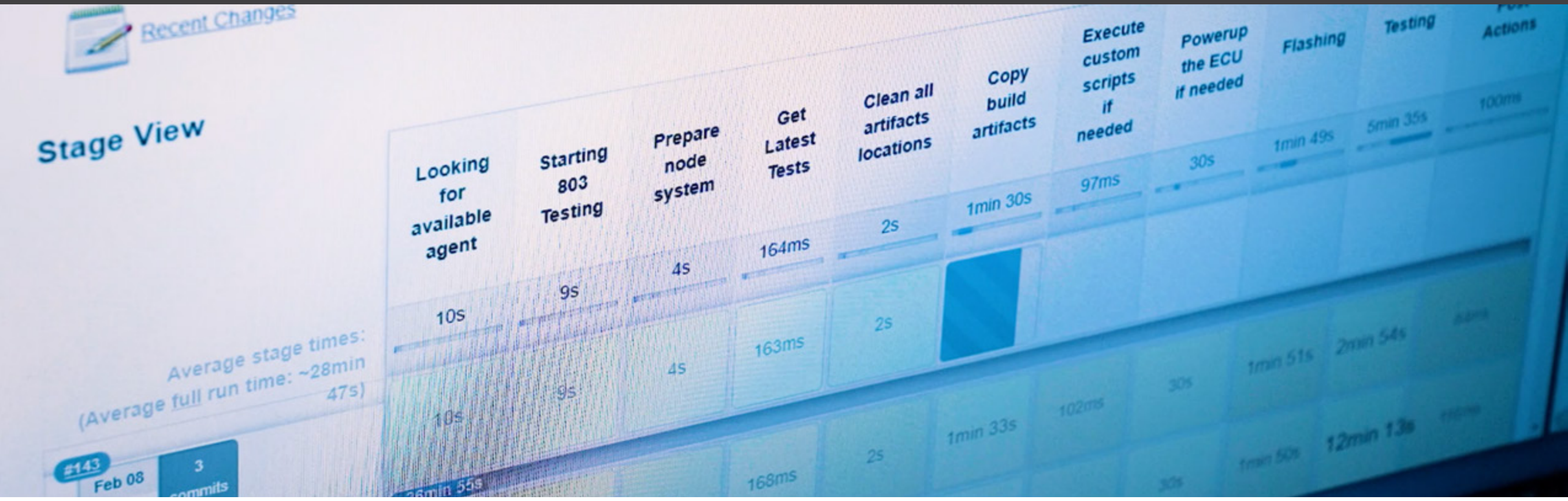
APTIV

WHITE PAPER



Continuous Testing Critical to Automotive Software Development

The automotive industry's transition to software-defined vehicles makes software testing more important than ever. Fortunately, testing methods are quickly evolving to enable better code, safer vehicles, shorter time to market and ongoing updates over the life of the product.

Testing is taking place earlier, in combination with agile, iterative development, while automation and cloud-enabled processes are making testing faster and more scalable. These breakthroughs are helping OEMs and suppliers meet the requirements of advanced vehicle platforms.

THE AUTOMOTIVE SOFTWARE REVOLUTION

The sheer amount of software required by a modern vehicle has grown to tens of millions of lines of code, putting pressure on OEMs and suppliers to write, deploy and integrate code more quickly and efficiently. New testing methods are accelerating that process.

At the same time, software has graduated from enabling infotainment and engine functions to controlling new safety-critical features such as advanced driver-assistance systems (ADAS) and autonomous driving systems, raising the stakes and vastly increasing the complexity of testing.

The pace of technological change puts pressure on OEMs to incorporate new features closer to the start of production and even after vehicles have been sold. Developers need short feedback loops, enabled by testing, to continually update code without lengthy approval processes.

A sea change in development

Changes in testing are part of a broader transition in the way the industry develops both software and hardware.

Traditionally, developers have written software for each hardware component and then integrated it with code for other parts of the vehicle. Testing of the integrated software has come late in the process, limiting the time for additional changes. Development of each component and vehicle platform has been a one-off process that starts over again for the next platform.

OEMs are beginning to shift from this incremental, vertically integrated approach to more agile, iterative methods in which independent teams continuously write, integrate and test their own code. Horizontal layers of middleware take the place of code specific to each component or vehicle platform. Applications and functions are integrated via application programming interfaces (APIs) shared among the development teams. This continuous integration/ continuous deployment (CI/CD) method lets development teams update their code throughout the life of a vehicle and reuse proven code from other components and platforms to minimize cost and errors.

The new software methods also help OEMs implement new architectures in which processing moves to large, centralized domain controllers from smaller electronic control units, reducing cost and complexity.

