual process. It’s a common occurrence to see engineers physically connecting to test stands for flashing ECUs, staging test sets, and playback reports. This time-consuming method of deploying software can take days and sometimes even weeks if the right hardware isn’t on hand. In contrast, challenger OEMs use the advantage of OTA (over-the-air) update infrastructure already in testing, making it a background process.

Leaders in this space ensure the latest software version always fits the current hardware-in-the-loop (HIL) by syncing hardware and software updates in good configuration management and fixed update cycles. If a hardware update has taken place, the software developers must know to push the corresponding software. In the same way, if new software has been developed for the base system but the new hardware has not been tested, the lack of compatibility can lead to major issues. To prevent this from happening, challenger OEMs operate on a tight schedule with strict deadlines, making software updates consistently every two weeks within a six-week hardware update cycle.

In the absence of a carefully regulated system and supplier dependencies, traditional OEMs may only sync hardware and software systems every three months. Often, they are left waiting on suppliers operating on separate toolchains, limiting the speed at which updates can be made. While OEMs don't need to develop everything themselves, by taking on the role of software integrator they enjoy the benefits of building the overall software package and testing it with the hardware in-house. This integration massively shortens the test frequency gap between competitors, avoids delays and potential crashes, and allows OEMs to develop in a truly agile manner.

Tech players and the advantages of a two-speed delivery model

Tech players in the sector are operating in a two-speed delivery model to operate like software companies. For the slower hardware platform and base OS development, we observe a traditional waterfall approach along the V-model with a six-week sample logic. Having a slower pace for hardware platform development and a faster pace for software feature development creates a stable platform and provides the foundation for greater innovation at the application layer. Tesla is known for optimized use of this model, where their two pipelines integrate into the central system more frequently for software features than for hardware features (see figure 1).

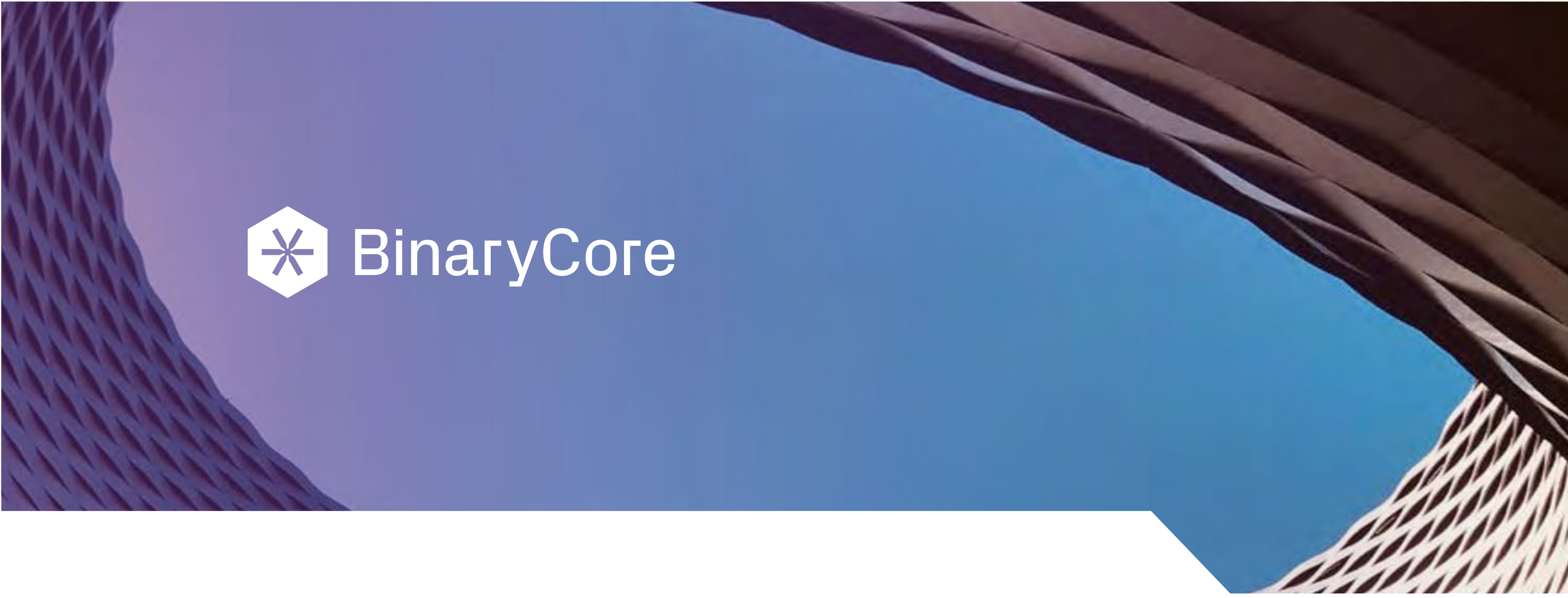
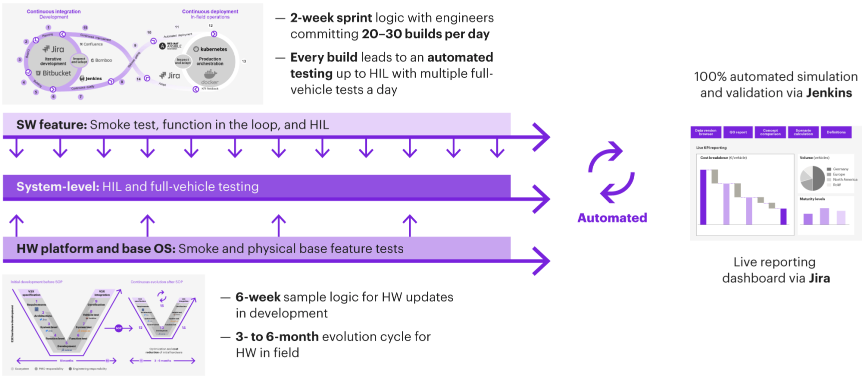


