Accelerating Innovation in Embedded Systems by Adopting DevOps

CIMdata Commentary

Key takeaways:

- *Embedded software development is undergoing a revolution driven by exponential increases in complexity driven by rapid technology advancements.*
- *In today's smart, connected products, embedded systems and embedded software is where the proverbial rubber meets the road. Think autonomous vehicle development or innovations in health care—embedded software is where the magic takes place.*
- *Successful embedded system software development demands enabling faster time to market, enhanced productivity, and the agility to support continuous product evolution to enhance customer experience, all of which drive the need to embrace DevOps for embedded software.*
- *Moving from conventional embedded systems development models to agile models incorporating the best of DevOps is challenging. A modern DevOps solution specifically designed for the embedded system software development lifecycle (SDLC) can address the challenges and help accelerate product development and delivery.*
- *HCL's holistic end-to-end DevOps approach for embedded software development enabled by frameworks and solutions is helping their customers meet productivity, time to market, and complexity challenges.*

Introduction

For decades, embedded software has been used to integrate hardware capabilities into useful functions and features within a multitude of embedded system products. Over the last
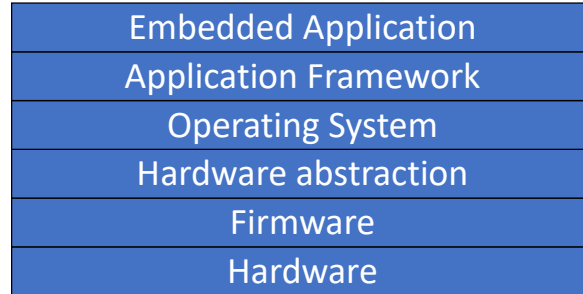
> **DevOps**
>
> Is a set of practices that combines software development (Dev) and IT operations (Ops). It aims to shorten the systems development life cycle and provide continuous delivery with high software quality. DevOps is complementary with Agile software development; several DevOps aspects came from the Agile methodology.
>
> *(Source: Wikipedia)*

twenty years or so, embedded software growth has been driven by the increasing incorporation of microprocessor-based controls into products. The adoption of Industry 4.0 is accelerating the trend. As the cost of integrated circuits, electronics, and sensors has fallen and their functionality increased, manufacturers have added many useful capabilities and features to their products. Over time, the technical building blocks have become standardized and organized into technology stacks such as the one shown in Figure 1, making advanced embedded system product capabilities easier to develop.[1]

---

[1] Research for this commentary was partially supported by HCL.

| Embedded Application |
| Application Framework |
| Operating System |
| Hardware abstraction |
| Firmware |
| Hardware |

Figure 1—A Typical Hardware/Software Technology Stack
(Courtesy of HCL)

The continuous expansion of software in automotive products is a good case study in the importance of embedded software and how product complexity has grown over the decades. In the 1970s and 1980s, electronics were incorporated to manage ignition systems and fuel injection in automobile engines. In 1980 microprocessors were added to improve combustion to meet increasingly stringent emission regulations. Over the years the sophistication of engine management grew as technology improved, and processor-based controllers were added to many other automotive systems such as HVAC, entertainment, braking systems, and stability control. While early controllers were primarily created using purpose-built electronics, the flexibility of software to program electronics drove the adoption of embedded software approaches. To support the connection of controller devices, networks, e.g., the Controller Area Network (CAN) bus, were created to enable communication between devices with embedded software. This enabled the linking of functions via software such as braking and engine management to create stability control features. Today, this is the approach being used within many industries; in which electronics and software are embedded into a wide array of products used by consumers, industry, and government.

The latest trends in the automotive industry include autonomous vehicles and electric propulsion, where embedded software capabilities play a major role. Within automotive the scope has grown and now interacts across all major vehicle systems, the operating environment, and soon infrastructure. Software needs to detect, analyze, and predict actions and execute commands in real time. Much of this processing needs to happen on the vehicle so safe and effective operation can be achieved.

Similar trends have emerged often based on Industry 4.0 in other industries including medical devices, avionics, infotainment, and consumer goods, where embedded system software development is a driving force. Increasing numbers of connected devices in regulated industries such as wearables and implantables and advancements in 5G and IOT have expanded the need for safe and high-quality embedded system products.