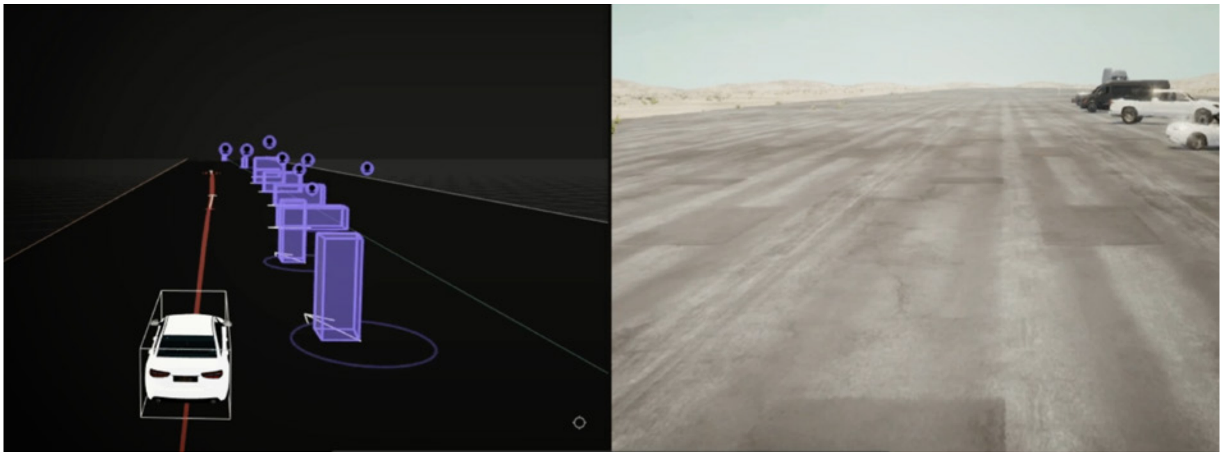
Continuous testing supports iterative development

New testing methods are part of this fundamental change. Testing software as it is being developed — to meet new demands for speed and scalability — requires the division of code into components linked by APIs and the use of automated tests on those components. Testing early and often, using automation, and adopting cloud platforms can improve all three stages of testing: software-in-the-loop (SIL), hardware-in-the-loop (HIL) and vehicle-in-the-loop (VIL).

SOFTWARE-IN-THE-LOOP

Because it is prohibitively expensive to test new software in vehicles on the road, OEMs first simulate both hardware and driving scenarios in software and run automated tests that are faster and repeatable. This testing can be performed on any standard desktop computer (or computing instance), so it is scalable and lets developers work anywhere. Results can easily be shared among teams at OEMs, suppliers and third parties. SIL testing by independent teams using CI/CD methods can shorten developers' feedback loops from months to minutes.

A key part of SIL testing is ensuring that applications can be integrated via APIs. This can be accelerated by testing applications against a "mock," a minimal piece of software that only answers API calls. Testing against a mock ensures that a pair of applications can be integrated using the stable API rather than any hidden functionality that might change in the future and break the integration. This approach streamlines SIL testing so developers can run, develop and test software on one system.



Software-in-the-loop: *With SIL, simulations occur entirely in software.*

HARDWARE-IN-THE-LOOP

The next step, HIL testing, runs hardware components on a test bench through simulated driving situations. Using test scripts based on data from live driving, a simulator gives inputs to components such as cameras and radars to simulate road scenarios. Based on those inputs, the sensors send signals to software running on an ECU, and testers evaluate the software's performance.

Test scripts can combine many variables: For example, a test script might simulate a vehicle going around a curve at 60 mph in the rain and encountering an oncoming car swerving across the center line. Cameras and radars send image data based on that event to the ECU, where the data is processed and the software decides on a course of action.