Figure 1
A slower pace for hardware platform development and a faster pace for software feature development creates a stable platform while providing the foundation for greater innovation at the application layer

Continuous synchronization during development sprints



Notes: HIL is hardware-in-the-loop. HW is hardware. OS is operating system. SOP is start of production. V2X is vehicle-to-everything.
Source: Kearney analysis

One of the major advantages of this model is that it unlocks huge efficiencies in time and cost. Because the system integration level is entirely automated regarding provisioning and regression testing, it allows for a lean maintenance team of a few people per model servicing around 50 test stands. With this setup, application developers can regularly commit software into the repository, triggering around 20 to 30 full vehicle software builds per day. To put this in context, traditional OEMs might have a same-size team servicing only one test stand and deploying software builds somewhere between every two weeks and three months.

Software builds using this model are mainly tested on HIL, where ECUs are connected to the central computer which simulates all the data without having to use the full vehicle. Additionally, challengers are able to perform up to two full-vehicle regression tests every day. Such high test frequency combined with the fixed regression test and resulting transparency, connected to the Jira requirements, allows for near real-time transparency on maturity grade and progress reporting. Without this overview and insight, traditional OEMs are more likely to approve every new software that comes in, often spending huge amounts of money without being able to measure the quality or qualify the impact.

Best-practice end-to-end hardware development

Referencing our extensive Automotive Software Benchmark Index, we identified two optimal V-cycle time lines for end-to-end hardware development and 15 key levers along the way. In this process, the initial ECU development before the start of production (SOP) takes up to 18 months, compared to a traditional setup of 24 to 30 months (see figure 2).

