SIEMENS
Ingenuity for life

Siemens Digital Industries Software

Orchestrating automotive embedded application development

Application delivery and monitoring

Executive summary

Automotive software applications must be configured and calibrated for the specific features and hardware equipped on each vehicle variant, resulting in hundreds of possible variants. Further escalating the challenge, OEMs must track and maintain these applications after release to vehicle manufacturing, and even into the field. A unified platform for software application development coordination facilitates the delivery of accurate and high-quality applications across the entire spectrum of vehicle variants. Such a platform also supports monitoring these applications after delivery.

Piyush Karkare
Siemens Digital Industries Software

Introduction

Vehicle electrification, connectivity, automation, and shared mobility are driving a need for remarkably sophisticated software to enable features like advanced driver assistance systems (ADAS), battery management, vehicle-to-everything (V2X) communication, and more (figure 1). Each of these features requires a combination of complex software applications to operate, resulting in a massive increase in the complexity of in-vehicle software overall. Today, vehicle software content commonly exceeds 150 million lines of code.

In addition to growing complexity, the automotive mega-trends of autonomy, connectivity, electrification, and shared mobility (ACES) are causing a shift in software and electrical and electronic architectures. These architectures are becoming more centralized, with a greater number of vehicle functions managed by a smaller number of domain control units (DCUs) and a standardized base-layer of software. The creation of a standard software platform will help OEMs scale software features across their product portfolio, but also adds another layer of complexity to the development and delivery of vehicle software builds. Automotive software engineers are being asked to manage this complexity to deliver simple and intuitive user experiences for the end consumer.

Disruption in today's automotive market is focused on a reinvention of personal mobility independence, rather than a reinvention of transportation technologies. Consumers demand greater choice and freedom in their transportation options. Today, that translates to increased options for vehicle personalization. In the future, this will lead to the creation of responsive, flexible mobility systems.. Automotive OEMs are already expanding options for vehicle customization through extensive catalogs of optional features. As autonomous, connected, and shared mobility takes hold, manufacturers will need to produce a mix of vehicle configurations to meet consumer needs. Software configurations must be developed to support each of these vehicle builds. Software engineers must optimize software configurations to support the maximum number of vehicle builds with fewest number of applications and application configurations.

Most customer-facing features are electrical, electronic, and software related. These typically include advanced driver assistance systems (ADAS), upgraded infotainment systems, configurable interior ambient lighting, exterior lighting, and mobile device connectivity. These features can be configured for vehicle builds to maximize on volume margins. However, more fundamental vehicle systems such as engine control, battery management, torque management, braking, and steering predominantly employ software controls and electronics to transfer driver input into action. As a result, each vehicle configuration has different software needs that are influenced by the features equipped on the vehicle and the system architecture used to accomplish those features.

As software engineers wrestle with the sophistication

Figure 1: The megatrends of autonomy, connectivity, electrification, and shared mobility are increasing the importance of software and electronics in the automotive market.

demanded of individual applications, they must also manage the configuration and calibration of these applications, as well as that of entire software builds, to match each vehicle variant. Each software configuration also has to be deployed to the correct vehicle build with the correct software calibration meant for that variant.

Application configuration and calibration processes are also undergoing changes as software architectures centralize. As software untethers from hardware, the configuration of applications will depend on a combination of hardware and software platform constraints. The push for hardware consolidation shifts the complexity to the software side, with software functions delivering applications. These applications can be ported to multiple ECU abstractions and can be configured for specific personalized vehicle builds. With a layer of base-layer software or firmware between application and hardware, it can also become more difficult to track how and where applications are deployed throughout a vehicle platform or across the OEMs lineup.

But, configuring and deploying applications and software builds to each variant is not the only challenge. Latent bugs or issues may occur in field-operational vehicles, requiring the OEM to identify and resolve the issue as quickly as possible. With the increasing complexity of software applications and the need for multiple configurations, monitoring automotive embedded applications is another growing challenge. As issues occur, OEMs must be able to identify the potential impact of the defect across the vehicle variants as quickly as possible, and develop a solution to implement in affected variants. This is equally complex when it comes to over-the-air updates and vehicle assembly builds.

