Automotive Industry: Accelerating Time to Market with DevOps

CloudBees.

Contents

Introduction: Faster, Better, Cheaper

The automotive sector might seem to many to be a mature industry. Already more than a century old, the developed world has long been saturated with motor vehicles of all kinds and even under-developed countries are catching up. But this venerable industry has been going through rapid innovation, spurred by emerging technology and a host of regulations covering aspects such as improved car efficiency and reduced pollution, among others. Many of the changes in the automotive industry are speeding up because of the widespread incorporation of electronics and software into systems that had traditionally been purely mechanical.

Vehicles of the 1970s and 1980s, if they had onboard computers at all, may have operated with a few hundred lines of code. In contrast, today's vehicles are now powered by many millions of lines of code (MLOC), often hundreds of millions of lines of code, powering everything from braking to collision avoidance to body electronics that control every system human drivers interact with. Thanks to the combination of regulations and market demand, automakers must achieve better performance and deliver more features and functionality than ever before. In addition, automakers and parts suppliers must continuously improve on every aspect of their product in order to meet annual product cycles and market replacement needs. Software is the key to meeting the ever-increasing regulations, satisfying customer expectations and staying ahead of intense competition.

While other complex transportation equipment, like aircraft, might have development cycles and service periods measured in decades, the automotive sector measures both in much shorter increments. In addition, the automotive industry is competitive, and industry stalwarts are seeing competition from new, more nimble startups that have the luxury of greenfield development, with no legacy products, manufacturing processes or technology to contend with.

Automotive manufacturers and suppliers that can quickly deliver innovation via high quality software will win in the market with products that outperform the competition, function better and often cost less. Furthermore, updating, improving and repairing units in the field now often involves developing and/or deploying software updates - not fixing hardware malfunctions. Achieving the kind of breakneck development pace needed to support this new paradigm is anything but easy. There can be risks if the process is handled poorly. Automotive recalls and lawsuits are just as likely to stem from software problems as from defective mechanical components or mistakes in design.

Recognizing the importance of high quality software, standards have emerged in the automotive industry that suppliers and manufacturers adhere to. For example, the ISO/IEC 15504 standard for information technology – process assessment, also known as Automotive SPICE (Software Process Improvement and Capability dEtermination), defines a set of documents for the software development process. This is just one such process standard for the automotive industry. There are others, such as: Automotive Open System Architecture (AUTOSAR), IEC 61508, ISO 26262 and MISRA C. Compliance with multiple automotive standards is another complexity for companies in the automotive industry.

Software is nearly as critical to today's car as the propulsion system and wheels were a few years back. Software is now integral to functions as diverse as communication and engine and powertrain controls, safety systems, entertainment systems, navigation, operational control of lights and accessories, proper display of instruments, brakes, sensors and more. With the advent of self-driving vehicles, the importance of reliable software rises to an even higher level.

Power and Control

In recent years there have been many developments in the design of engines and power trains such as improved alloys, more highly engineered cylinder heads and efficient continuously variable transmission. All of these engineering improvements have been amplified by software which utilizes sensors and embedded processors to optimize operations reliably and in real time. Similarly, advanced electric and hybrid vehicles depend on sophisticated software to ensure smooth operation and optimal power management. In short, minus software, today's cars would become nearly as unreliable and "clunky" as cars of the 1970s. Given both consumer demand and the unrelenting flow of environmental and efficiency regulations, the pressure to do ever more within the narrowing limits of what is theoretically possible, will drive demand for more and ever better software.

Safety

Stopping a skid, preventing a rollover and anticipating and avoiding a collision are just a few of the capabilities now emerging as expected features in the automotive world. These, and many other capabilities involving the real-time integration of sensor data are the software-driven innovations that make cars safer, more reliable and more reassuring, no matter the driving conditions. The demand for these safety features will continue to grow, and what's more, consumers and regulators will expect better and faster performance. For example, just because a vehicle has an ABS system doesn't mean it's good enough any more – consumers will come to expect more advanced features, like active braking. Software will continue to raise the bar on safety, putting competitive pressure on vehicle manufacturers and their suppliers. And, consumers don't just want more capabilities and features, they want them faster and want them to be a standard offering in their vehicle of choice. If last year's top of the line model had important, technology-

driven safety features, why can't it be incorporated into the entire product line next year? In addition, if the features are software-based, why can't older models receive a software update to continually improve safety?