Challenges in the Embedded SDLC

Adding electronics and software to products has radically changed product development. Companies can develop product features driven by software that have virtually no additional manufacturing cost, thus improving profitability. This capability also enables manufacturers to create new, recurring revenue business models. The planning and upfront development required to incorporate electronics and software effectively is complex and often requires significant upfront investment. However, the resulting rewards of easier product differentiation, improved performance and safety, lower manufacturing cost, superior customer experience, higher profit margins, and new business models ensure the use of electronics and software will continue to grow.

Configuration management is a core capability of all product development domains. Ensuring the right version of items, whether mechanical, electronic, software, or even ingredients are correctly combined to create a product is critical to safe, effective, product operation. Within the embedded SDLC software not only needs to be managed relative to other software, but it must also be tightly coupled with associated hardware. Software updates, whether installed manually or over-the-air (OTA), add to the configuration management complexity since the version of software running on a product can have operational impacts on the product.

Electronic components used in circuits can have short lifecycles, requiring manufacturers to regularly update designs and related software. In the best-case, alternate components are directly compatible and software will continue to function unchanged. In the worst case, when an obsolete component doesn't have a direct replacement, a redesign is necessary, often requiring embedded software changes.

In today's fast-paced world, product development is concurrent. The end-state hardware environment is not always available early enough to test software so software must be developed using virtual hardware to simulate physical hardware. This approach is becoming more and more common, but requires a well-managed, sophisticated software development environment and developers must be confident that emulators simulate the hardware environment correctly.

Going from requirements to the software loaded on a device is a complex process even for a simple product. The software applications required to transform an idea into a product include requirements management, change management, model creation, integrated development environments, code configuration management, software build management, automated testing, bug tracking, software packaging, and distribution. Software distribution has changed dramatically in recent years, Historically the process was to download the firmware, software, or application onto the target device, but OTA updates are becoming more common. Embedded tool chains incorporate products from many independent software vendors (ISVs) and they are independently updated, so ensuring that the chain stays unbroken is a complex task.

Safety critical and regulated products have always had strong traceability requirements. Imagine software in a medical device such as an insulin pump. If the software fails, the patient could be injured or die. Software traceability is much more than understanding the flow of data and connectivity between source code modules. The relationships between requirements, code, hardware, and product features need to be understandable, traceable, and manageable. When something goes wrong, manufacturers and regulators need to understand what happened and why. But manufacturers also have an opportunity in that traceability also makes changes and reuse processes easier to execute. With proper traceability in place, impact analyses can be easily performed, enabling developers to identify impacted areas. For example, if a product manager wants to increase performance, developers can easily identify the hardware and software elements that impact performance requirements, assess the feasibly of what needs to change in the software, and ascertain the compatibility of the hardware that can support it.

DevOps in Embedded Software Development

The agile approach to software development has taken over the SDLC during the past twenty years. The most recent advancement in software development is known as DevOps. It is widely used in cloud-based infrastructure, software-as-a-service, and enterprise domain products. By tightly coupling development and operations, software developers can continuously improve

product and service offerings. By combining agile methods with automated testing and deployment, companies can regularly, and even continuously, update their solutions and monitor the performance of production instances. This closes the loop and feeds information back to development from operations for improvement.

While DevOps is common for ISVs moving to SaaS and website development, the application of it to embedded software development has been constrained by several gaps that needed to be addressed. Within a website or SaaS solution there is typically a single production version in use. Embedded software is usually installed on individual hardware instances that can have a long lifespan and there can be many variants of hardware to support. Source code needs to be developed, tested, and properly deployed to hardware variants out in the field.

Furthermore, deploying embedded software has undergone a revolution in recent years. Historically, deployment was done during manufacturing. The latest version was flashed (i.e., loaded into the electronics) and, since there was not too much interdependency between systems, testing was primarily focused on ensuring upward compatibility if an update was installed. With today's interdependencies, it's not only the variants of the hardware that need to be tested but the compatibility of versions across the network of hardware or components within the embedded network. Service personnel can re-flash software when products are serviced or distribute OTA via cellular connections.

Security and safety are critical elements that must be complete and correct. As embedded software must interact with the vagaries of the physical world, it needs to understand all the operating conditions and failure modes and address them properly. Hardware degradation and failures are a major concern, but even more critical are cyber-security concerns. Malicious software can intentionally damage products and cause harm ranging from equipment shutdown that disrupts customers to creating life threatening situations. An extreme example is safety testing for autonomous vehicles. The number of use cases that describe the possible operational scenarios is so large that they can't all be tested physically or in real-time. They need to be done virtually using automated testing software.