Key challenges in application delivery and monitoring

As these particular set of tasks deal with the final deliverables to the vehicle structure, they are arguably the most important (figure 2).

Key challenges

Common challenges that arise during this phase of embedded software development include:

- Last minute changes. Maintaining control over these changes, with complete traceability of the need for the change to and from the system-level is critical and often ignored. This eventually causes scope-creep, delays, and costly downstream rework as late changes may cause new defects that go undetected as teams scramble to deliver against vehicle milestone timelines
- Cross-domain changes. Last minute changes are not just a software issue; electrical systems, mechanical systems, and others frequently encounter design changes even after all the reviews and sign-offs are complete. Even frozen data, which is supposed to be fully committed, undergoes changes. Keeping a close eye on these cross-domain changes and updating for the software deliveries in concert is an intensely complicated task that increases the potential for additional risks considerably. All engineering teams need clear visibility of changes and change impacts, from the system level down, and back up
- Application configuration management. Vehicle platforms spawn dozens of discreet vehicle builds with a mix of shared and unique features, components, and embedded hardware. The software domain has to configure software builds to match each of these vehicle builds, rapidly increasing complexity. Software engineers must ensure that the software being delivered to each vehicle is not only complete from a functional perspective, but also fully compatible with the hardware in the vehicle build; it must deliver the correct configuration with the coinciding calibration and boot-loaders. This is especially important when updating customer vehicles in the field through OTA updates

Moreover, to ensure that compatible applications are delivered on time and at a high level of quality, engineers require a robust embedded application development platform that can coordinate the testing and quality assurance process from planning to testing and documentation; manage late-stage and cross-domain changes; and collate this data to supply the final vehicle BoM.

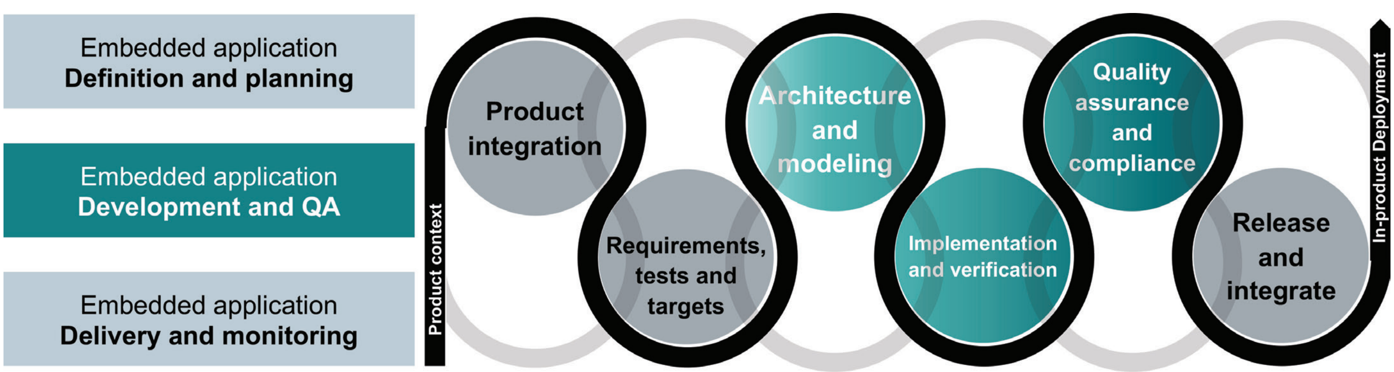

Figure 2: Application delivery and monitoring covers the implementation of changes from quality assurance, and the delivery and monitoring of final application binary to software builds and vehicles.

Embedded application delivery and monitoring with a unified platform

Modern embedded application engineering platform solutions coordinate the delivery of applications to the correct vehicle build bills-of-materials (BoMs; as-designed, to as-released, to as-built), and ensure engineers can monitor applications after release (as-built to as-serviced BoM). These solutions can coordinate the verification and validation of application builds in the context of the appropriate system or product variant constraints. This includes identifying the ecosystem of vehicle hardware and components created by each product variant and ensuring that applications are tested in all relevant hardware ecosystems.