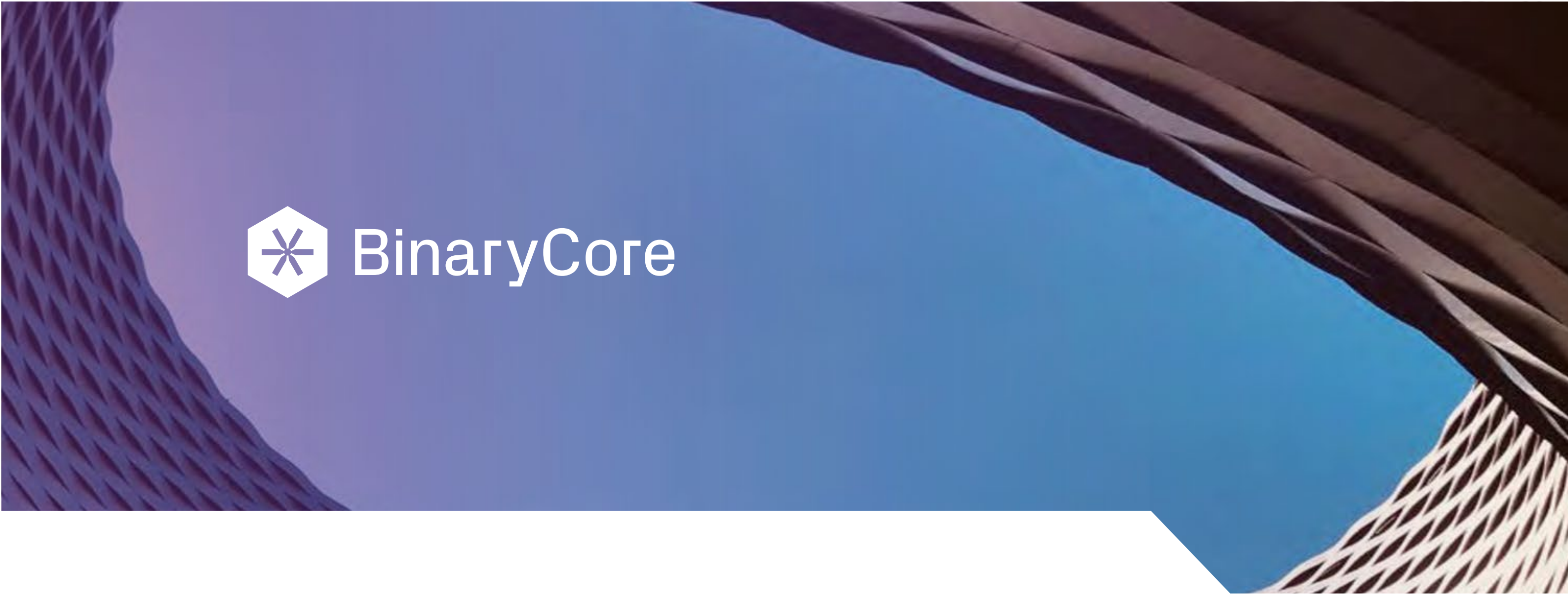
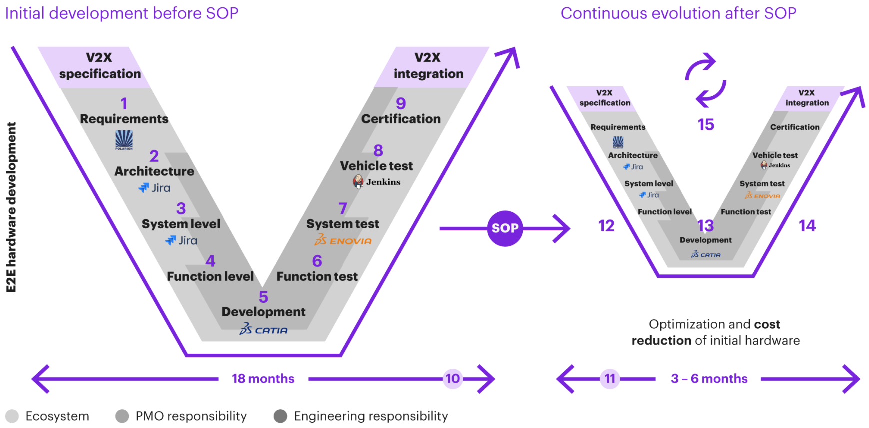


Figure 2
Best-practice E2E hardware development



Notes: E2E is end-to-end. SOP is start of production. V2X is vehicle-to-everything.
Source: Kearney analysis

15 key levers

Pre-SOP specification

1. End-to-end integrated requirement engineering
2. Zonal ECU architecture
3. Early expert validation within product teams
4. Hardware over specification on parts and power overhead

Pre-SOP development

5. Tooling and central CAD model storage

Visit www.binarycore.com for more information.

BinaryCore

Pre-SOP integration and testing

6. Hardware and software synchronization
7. In-house testing and rapid prototyping
8. Automated full-vehicle testing
9. Risk-based shortening of industrialization and homologation
10. 18-month development cycle in sample logic

