What the Next Phase of Automotive Software Development Looks Like

For years, the automotive industry has been evolving toward the software-defined vehicle, in which more features and functions are primarily enabled through software that can be updated quickly and easily.

Tens of millions of lines of code power today's vehicles — and yet the industry has only scratched the surface. Massive transformations in connectivity, autonomy and user experience are still to come, and each innovation will require complex software to run it.

Ensuring that developers have superior tools, processes and structures to create, test and update that software efficiently is a top priority for the entire industry.

The answer lies in continuous integration and continuous deployment (CI/CD), but the platform must have a cloud-native development and orchestration environment that is powerful enough to both enable real-time, global collaboration and perform the rigorous, extensive testing demanded by automotive applications.

A BETTER APPROACH

As software takes vehicles to new levels of safety, comfort and convenience, developers now need software development environments that are much more modern and capable than legacy systems.

In the past, software development followed the rigid, slow "waterfall" method and used highly fragmented toolchains. Development would go through discrete phases, with each phase being completed before the next began. Many of the development processes were manual, as were handoffs from one part of the toolchain to the next. It would take six to eight months for a new version of the software to emerge in that approach.

Today, software is becoming more abstracted from hardware in the automotive world, and developers have an opportunity to take advantage of that architectural shift. They can use modern agile and DevOps methods to update the software on its own schedule, much more quickly and free from ties to hardware changes or other physical updates. This approach provides unprecedented levels of speed, scalability, quality and security.

In continuous integration (CI), the work of building software is automated, and all of the steps of software production are included in a "CI chain," as the output of one part becomes the input to the next. Continuous deployment (CD) is the automated deployment of new versions of software in the field. CI/CD — as well as continuous testing (CT) — is now possible in automotive, but testing new, advanced applications presents a unique problem in the automotive industry, because tests need to be conducted on a physical vehicle or in a complex simulation on a test bench. For example, imagine the complexity of testing software that analyzes input from multiple radars and cameras in order to autonomously execute a maneuver like passing a slower car on the highway or navigating a complex urban environment with vehicles, pedestrians and other obstacles.

The ideal approach eliminates bottlenecks in the development and testing processes through a cloud-based, centralized platform that enables developers to execute any tests on related test benches located in any test center in the world. The tests can put software-in-the-loop (SIL), hardware-in-the-loop (HIL) or even the vehicle-in-the-loop (VIL) for ever more realistic testing. This approach allows for the scale required to tackle today's toughest challenges, using global resources.

STEP CHANGE NEEDED

"The transition from combustion engine to electric vehicles (EVs) alone would have been enough to rock the industry, but innovation through software has become a game changer," writes Boston Consulting Group (BCG). "In the midst of this unprecedented shift within the mobility industry, automakers need to accelerate speed to market and become much more responsive to customers' requirements. ... Companies that continue to work in the traditional way won't achieve the step change needed to compete in this new environment." Agile delivers many advantages, including speed to market, reduced risk and better products, according to BCG.