To optimize a product, all its elements need to work in concert. Often requirements can be satisfied using hardware or software, each approach with its own positive and negative effects. Using a co-design approach to managing the trade-off between decisions and execution and ensure the desired capabilities meet the requirements requires an integrated approach. To do this, the electronics and software product design teams must be properly integrated to enable smooth communication.

Achieving regulatory compliance and certification means "say what you do, do what you say, and prove it." Historically quality systems define this process using documents such as requirements, test plans, and test reports. With the complexity and volume of information related to modern products and production, these systems have become data driven. Products need to show compliance to requirements that can number in the thousands, and many are validated virtually rather than physically using automated testing. By capturing compliance information as data, it becomes easy to generate reports that prove compliance and show traceability. This approach also makes testing a proactive management tool rather than just a post-development reporting process.

HCL's Approach to Accelerate Embedded SDLC

HCL takes a holistic view of embedded software development. Their closed-loop approach spans continuous planning, development, integration, testing, monitoring, and feedback phases of a typical DevOps lifecycle. This process is a continuous flow that leverages the agile

software development methodology to ensure that the embedded software is kept up to date with the learning that happens as a project executes.

HCL's RAPID Embedded DevOps (RED) Framework helps define a method for incremental adoption of DevOps independent of the current maturity level. Within HCL's end-to-end framework, the key elements are strategize, incubate, expand, equilibrate, and soar. This framework defines practices and tools that complement and align with the SDLC of embedded software and system engineering processes. As companies mature, they adopt the processes and technologies to enable the elements shown in Figure 2.

STRATEGIZE	INCUBATE	EXPAND	EQUILIBRATE	SOAR
Assessment of current stage, gap analysis and identify baseline	Strongly Gated CI & Agile practices	Continuous Flashing and Continuous Testing	Measure and Remediate	Next Gen Embedded DevOps Practices
• CSDE Assessment • Tools Survey • Survey • CI Tools Selection • Agile Practices & Tools	• Socialize & Cultural change • Feature development • Unit test, Code Analysis, Code Review, SAST – Gates • Agile branching & Code merging strategy	• Identify chip vendor tools for firmware flashing • Continuous Testing Scripts • Testing tools like HOTE • One Touch Continuous Deployment Pipelines	• Standardize KPI & Performance Metrics • Track DevOps KPIs • DevOps Dashboard & reports to measure • Implement learnings	• Device Security, Performance • Release Experiments like A/B testing • Device Log Analytics • Device Lab as a service on private cloud

Figure 2—HCL's End-To-End RED Framework for Adoption of DevOps
(Courtesy of HCL)

The tools are structured in an end-to-end embedded SDLC platform known as Rapid Hadron which support the requirements defined by the RED framework shown in Figure 2. RAPID is a cloud agnostic self-service DevOps platform with built-in features such as SSO, SSL, anti-forgery, cyber security, data encryption, identity management, authorization, and cost governance. It has a family of orchestration engines to automate the whole journey of DevOps implementation defined in RED. Rapid Hadron abstracts the complexity of hardware, networking, and tools configuration and accelerates development via automation, low code orchestration of continuous integration-continuous deployment (CI-CD), and Infrastructure pipelines. CIMdata believes the comprehensive structure of Rapid Hadron lowers the barrier of entry for DevOps adoption.

The Rapid Hadron platform includes a variety of solution accelerators to support the embedded SDLC including testing, toolchain configuration, dashboards, and integration. Along with embedded software development, the platform aids automation of the FPGA compilation, verification and validation workflows to increase the process and product efficiency. The FPGA workflow stages of design, synthesis, implementation, and simulation using EDA tools can be built to enable FPGA continuous integration at module, sub-system, and system levels. CIMdata strongly believes in the platform approach as it enables customers to choose the tools that make sense for their business. That said, HCL also provides all the tools for end-to-end support reducing the complexity of supporting the RED environment.

Automated testing is critical in modern software development to ensure product quality and minimize quality regressions. The number of use cases has grown exponentially in complex systems so manual or physical testing is not possible. HCL provides two test automation frameworks (TAF) out-of-the-box; SMARTest and HCL One Test Embedded (HOTE) depending on the type of product.