Post-SOP optimization and updates

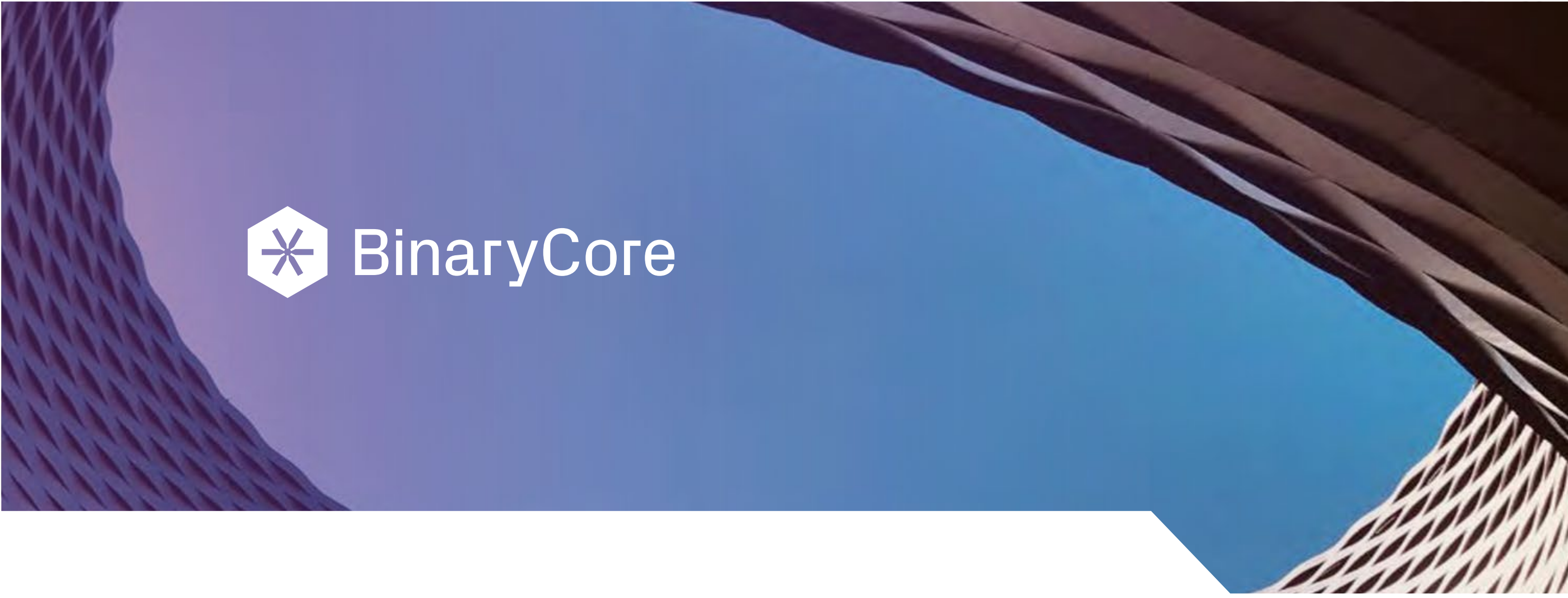
11. Continuous evolution and optimization based on usage data
12. Re-specification of systems in vehicle behavior board
13. Minor and major product changes committed to the central model
14. Integration, testing, and self-certification
15. In-field hardware upgrades and central configuration management

Challenger OEMs and the continuous optimization

Of these 15 levers, some of the most crucial fall within post-SOP optimization. At the start of production, ECUs and mechanical hardware are often over-specified in power and durability, leading to part redundancy. Once released, actual tester and customer field data is used to optimize these parts and derive updates. These updates are then released in a three- or six-month cycle, depending on the scale of the change.

The front trunk closing system is a great example of this continuous optimization. When initially launched, the original setup was for 10,000 openings. However, Tesla soon saw from user data that the function was only being used in the first three months of ownership when customers were showing it to friends. Based on this insight, the durability of the closing system was reduced for future models by a factor of 10, to around 1,000 openings and closings. As well as significantly reducing costs, this optimization models true customer centricity, innovating around the customer by understanding exactly what they do and don't need.

The same principle holds for ECU optimization and part reduction. At the start of production, challenger OEMs will double vital parts on the ECUs just in case one of them breaks. In the event of a fault, the backup part can be activated via software update. Similarly, if the parts prove good enough, they will be stripped down to the most efficient configuration. Where more traditional OEMs tend to optimize before the start of production, at times spending more than a year optimizing components, challenger OEMs are free to add as many features as they like over time, without being forced into a trade-off. This approach is a game-changer in the production of hardware and the reduction of time-to-market.



Want to know more about how to optimize hardware development?

For more insight on the approach and how we can help, or to discuss access to our software features database please contact socialmedia@binarycore.com

▸ Authors

Felix Kreichgauer, Michael Roemer, Marcus Weber, Sebastian Werner

Visit www.binarycore.com for more information.

BinaryCore

Footer