This ensures that engineers can deploy fully verified and validated builds that meet all hardware and system constraints. This also reduces the cost of compliance and product liability by maintaining a complete record of verification activities.

Implementation and verification

During the application development and quality assurance processes, application code is continuously created, tested, updated, and tested again to verify that the application works as designed, and validate that the design is correct. In the context of the application delivery and monitoring activities, engineers are focused on maturing and committing code so that applications can be prepared for delivery to the vehicles (figure 3). Software testing transitions from primarily model-and software-in-the-loop (MiL/SiL) testing to hardware- and vehicle-in-the-loop (HiL/ViL), which verifies the application using virtual (HiL) and physical (ViL) hardware, and even the entire vehicle on real-world test tracks.

Engineers must constantly confirm that the implementation is consistent with the vehicle hardware and system constraints. Changes made in other related applications must also be considered. Maintaining alignment with system and hardware constraints and requirements is particularly challenging as code changes are identified and applied. An auditable change management process is critical to ensuring that changes are controlled and fully traceable to application constraints and requirements. Such a process must also support sufficient flexibility to enable code updates in response to MiL, SiL, HiL, and ViL testing.

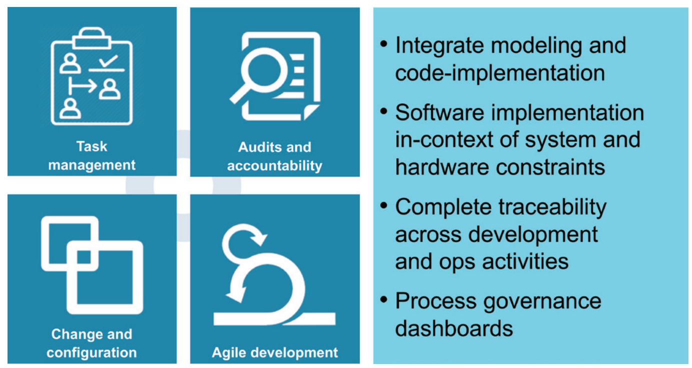

Figure 3: Engineers must continuously verify and validate application functionality as code matures.

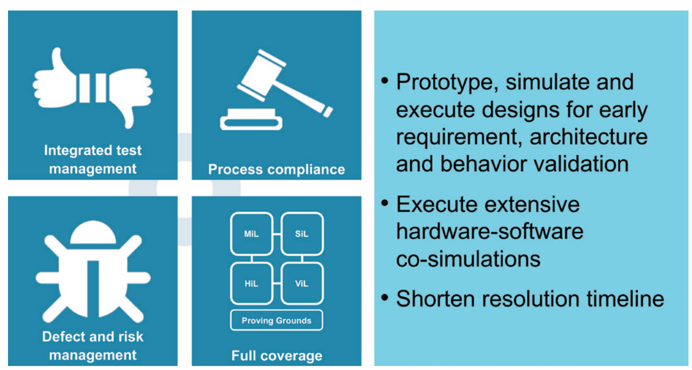

Figure 4: Quality assurance verifies and validates embedded applications in the context of system hardware.

Quality assurance

Quality assurance and compliance certification is typically a continuous process throughout development. The goal of quality assurance is to verify (working as designed) and validate (designed to work) applications in the context of system hardware (figure 4). Software engineers define test cases, test plans, and test vectors while design requirements, architectures, and models are created. The challenge for engineers is coordinating all of this data to the relevant work items at various levels of abstraction. Software must be tested at all levels of abstraction, from individual functions all the way up to vehicle-level features. Furthermore, as tests are run, the results must be associated to the relevant work item to show that the item is performing as needed, or that it is faulty and requires fixing.

However, as engineers develop and test software architectures, models, and actual code, they are constantly creating new data. For each round of testing, software engineers have to execute test plans that detail the test case, test execution strategy, test vectors, and other factors for each work item. As tests are completed, issues must be logged (potentially automatically or semi-automatically based on certain conditions) and resolved with test results linked back to the relevant work items. These data associations are crucially important to demonstrate test coverage and support complete traceability from requirements to implementation.