HCL's XCITE is an accelerator to create and manage DevOps toolchains for embedded software products. It includes proven pre-built pipeline templates for the automotive,

aerospace, and medical device industries. Integrations with popular ALM solutions from IBM, Siemens, PTC, and others are included. HCL also includes predefined Pipeline-as-a-Code templates designed for engineering tools across the embedded SDLC such as MATLAB, dSPACE, Vector, NI, Helix, etc.

Dashboards provide easy access to information to support better decision making. eDASH is HCL's engineering dashboard accelerator for DevOps products. It provides enterprise-grade templates with role-wise views, roll-up reports and drill down capabilities. As with XCITE predefined templates provide OOTB tailored solutions for supported industry KPIs and widgets. HCL claims more than 120 KPIs and widgets are available in the library. Again, the power of the platform is applied to reporting engines. OOTB integrations are available for Grafana, Kibana, PowerBI, Tableau, and Qlik Sense.

High-quality, flexible integration capabilities are required to have a competitive platform. HCL's FUZE accelerator is an enterprise service bus (ESB) based framework built on a microservices architecture. It is designed specifically to cater to integrations with ALM, PLM, and Embedded Engineering SDLC Tools and supports OSLC and non-OSLC integrations.

- As technologies and processes change, these accelerators can be incorporated incrementally and dynamically into the framework, making development, support, operation, and management easier. HCL claims the following significant benefits from its platform-based accelerators:
    - RAPID platform enables 5-times acceleration in DevOps adoption and improves productivity of the Dev and Ops teams by 60 to 80%
    - XCITE can reduce DevOps configuration by 30% and reduce ALM and engineering Pipeline Workflow integrations by 60%.
    - eDASH can reduce dashboard implementation efforts by 70%.
    - FUZE can reduce ALM integration effort by 60% and the engineering tool integration effort by 25%.

CIMdata is a believer in the accelerator approach to solution deployment as it can shorten time to value, thus delivering a better ROI.

Success Story

HCL provided a customer success story describing how they helped an automotive industrial customer develop a DevOps pipeline to create continuous integration (CI) and continuous test (CT) workflows so that they could increase their operational efficiency. The end-to-end process used the RED framework and the related accelerators to create the workflow and leveraged HCL's TAF. Beyond the software, the solution also transferred artifacts to the appropriate repositories and incorporated the flashing process to ensure products received the correct software version. Consolidating the steps into a single integrated pipeline provided complete transparency and traceability increasing quality and productivity.

Conclusion

As product complexity and Industry 4.0 compliance increase more electronics and software get incorporated, companies are evolving their embedded software development processes. DevOps, a proven software development approach, is being adopted to address growing software complexity, quality, and time to market requirements. A modern DevOps paradigm, focused on cloud software development, must be adapted to support embedded software development. HCL developed their DevOps-based approach to embedded software

development based on their experience with customers across many industries. Their Rapid Hadron platform, along with XCITE, eDASH, FUZE, and SMARTest, and HOTE TAF tools supports the end-to-end requirements of embedded software development. These accelerators combined with HCL's domain expertise and system integration skills support a configurable approach to creating solutions that meet customer DevOps requirements now and into the future. Companies exploring improvements to their embedded software development processes should evaluate HCL's DevOps approach for Embedded Software Development platform, workflows, and integrated accelerators can help organizations realize significant improvements in their embedded SDLC.

About CIMdata

CIMdata, an independent worldwide firm, provides strategic management consulting to maximize an enterprise's ability to design, deliver, and support innovative products and services through the identification and implementation of appropriate digital initiatives. Since its founding nearly forty years ago, CIMdata has delivered world-class knowledge, expertise, and best-practice methods on a broad set of product lifecycle management (PLM) solutions and the digital transformation they enable. CIMdata also offers research, subscription services, publications, and education through certificate programs and international conferences. To learn more about CIMdata's services, visit our website at www.CIMdata.com or contact CIMdata at: CIMdata, Inc. | c/o Mr. Peter Bilello, CEO | 6863 Daly Road | Dexter, MI 48130, USA | Tel: +1 734.668.9922 | Fax: +1 734.668.1957; or at Oogststraat 20 | 6004 CV Weert | The Netherlands| Tel: +31 (0) 495.533.666.