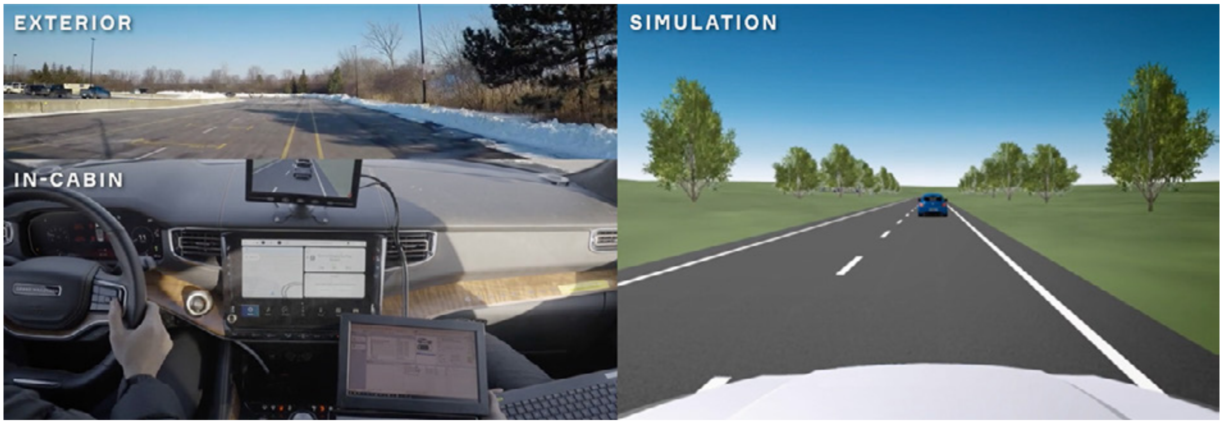
Hardware-in-the-loop: *A HIL bench presents simulated inputs to real hardware.*

VEHICLE-IN-THE-LOOP

Advanced simulation technology offers yet another way to test vehicle software against real-world situations without the cost and time commitment of full-scale road testing. VIL testing evaluates an entire vehicle, with a driver on a closed course, but uses sensor inputs instead of actual hazards and other conditions. It lets developers study how the whole vehicle works as a system executing complex functions such as parking assistance, automated lane-changing, emergency braking and autonomous driving.

For example, a test script could give inputs to onboard sensors to simulate a careless pedestrian suddenly stepping in front of the vehicle. When the ECU received data from those sensors, it would respond by sending a signal to apply the brakes. The test would show how the sensors, ECU and brakes work together.

Using VIL, OEMs can safely evaluate safety-critical software, such as ADAS and automated driving features, on vehicles already on the road. This is done by updating the software over the air but temporarily preventing the new code from controlling actuators. The updated software can then process inputs from sensors in real-world situations, helping developers refine its responses. This method also lets OEMs refine other features, such as infotainment options, by analyzing how drivers and passengers use them. Some OEMs plan to use postproduction VIL testing on models coming in the middle of this decade.



Vehicle-in-the-loop: *In VIL, simulated inputs are fed to real hardware in a full vehicle.*

GETTING THERE

Automakers already recognize the value of the new testing methods and in some cases are already using them. However, the testing must be implemented as part of a larger change in automotive software development and business models.

For example, the traditional approach of developing software for each hardware component, on a project basis, makes testing a one-time process. The way most developer tools are licensed now does not even allow developers to implement test automation. An automated testing capability, along with collaborative CI/CD development and timely over-the-air updates, adds value to automotive software. But current business models, which are based on suppliers selling hardware to OEMs, do not recognize that value.

When business models have evolved, suppliers and OEMs will be able to treat continuous testing, at all stages, as a vital part of ongoing development. Collaborative advancements in testing will benefit all partners in the creation of next-generation vehicles.

As part of its implementation of CI/CD software development over the past five years, Aptiv has automated SIL and HIL testing and adopted a global, cloud-based architecture for central control of test benches. These advances have significantly reduced daily build times and demonstrated the benefits of continuous, automated testing for vehicle development. By leading the transformation of testing, Aptiv is paving the way for OEMs to accelerate and improve automotive software development.

ABOUT THE AUTHORS

Olaf Kammel
Global Director, Test & Validation

Olaf Kammel is passionate about efficient system integration and testing. He has more than 20 years of engineering and leadership experience in the telecommunications, electronics and automotive industries, spanning multiple development, engineering operations and leadership roles. In his current assignment, Olaf leads the global Aptiv Advanced Safety & User Experience (AS&UX) Test & Validation organization under the umbrella of Global Engineering Services. His organization performs all AS&UX product hardware and software qualification, vehicle system testing in Europe, and product cybersecurity verification during development cycles. He is based in Aptiv's technical center in Krakow, Poland.

Justin L. Koegle
Global Chief Engineer, HIL/VIL/Simulation, Test & Validation

Justin L. Koegle is a forward-looking innovator in test automation, simulation and processes. With 10 years of experience in test design, architecture and software quality, Justin has held various leadership positions in electrification and advanced safety, including directing Aptiv's global HIL farm. He is currently the global chief engineer of the HIL/VIL/Simulation organization, focusing on virtual simulation and the development of complex, fully automated system integration and system test environments. Justin continues to use his talents to drive higher efficiency in Aptiv's software release processes and methods.

LEARN MORE AT APTIV.COM/ADVANCED-SAFETY →