



CHALLENGE

Shorten delivery times for more than 180 applications that power 13 different Web sites

SOLUTION

Refine software delivery processes and implement open source Jenkins and CloudBees solutions for continuous delivery to automate tests, ensure reliable builds and increase build consistency across the organization

RESULTS

- >> Release cycles cut by more than 75%
- >> Teams focused on high-value tasks
- >> User experience enhanced through increased multivariate testing

PRODUCTS USED

- >> CloudBees Jenkins Enterprise

Orbitz Moves to Continuous Delivery

Orbitz Worldwide (NYSE: OWW) is a leading global online travel company using technology to transform the way consumers around the world plan and purchase travel. Orbitz Worldwide sells more than \$12 billion in travel annually and operates the consumer travel planning sites [Orbitz](#), [ebookers](#), [HotelClub](#) and [CheapTickets](#). Also within the Orbitz Worldwide family, [Orbitz Partner Network](#) delivers private label travel technology solutions to a broad range of partners including some of the world's largest airlines, bank loyalty programs and travel agencies and [Orbitz for Business](#) delivers managed travel solutions for companies of all sizes. The company's business platforms are supported by more than 180 different applications that work together to serve Orbitz customers and business partners.

With more than 350 developers on 35 different teams in multiple locations around the world, the Orbitz development organization has relied on continuous integration (CI) with open source Jenkins to build software for several years. Recently, Orbitz began using CloudBees Jenkins Enterprise as part of a move towards continuous delivery (CD).

"This year, one of our technical focuses is on delivering not only consistently and reliably, but also much faster. By making our process more efficient, and adopting continuous integration and continuous delivery practices with open source Jenkins and CloudBees Jenkins Enterprise, we have cut our release times in half, and then cut them in half again," says Jacob Tomaw, principal engineer at Orbitz. "Additionally, we now ensure that each one of the artifacts we build for our applications is produced consistently and in a repeatable way — a capability that is essential for continuous delivery."

Challenge: Shorten Release Cycles, Deliver Innovation More Quickly

Over time, the testing process at Orbitz had become burdensome. A problem noticed in production would lead to additional and sometimes redundant tests that had to be run for each subsequent release. "Our feedback loop on innovation was about 18 days, and most of that was consumed by manual tests and other checks that we had put in place in reaction to previous incidents," says Tomaw. Because two weeks or more passed before each new release, developers who wanted to get new functionality into production more quickly would frequently request patches, a practice that was both costly and inefficient.

Orbitz viewed the move to CD as a way to shorten release cycles and get new features into production faster. That move, however, would require a shift in the company's use of Jenkins. "We wanted to standardize on a single infrastructure, not just for continuous integration, but for releasing our applications. We no longer wanted to hear 'But it works on my machine' when a test failed," says Tomaw. "We needed high availability, more automated testing, consistency in our builds and an easy way to organize our jobs."

Solution: Enable Shorter Software Delivery Cycles with CI and CD

Tomaw and his colleagues began the change in culture required for CD by documenting existing processes from development through to production and identifying areas of potential improvement such as the lengthy test times and frequent patches. The documentation was part of a consistent effort throughout the process to provide visibility and transparency to all stakeholders, enabling the team to understand how they would meet key milestones, which included halving testing times, releasing once per week, halving testing times again and releasing twice per week.

As part of the move to continuous delivery, Orbitz added CloudBees Jenkins Enterprise to the environment to provide a common place for continuous integration and the release of software to production. These steps were vital to shorten delivery cycles and to support the move to CD.

The Orbitz development team began using the [Role-based Access Control](#) plugin and the [Folders](#) plugin from CloudBees Jenkins Enterprise to secure access to jobs organized by domain (such as hotel, air and car) and by tier (such as web and middle tier).

“At Orbitz, we are encouraged to ‘Mind the Store’ – to do what is right in our jobs and deliver as best we can for our customers. From that top-down vision, the goal of continuous delivery started as a bottom-up initiative from developers who knew it was the right way to go. With Jenkins and CloudBees, we have achieved several milestones on our way to that goal that have already enabled us to innovate faster.”

