Websym

Case Study

The company is one of the leading digital video entertainment companies that runs a number of YouTube channels. They have their own asset management and processing platform which serves as the technology backend for manging the lifecycle of all digital assets.

Product Characteristics

- An n-tier application hosted on AWS.
- Platform has a set of micro services developed in Node JS.
- Mongo and RDS used as asset and metadata repositories for the platform.

Development & Deployment Environment

- Atlassian Bitbucket used as the code repository.
- JIRA used as the ALM tool.
- Dev environments were local to developers.
- Test, Staging and Production environments hosted on AWS

	Identified Issues	Root Cause	Solution	Impact
SCCM	• Low confidence in code fixes being applied for bug resolution. • Error prone and time consuming platform upgrades.	• Code reviews not enforced. • No branching strategy being followed. • Code check-ins not tagged to work items/bugs.	• Code analysis using SonarQube configured to validate code quality during compilation. • Pre-commit code reviews enforced using ReviewBoard to ensure code quality. • GitFlow branching strategy implemented in BitBucket to streamline merge paths and enable release of patches. • Smart commits configured in JIRA and Bit-bucket for maintaining traceability of code changes.	• Code merging time reduced by more than 70% • Issues related to incorrect code check-in's reduced by more than 80% • Substantial improvement is code quality as measured through code analysis metrics • Automated creation of Mongo and AWS RDS update scripts has significantly reduced deployment errors related to database upgrades.
Build and Release	• Deployment of builds took 5-6 hours. • Setting up multiple parallel production environments was time consuming and error prone. • Test and Staging environments were static in nature resulting in increased costs.	• No CI pipeline in place. • No centralized repository for storing build packages. Build packages stored in the file system. • No governance enforced on deployment of builds. • Build promotion between Testing, Staging and various Production environments was done by recompiling source code.	• CI pipeline configured to generate scheduled builds using Jenkins. • Distelli configured to manage various deployment environments and to enforce release governance and automate roll-backs in case of deployment errors. • Chef used for infrastructure automation. Test and Staging environments created on the fly and decommissioned automatically as soon as the test cycles are completed. • Build promotion between Testing, Staging and various Production environments implemented using packages stored in the Artifactory package repository.	• Release cadence reduced to 2 weeks from 4 weeks. • AWS costs reduced by more than 70%. • Environment setup time reduced to 1 hour from 1.5 days • Build creation time reduced by more than 90%. • Upgrade errors reduced by more than 30%.
Monitoring	• Slow response to application failure incidents. • Root cause analysis of incidents took a long time.	• No monitoring of application parameters in Production environments.	• New Relic and AWS CloudWatch used to capture and display environment and appli-cation performance metrics via custom dash-boards. • Alert policies configured to inform relevant users about potential application failure risks.	• Application failure rate reduced by 25% through risk mitigation done via proac-tive monitoring. • Incident resolution time reduced by more than 30% due to availability of richer application logs.

Websym Solutions Pvt. Ltd.
Plot no. 34/2, Rajiv Gandhi Infotech Park – Phase 1,
Hinjewadi, Pune 411 057 – India

Phone: +91 (20) 661 43 400
Fax: +91 (20) 661 43 500
Email: sales@websym.com