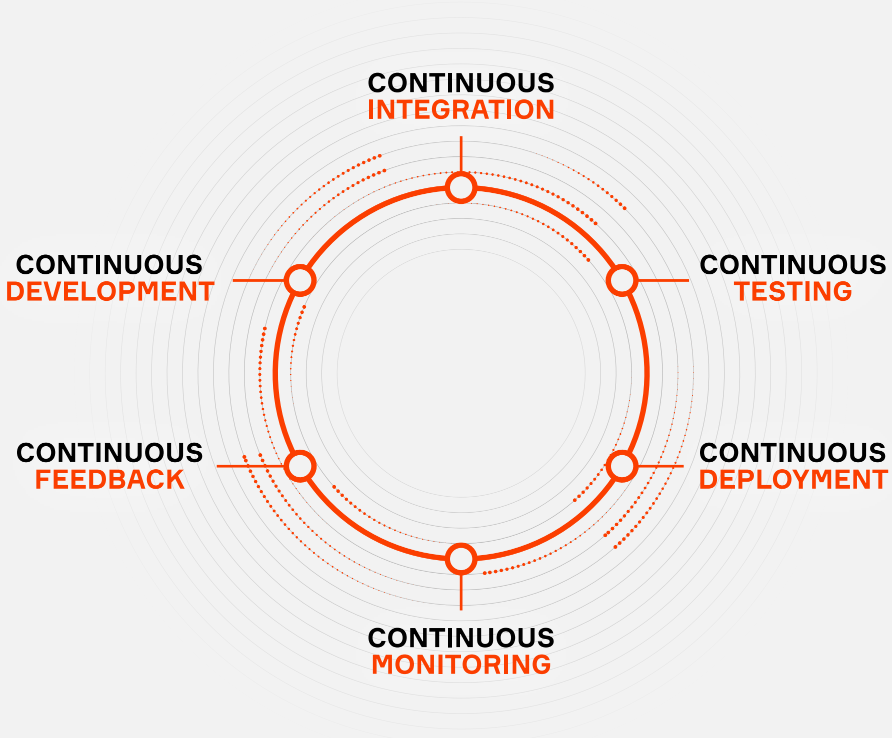
PROBLEMS WITH LEGACY APPROACHES

When developers are writing millions of lines of code a day, legacy approaches to developing, testing, integrating and deploying software are unsustainable. The requirement for HIL testing has meant that as code changes have been integrated into the software during CI, the CI systems have had to be physically located next to HIL benches. This approach has numerous drawbacks:

- A single build could take many hours, when it needs to be completed in several minutes or less. This is a significant bottleneck that affects speed to market as well as quality, because longer builds often result in less time for executing automated testing. When builds take too long, it consumes valuable time that would be better used for additional testing or feature development.
- Legacy infrastructure is expensive and lacks scale, security and flexibility between programs.
- Every program is started from scratch, with no reuse of code or use of off-the-shelf solutions.
- Without a universally accepted software methodology, shadow IT inevitably creeps in, which means developers sometimes use their own preferred tools. Not only does this create inefficiencies, but it also can become a security risk because IT lacks visibility into the development process. And it is harder for developers to collaborate with customers or each other when they are using different toolsets.
- CI chains contributing to the same software program have been distributed across multiple technical centers around the world – each with its own HIL test benches. Without a standard approach, companies have not been able to develop in the cloud and to achieve the consistency and repeatability that comes with that global standardization.
- With multiple teams or even suppliers contributing to today's mobility software products, increases in integration points

CONTINUOUS DELIVERY

With a comprehensive continuous-delivery approach, each action informs the next in an ongoing cycle.



have driven increases in quality, delivery and timing issues. The best practice is to integrate code early and often, but the traditional approach creates inflexible gates during the development process, resulting in issues that cannot be tested until the next merge gate. This results in unnecessary delays to the overall project.