>> *Jacob Tomaw, Orbitz*

Learn More About Orbitz:

www.orbitz.com

CloudBees, Inc.

289 South San Antonio Rd
Suite 200
Los Altos, CA 94022
United States

www.cloudbees.com
info@cloudbees.com



CloudBees
The Enterprise Jenkins Company

To ensure that work could continue in the event of a Jenkins master failure, Orbitz deployed the [High-Availability plugin](#). After they deployed the [Templates plugin](#), the development tools team provided all development teams with standardized job templates to promote consistency across the organization.

“The Role-Based Access Control plugin and Templates plugin have freed our development tools team to focus on ensuring that our development environment is working as it should, instead of having to treat every build as a unique snowflake,” says Tomaw.

When a developer commits code to the Orbitz Git repository, a Jenkins job runs a build, unit tests, and a variety of automated UI tests, many of which were performed manually previously. Jenkins also creates a snapshot of the latest artifacts, and every twelve hours these artifacts are automatically deployed to the pre-production quality assurance environment.

Availability tests and other Apache JMeter tests are run in the pre-production environment continuously. Twice a week, Jenkins creates a new incremental release, sends a company-wide email notification of the pending release and then deploys the release to the quality assurance environment, where it undergoes further testing. Finally, the release is deployed to a canary release environment, and then to the entire production environment.

In the past, patches were the only way to get code into production outside of the 2-week release process. Today, significantly fewer patch requests are made and most of them are business-driven, originating from product owners rather than from developers. With faster cycle times, Orbitz instituted a new policy that requires each patch to be evaluated for its contribution to margin. In order for a patch to be approved, it has to contribute projected revenue at least equal to the labor expense of deploying it – otherwise it must wait for the next release, which is now at most four days away instead of 18.

Results

>> **Release cycles cut by more than 75%.**

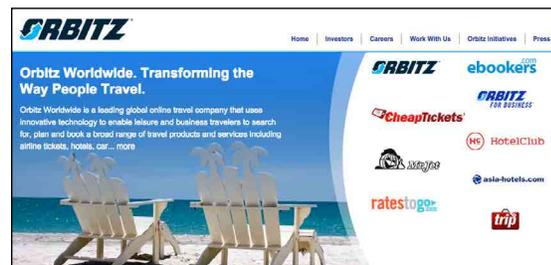
“In the past it took 18 days to test and deploy a new feature or innovation,” says Tomaw. “With Jenkins, CloudBees and our new processes, we streamlined and automated our testing, and gradually increased the frequency of our releases to once per week and then twice per week as we move to continuous delivery.”

>> **Teams focused on high-value tasks.**

“We’ve used the Templates plugin to make our builds much more consistent. That standardization has enabled our development tools team to focus on increasing reliability and efficiency instead of the quirks of every team’s unique build,” says Tomaw. “Similarly, automated testing has freed our testers to focus on higher-value exploratory testing, instead of rote manual tests.”

>> **User experience enhanced through increased multivariate testing.**

“When we do multivariate testing we provide different experiences for our customers. Before we refined our software delivery processes and started using CloudBees Jenkins Enterprise, it would take more than two weeks to cook test results into the main Web site and move onto the next level of testing,” says Tomaw. “Now, we can push updates made based on the test results into production within a few days. We’re able to perform more frequent multivariate testing to find what works best for our customers.”



CloudBees Jenkins Enterprise and CloudBees Jenkins Operations Center are built on top of open source Jenkins, an independent community project. Read more at: www.cloudbees.com/jenkins/about

© 2015 CloudBees, Inc. CloudBees is a registered trademark and CloudBees Jenkins Enterprise, CloudBees Jenkins Operations Center and DEV@cloud are trademarks of CloudBees, Inc. Other product or brand names may be trademarks or registered trademarks of their respective holders. 0115v00