Communication and Entertainment

It's hard to believe that consumers once counted themselves fortunate to have features like an AM/FM radio or seat belts in their car. Today, integration with personal electronic devices, WiFi support for multiple devices, satellite radio, streaming content and a host of other bells and whistles come together to form in-vehicle infotainment (IVI) systems. A drive in a car is now expected to be a seamless part of a fully-connected day. The challenge for vehicle manufacturers and their partners is difficult. How do you make a car that is meant to last approximately ten years work with constantly changing and improving consumer technologies? In addition, add to the mix camera systems, black box data recorders and numerous subsystems and you have a massive web of software that needs to be able to adapt and perform reliability.

Integration and Security

Ultimately, everything in the vehicle must work together. It's a tall order. Underlying every device and system are many different control units and processors produced by different manufacturers. There are vast numbers of sensors and enormous quantities of code, most of which is siloed within specific functional areas. Much of the power of software comes from its ability to connect and coordinate – and that means that automakers have a huge integration challenge. Last but not least, hidden within that integration challenge is a growing requirement to ensure the security of all of these systems – something hardly anticipated to be a problem in the past. However with cars being more connected than ever, there are growing fears that cars might be hacked. The potentially grave ramifications of a hacked car are so great that manufacturers and partners cannot make security an afterthought. Security must be designed into automotive software systems from the onset.

Putting the Pedal to the Metal – The Role of DevOps in Speeding Deployment

For many years now, the automotive industry has been applying agile methods and practices in embedded software development. These development practices are foundational and helpful, but they are no longer enough. One of the keys to quickly creating reliable software that meets and exceeds customers' expectations is by automation - adopting continuous integration (CI), continuous delivery (CD) and a DevOps methodology. When an organization adopts DevOps, CD can become a regular occurrence. The automotive industry has been all about automation for some time. But even their software delivery processes can be accelerated by adoption of CD and a DevOps culture.

DevOps is a term that combines two terms: the "Dev" describes the development teams that code and test software, while the "Ops" part describes the operations teams associated with application delivery/deployment and underlying infrastructure. DevOps describes a style of software practice that is intended to eliminate older, slower and very siloed approaches to creating and deploying software. It uses automation and reusable techniques so that development and operations can focus on strategic elements of the process rather than routine tasks. In addition to promoting a

culture that supports more rapid development, DevOps also makes it possible to achieve more frequent application deployments; something particularly crucial in the automotive sector. These frequent updates are needed to support new or better functionality. In addition, software updates can help correct or avoid problems consumers experience in automobile operation, or improve performance or the driving experience for older vehicles.

A DevOps culture emphasizes more effective communication and collaboration between diverse teams such as development, operations and QA. As a result, IT teams deliver more reliable releases through improved productivity and efficiency. However, adoption requires an organizational culture shift. The cultural shift that is required for success is often the most substantial challenge involved in adoption.

Continuous Delivery as a DevOps Enabler

Continuous delivery is achieved through process refinement. Adding automation is a key enabler for achieving continuous delivery and, ultimately, DevOps. Together, continuous delivery and DevOps enable development teams to greatly accelerate delivery of software. With continuous delivery processes, teams can continually deliver secure and tested code that is in a production-ready state at all times. This includes the delivery of functional updates, the quick delivery of which can be critical to ensuring vehicle safety. Indeed, the potential ability to deliver updates remotely or over the air (think Tesla) to cars already on the road, is an exciting possibility that is emerging now.

DevOps centers around the collaborative effort required for rapid and frequent application development, testing and delivery. Because it's about culture, there is no single DevOps tool. Further, you don't just flip a switch and "do" DevOps. Getting to a DevOps culture requires a lot of hard work and effort. There must be support for it at all management levels.