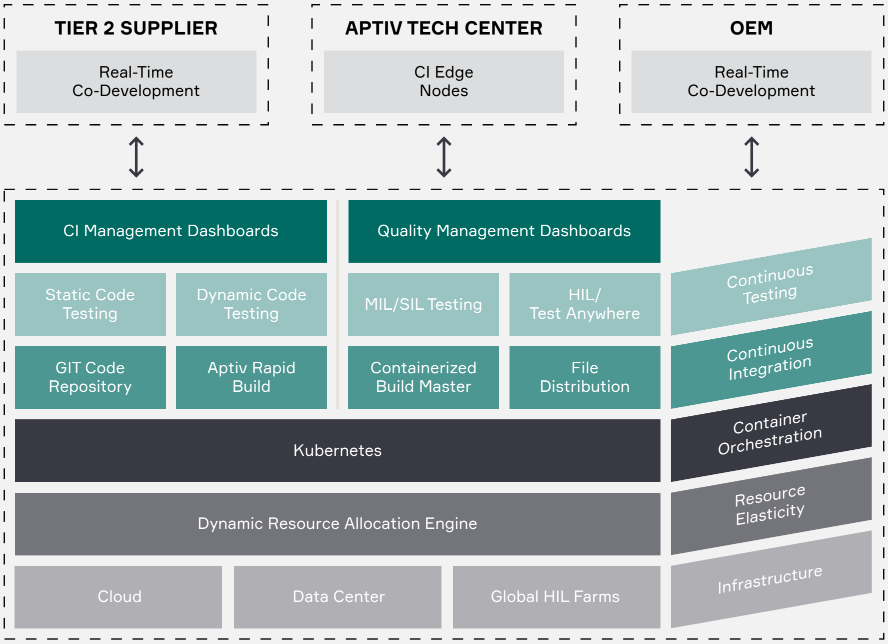
TO THE CLOUD

Because vehicles are life-safety systems, automotive companies have long taken a requirement-centric approach to designing and testing them. They validate systems against a list of requirements to ensure that safety and legal considerations are properly addressed — and in fact, the ISO 26262 functional safety standard for the industry results in the need to test each requirement of a function. Put simply, manufacturers have to prove that their solutions meet specs on day one — and with every update post-launch, they have to ensure that everything that worked yesterday still works today.

Over the past 10 years, the number of software requirements in auto industry applications has grown from hundreds to tens of thousands. Many of those requirements must be tested with the embedded software running on the same electronic control unit (ECU) hardware that will reside in the vehicle. The tests also have to be performed in real time to ensure that response times are predictable, which means that each test takes several minutes, on average. On top of that, software is often distributed across multiple ECUs, sensors and other devices and could be developed by equally distributed teams at different companies.

CONTINUOUS INTEGRATION

The next generation of continuous integration requires a broad set of cloud-based tools, brought together in a way that emphasizes program quality and velocity.



These factors have led to companies using multiple CI chains, in multiple locations, all with their own HIL test benches and all contributing to the same program.

A better approach is the creation of a single, consolidated toolchain for each program. That toolchain should be cloud-based and globally available. And there should be a mechanism for the quick and easy deployment of new toolchains.

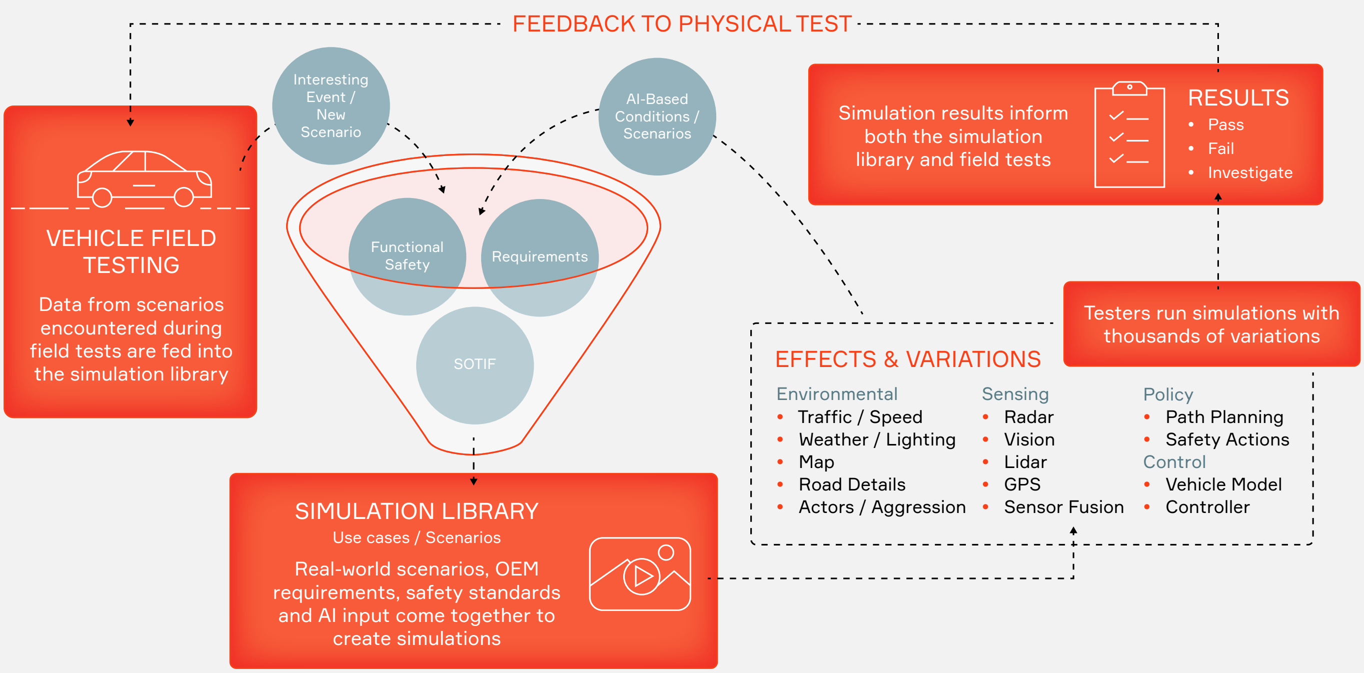
A cloud-based toolchain has the scalability to ensure that CI build times do not become a bottleneck — which is important, because bottlenecks can compress the time available for testing and can delay development.

