

Edmunds.com Case Study: Perforce as a Hub for Extreme Continuous Delivery

Company

Edmunds.com is one of the world's most popular online resources for automobile purchasing. It connects dealers and consumers, providing more than eighteen million unique visitors each month with research and assistance supporting new and used vehicle purchases. For dealers, edmunds.com provides broad exposure to prospective buyers. For consumers, it delivers in-depth information on product availability along with up-front, "no haggle" pricing and a personal dealer contact to facilitate the buying process.

Interviewee

Ajit Zadgaonkar, Senior Director of Engineering, is a core member of the Delivery Systems team for edmunds.com.

Online Buying and the Auto Industry

For most consumers, automobiles are their second-most expensive asset—after their homes. Today's smartest consumers spend a significant amount of time "doing their homework" before actually writing checks. The Internet enables them to compare products, models and prices, and increasingly, to research the opinions and experiences of other consumers as well.

With more than 200 million monthly page views, the design and performance of the edmunds.com site can dramatically impact both dealer revenue and customer satisfaction. The site must be constantly updated to meet or exceed expectations as customer and dealer needs evolve. Models and options are constantly changing. Pricing is variable based on geographic, macroeconomic and seasonal factors. New dealers come on board and existing dealers add products. Each of these factors, along with changing sales and marketing initiatives, must be reflected in the site content.

This volatility has challenged edmunds.com to become increasingly more agile in keeping site content topical and up-to-date. Like many companies, it was pressured by the need to find better ways to manage the creation and ongoing administration of software assets. This was compounded by the need to accelerate software testing and deployment as well.

At the same time, accelerated software delivery can be a risky proposition. In the case of edmunds.com, many post-development tasks—such as Quality Assurance (QA) testing and software deployment—were being done manually. Manual processes are error-prone, and software configuration or deployment errors can have adverse impact on production users.

For edmunds.com, this meant that new code releases could be held for months—as Operations personnel waited for the "right" time to deploy them. As with many other companies, there was a fear factor associated with potential adverse production impact—and this was inhibiting business agility.

For edmunds.com, the Perforce platform has become the foundation for an accelerated and automated approach to Continuous Delivery.

Using Perforce as Part of “Extreme Continuous Delivery”

Perforce bills itself as a pioneer of “Enterprise Version Management.” However for edmunds.com, the Perforce platform has become the foundation for an accelerated and automated approach to Continuous Delivery.

The platform is used by multiple stakeholders across the company. Publishers and editors use it to version and store articles, for example. However, its primary use case is for development-related assets, such as code, testing artifacts, and deployment scripts. According to Ajit, “Everything is code, and all code-related assets are centrally managed in Perforce. This includes Java and Shell scripts, front-end HTML code, digital metadata of media assets, Chef-based configurations, automated test scripts, and database scripts and procedures—all are stored in the Perforce repository.”

Ajit’s comments relating to his company’s use of Perforce are straight and to the point. “You deploy it and forget it, and it just works the way you expect it to. It doesn’t crash, it offers easy administration, and it scales. We don’t have to worry about our growing volume of code, because we know Perforce can handle it.”

In addition, Perforce’s versatility is the foundation for edmunds.com’s “extreme” approach to Continuous Delivery—software is being deployed to production as part of the “checkin” work stream in an automated fashion. Ajit and his team have automated the software delivery pipeline with a “watcher,” a small software module which automatically executes when engineers check code into Perforce. Once code is checked into the repository, multiple steps across a variety of build and staging platforms take place. At a high level, these steps include the following:

- Code is automatically compiled and a basic unit test is kicked off.
- If the software passes unit testing, it is promoted to an integration testing environment, which is a scaled replica of production.
- Integration and load tests are run.
- If the software passes all tests, it is automatically deployed to the production environment.

Outcomes

Most companies that have implemented agile methodologies encounter difficulties with code management, version control, software asset orchestration, and deployment to production. In the past, this was the case with edmunds.com as well. However, working with Perforce and other leading vendors, Ajit and his team have been able to automate the full delivery chain by centralizing code management, adding automation scripts, and synchronizing software asset checkins with automated testing and deployment. In doing so, the team has gained a high level of confidence that software will be deployed correctly and with minimal risk of adverse production impact.

This “hands off” approach highlights the level of confidence that edmunds.com now has in its software practices. Virtually every step in the process is “hands free,” and edmunds.com is one of only a small number of companies that has automated the entire software delivery chain from code development to production deployment.

“You deploy it and forget it, and it just works the way you expect it to. It doesn’t crash, it offers easy administration, and it scales. We don’t have to worry about our growing volume of code, because we know Perforce can handle it.”

Ajit and his team have been able to automate the full delivery chain by centralizing code management, adding automation scripts, and synchronizing software asset checkins with automated testing and deployment.

Ajit states, “We have modeled our software delivery process based on the way the business wants to operate. Communicating with consumers continuously is a business decision. We want to reach out to consumers super fast and don’t want shelf life for features and fixes. We want to be very nimble and very agile—and if the software is ready, we want to deliver it to production.”

Edmunds.com has taken a “time is of the essence” approach to software delivery. To support this “high risk, high reward” model, an “all hands on deck” culture quickly resolves issues if and when they occur. While this may not be the best approach for every company, it is working very well at edmunds.com.

Virtually every step in the process is “hands free,” and edmunds.com is one of only a small number of companies that has automated the entire software delivery chain from code development to production deployment.

Quote

“If we make a mistake, we impact our own customers’ wired or mobile access, and more than 150 API-connected partners. When you become the central hub, you impact everyone. The day you realize this is the role you play is the day you become more nimble in supporting all users.”

About EMA

Founded in 1996, Enterprise Management Associates (EMA) is a leading industry analyst firm that provides deep insight across the full spectrum of IT and data management technologies. EMA analysts leverage a unique combination of practical experience, insight into industry best practices, and in-depth knowledge of current and planned vendor solutions to help its clients achieve their goals. Learn more about EMA research, analysis, and consulting services for enterprise line of business users, IT professionals and IT vendors at www.enterprisemanagement.com or blogs.enterprisemanagement.com. You can also follow EMA on [Twitter](#) or [Facebook](#). 2832.012314