Tools do not create DevOps, but successful DevOps teams need to have the right tools to enable continuous delivery, in support of DevOps. There are tools and technologies to enable and support transformation to a continuous delivery and DevOps culture. As the pressure continues to build for the delivery of new and improved automotive applications, a very pragmatic tools-based approach is required in order to avoid potential problems. Some of the common challenges that organizations encounter when implementing new tools include: high costs for support and training; risks related to the overwhelming task of managing updates and patches; and barriers to collaboration resulting from the use of disparate toolsets. To get around some of these challenges, many organizations have adopted open source tools to help with a variety of software delivery tasks across the entire development and delivery process.

Open Source and Its Role in Continuous Delivery

A growing number of organizations are building DevOps around an open source automation server called Jenkins®. Jenkins is already the de facto continuous delivery tool of choice. According to a survey conducted by ZeroTurnaround, 70% of developers use Jenkins for continuous delivery.

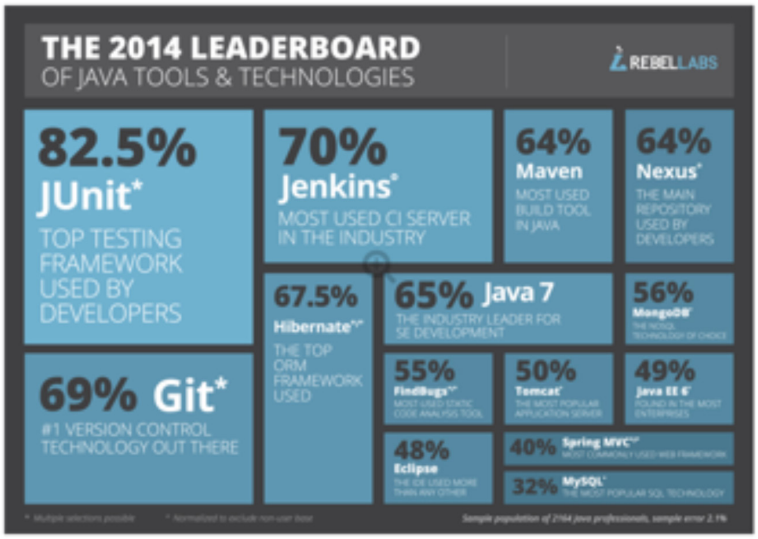

Figure 1: The Leaderboard of Java Tools and Technologies - from a survey conducted by ZeroTurnaround

One of the important benefits of open source projects is that they are supported by an active ecosystem of contributors; Jenkins has a very vibrant community of contributors and users. While open source has obvious appeals rooted in its sharing economy, which includes a low cost of ownership, many businesses have concerns about using open source tools without formal technical support. Other concerns related to open source software are in the areas of security, reliability and functionality. When open source software is built with the input of many contributors, testing may or may not have been rigorous enough to ensure reliability and security. Furthermore, if a problem arises during or after development that goes beyond the competence of your own team, who do you call?

To address these concerns, there is usually a commercial company behind major open source projects. This company typically provides formal technical support, verification and testing of the open source application and enhanced functionality to supplement the functionality that already exists within the open source software. Software companies have been launched to add an additional layer to open source software. They provide the safety and comfort of a traditional proprietary software supplier, while retaining the vibrancy of the open source community. In the case of Jenkins, CloudBees has commercialized the software, adding enterprise-level features and providing the formal technical support that organizations expect - even require. CloudBees CI adds additional enterprise functionality in the areas of security, scalability, manageability and resiliency. The Jenkins distribution it is based on is also verified through the CloudBees Assurance Program, ensuring a rock-solid, stable implementation of Jenkins.

Hurwitz & Associates recently completed a CloudBees-sponsored study of 150 IT decision makers. The survey delved into how they handle the challenges of software development and deployment. The respondents were from a range of industries, including the automotive industry. The survey participants were asked to identify their top reasons for deciding to purchase enterprise-supported versions of open source tools. The clear drivers for purchasing support for open source software were:

- Advanced enterprise features
- Professional support
- Improved ease of use

The Value of Jenkins

With Jenkins, organizations can accelerate the software delivery process through automation because Jenkins manages and controls software delivery processes throughout the entire lifecycle, including build, document, test, package, stage, deploy, static code analysis and many more.