With a cloud-based approach, development is frictionless because updates to code are shared across the development teams, so everyone is on the same page regarding the status of the project. This enables increased automation, remote troubleshooting, fast feedback, clear team assignments and clear product ownership.

Using cloud-based CI means that new code created by development teams is automatically integrated into the larger code base, and cloud-based CD — using over-the-air (OTA) updates — ensures that successfully built code automatically deploys into testing or production environments wherever they are. Done right, the cloud approach can also build in security, including disaster recovery, throughout the process. And centralized infrastructure hosted in the cloud can allow for a single management view into all of those highly secure and resilient CI chains.

CONTINUOUS TESTING

Continuous testing in the automotive world means creating a feedback loop between physical tests and software-based simulations.



In addition, the flexibility allows developers to refine the product later in the development cycle. Instead of deciding today what the consumer will want in a few years, developers can define requirements the year before, test against them and thus build systems that are closer to consumer expectations.

They can then deploy OTA software updates as needed for years to come, reducing warranty costs and ensuring the software continues to meet consumer expectations.

TESTING IS KEY

Testing is where the rubber really meets the road in the automotive industry. At $7 to $10 per mile, it is extremely expensive — not to mention time-consuming — to physically load new software into a vehicle and test-drive it for the potentially hundreds of thousands of miles needed to make sure the software works under all types of driving conditions.

Simulation programs enable companies writing software for the automotive industry to cut costs while delivering flexibility and repeatability. In addition, by going the simulation route, organizations can test pieces of code or components of a complex solution as they are being developed, rather than waiting for the entire product to be finished and then testing, going back and making fixes, then testing all over again.

With simulation technology, in combination with SIL or HIL, daily builds can be tested and validated. And multithreaded SIL or HIL enables multiple tests to be conducted at the same time rather than sequentially, which also saves time and improves efficiency.

SIL tests take place completely within a software-generated modeling environment. SIL testing has the advantage of not requiring special hardware (it can be conducted on virtually any laptop or other compute platform) and is most appropriate for testing a design in its early stages.

In HIL testing, the system simulates vehicle and environmental inputs for the ECU, causing it to believe that it is connected to a real vehicle. The HIL bench contains all of the relevant vehicle components. A simulator presents inputs to actual cameras and radar systems, which in turn send signals to the system under test to see whether it responds correctly to the inputs.

For example, test scripts can create a scenario in which a vehicle traveling at 60 mph around a curve in the rain encounters an unknown object in the road or an oncoming car swerving across the center line. Cameras and radars attached to the HIL test bench send images to the ECU, and the system under test has to process that data in real time and make a decision on what course of action to take.

Using these approaches, Aptiv testers can perform more than 1 million object-level simulations per day, “driving” 200,000 miles each day with realistic sensor simulations, including radar and camera inputs.

The simulations also allow testers to quickly test rare or potentially dangerous use cases. Normally, someone might have to drive hundreds or thousands of miles to attempt to reproduce a specific driving condition or component issue. Simulations allow testing on demand, with re-simulation of particularly challenging scenarios to show how different versions of the software react to the same inputs.