This data must be tracked and associated to the appropriate software build or configuration. Traditionally, engineers must manually maintain these complex data relationships, taking time and effort away from development and engineering work.

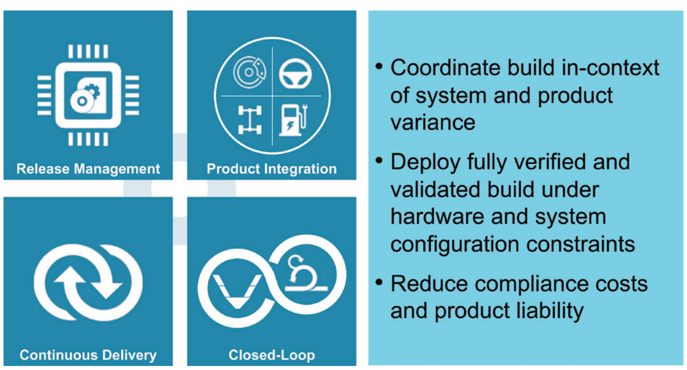

Figure 5: Application release and integration requires that application builds are aligned with specific product BoMs.

Application release and integration

To release applications, software engineers must coordinate application builds for their specific product BoMs (figure 5). Most importantly, each application build must be prepared in the context of the vehicle or system variant in which it will be deployed. To do this, engineers must ensure that the application is fully verified and validated under the specific hardware and system constraints of the vehicle variant. Frequent software testing and updates, as well as changes to the variant hardware or system configuration make this extremely challenging.

Upon release, each application must be traceable to the as-built and as-serviced BoMs of the vehicle on which it is installed. This traceability is key for tracking applications as they enter the field in customer-owned vehicles, ensuring that applications remain up-to-date over the lifetime of the vehicle.

Conclusion

Automotive application development is a multifaceted and challenging process from the initial definitions and planning stage right up to the final hurdle of delivering quality applications, in the right configuration, to the correct vehicle. This feat requires the collaboration of many cross-domain teams and stakeholders. A unified digital platform for automotive embedded application development facilitates collaboration and visibility across disparate domains, including software, hardware, and system engineering. Such a platform also enables companies to reuse proven software components accelerate the development of new and varied applications.

Throughout the application development, delivery, and monitoring, the embedded application development platform can connect with various tools to provide code performance coverage data, and to ensure alignment with methods and coding standards such as MISRA-C.

Comprehensive traceability provided by the software engineering platform enables a closed-loop system of monitoring and updates. As software engineers monitor the deployed software, traceability from the product or vehicle-level all the way to the concept-level requirements, specifications, and other data artifacts enables quick re-engineering response. This response is critical to resolving issues that come up in the field, improving consumer safety, and reducing or eliminating warrantee costs from faulty vehicles.

Cars are not only becoming more electrified, but also digitalized. Software is assuming control over more and more of the vehicle features that matter the most to consumers. As consumers look for increasingly connected and intelligent vehicles, OEMs must evolve their software development processes. A single platform to collect, collaborate, trace, and control the entire application development process will enable the innovation necessary to compete in this new market.

To read more about how Siemens solutions can help you, please visit siemens.com/aes where you will find additional content such as blogs, whitepapers, podcasts, product videos, webinars, solution capabilities and infographics.

Siemens Digital Industries Software

Headquarters
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

Americas
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

Europe
Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

Asia-Pacific
Unit 901-902, 9/F
Tower B, Manulife Financial Centre
223-231 Wai Yip Street, Kwun Tong
Kowloon, Hong Kong
+852 2230 3333

About Siemens Digital Industries Software

Siemens Digital Industries Software is driving transformation to enable a digital enterprise where engineering, manufacturing and electronics design meet tomorrow. Our solutions help companies of all sizes create and leverage digital twins that provide organizations with new insights, opportunities and levels of automation to drive innovation. For more information on Siemens Digital Industries Software products and services, visit siemens.com/software or follow us on LinkedIn, Twitter, Facebook and Instagram. Siemens Digital Industries Software – Where today meets tomorrow.

siemens.com/software

81257-C1 12/19 C