Jenkins is sometimes referred to as an orchestrator of software delivery because it is a tool that can round up snippets of code from various modules, and get that code to where it belongs – much like an orchestra director conducting musicians to work together in the delivery of music.

Jenkins is already the de facto continuous delivery tool of choice for many DevOps organizations and is used pervasively within the automotive industry. There are a few reasons for this popularity. As noted, one factor is that Jenkins is open source, freely available, widely used and has the support of an active community. Jenkins also has an extensible architecture, giving it the ability to integrate with a wide array of code from disparate tools, devices and sources.

In short, the increasingly software-centric automotive sector needs and expects much from its software delivery teams. Coding efforts are often massive. They nearly always have actual or potential operational safety issues and as such must conform to specific practices.

Continuous Delivery and the Automotive Industry – Success at the Enterprise Level

The automotive sector is poised to lead the digital revolution. But, success in leveraging the benefits of continuous delivery depends upon several best practices. These best practices include:

- **Ensure a strong, ongoing commitment to DevOps.** Development and operations teams need to ensure a culture of collaboration, teamwork and shared goals throughout the software delivery process. In a DevOps culture, there is no finger pointing when things go wrong, but a spirit of teamwork and collaboration to troubleshoot and resolve, with lessons learned and procedures revised to prevent the same issue from happening next time.
- **Extend meaningful collaboration to include line-of-business executives.** As we've seen, the demand for speed, highly sophisticated functionality and the need for continual deployment, puts tremendous pressure on development teams.
- **Use tools that are up to the challenge.** This means teams must take a pragmatic approach to adopting tools that can support the scalability, manageability, high availability and support that enterprises require.

Automotive Industry Trends - The Road Ahead

There is no doubt that the automotive industry is in the midst of one of the greatest upheavals in its history - led by software. On top of relentless global competition, the industry is striving to meet incredible demands for better performance, innovation, efficiency and safety. Just in Time (JIT) manufacturing, Kanban and the relentless application

of automation have transformed the physical production of vehicles already. Today automotive companies and their suppliers are grasping for faster, more scalable and more reliable software. To succeed, software delivery teams must have the tools to enable them to create predictable and secure solutions. Much rides on the ability to provide new applications and updates faster; at the same time, new practices, methodologies and tools can be applied to meet the challenges. The most successful companies will be those that bring together software (and business teams) to work together in accelerating innovation.

DevOps is a movement which promotes a series of processes and methods for rapid and frequent application development; continuous delivery is a process which enables the delivery of code and fixes. With Jenkins, we have a tool that enables the movement and facilitates the process.

CloudBees Drives Enterprise DevOps and Continuous Delivery

CloudBees is the hub of DevOps, providing companies of all sizes with smarter automation solutions and actionable insights for accelerating software delivery. Our continuous delivery solutions offer DevOps teams the industry's broadest suite of products, from on-premise to cloud native, from self-managed to self-service, from guided best practices to flexible choice. By making the software delivery process more productive, manageable and hassle-free, CloudBees puts companies on the fastest path to transforming great ideas into great software and returning value to the business more quickly.

Learn More


CloudBees. Case Study

BOSCH

Bosch Drives Development with Continuous Integration and Continuous Delivery

95% Build Pipeline Automation

Read the case study
Bosch

Download the Forrester Whitepaper:
Open Source is Essential for Modern Application Delivery

Read the Whitepaper
The Business Value of Continuous Delivery

Learn more

www.cloudbees.com/products

CloudBees CI is built on top of Jenkins, an independent community project. Read more about Jenkins at: www.cloudbees.com/jenkins/about

0820v03

CloudBees, Inc.
4 North Second Street | Suite 1270
San Jose, CA 95113
United States
www.cloudbees.com
info@cloudbees.com