BENEFITS OF CLOUD-BASED CI/CD/CT

A global, integrated, cloud-native architecture based on open standards and agile software development principles eliminates traditional bottlenecks and delivers key benefits for companies developing advanced software for the automotive industry. These include:

- **Performance:** A cloud-based, globally available architecture can enable central control of test benches remotely from anywhere in the world. This increases scalability and flexibility.
- **Velocity:** Automation can slash build times by up to 80 percent. Companies can eliminate bottlenecks through a container-based architecture and code-merge debottlenecking approaches. They can rebuild and test only the portion that changed, resulting in 60 percent faster deployment.
- **Co-development:** When multiple partners can test against one another's code in real time, everyone gains the ability to find, fix and test integration issues on the same day a supplier commits its code. The result is fast, high-quality, low-risk delivery of complex software-enabled functionality for OEMs, plus cost-effective, simplified maintenance and enhancements in post-production.
- **Quality:** Full integration with remote SIL/HIL test benches, a traditional challenge for agile development in the automotive industry, improves product quality because developers can test anywhere and conduct unlimited tests in parallel.
- **Transparency:** The full team gains full visibility into the latest status of all software development chains, regardless of program size, complexity or location. CI best practices can be enforced and supported, and issues can be identified before quality is affected. This also greatly simplifies enforcement of security policies.

DATA MINE

Aptiv drives and records 400,000 to 1 million miles of data per customer project. This vast database of raw perception and scenario data includes radar and camera data from around the world in all environments, at various times of day and for a range of weather conditions. Product developers can analyze the log files to see how Aptiv sensors and controls and the OEM's vehicle interact as a whole system and then use that data to inform further development.

APTIV'S CI TOOLCHAIN

Aptiv uses a cloud-based scale-up/scale-down technology in its CI software development toolchain, allowing multiple tests to be conducted in parallel. This supports new ways of working that allow features to be defined much closer to the release date rather than when development first starts. And it enables deep collaboration among Aptiv, OEMs and third-party developers, who are all using the same toolsets and working from the same dashboards in real time.

The results have been impressive. Today, the platform builds and tests 10 million lines of code daily, which has resulted in breakthrough velocity and quality improvements. For example, the complete build time for an advanced safety system before applying Aptiv's toolchain was more than 12 hours, and while it was based on current industry-standard build technologies, it required that one entire build be completed before the next was started.

The platform reduces the build time by 70 percent, on average. This removal of the biggest bottleneck in automotive CI can enable a quantum leap in quality by allowing complete execution of a wider set of tests for every build. It helps enable a critical activity of continuous testing known as a code merge gate, which consists of a complete build followed by full execution of automated tests on every code check-in.

The exponential growth of the complexity of automotive solutions has outpaced the capability of legacy CI chains based on static infrastructure. Aptiv's CD platform solves these challenges by adopting a completely new, cloud-native approach that includes smart build and elastic infrastructure technologies. With these technologies, the platform can easily scale from tens to hundreds of concurrent builds, and it automatically provisions just the right amount of compute for optimal execution of code building and testing.

New CI chains can be deployed in minutes, without bottlenecks. A single management view enables visibility into all CI chains globally, with security and disaster recovery built in. Updates can be rolled out with a single click of a button, and enhancements are integrated with every sprint. And the toolchain offers full traceability of requirements and process compliance, including support for ASPICE and TISAX 3.

The result: reduced development costs, increased speed to market, increased flexibility, and true real-time co-development between Aptiv, OEMs and third parties.

THE POWER OF AUTOMATION

In the past, developers tested software manually. Software releases occurred monthly, and it took nearly that long to perform the testing. With the dramatic increase in lines of code associated with each build, it would have been impossible to continue with that approach.

As Aptiv gained experience in developing and testing major automotive software projects, we also gained a deep understanding of the challenges and the tools that are needed to solve them. Today, thanks to a CI/CT pipeline, 96 percent of the testing process is automated, and execution time is measured in days, even with many more requirements to test.

We continue to increase the level of automation to reduce the testing window for complete builds to a single workday of six to 10 hours.

ABOUT THE AUTHOR

Brian Murphy
Director, Engineering Services

Brian Murphy is passionate about using cutting-edge technologies and processes to drive quality and efficiency in product development. He has more than 25 years of experience in software development, architecture and management across the financial, health, telecommunications and automotive sectors. Building high-performing teams, delivering at the speed of business and innovating beyond expectations are three core values that Brian practices as standard work.

LEARN MORE AT APTIV.COM/SVA →