

CASE STUDY

Nikon Leverages SCM Automation for Global Development

Sponsored by: Perforce

Melinda-Carol Ballou

January 2010

IDC OPINION

Managing and coordinating distributed corporate and IT resources helps organizations achieve efficiencies of scale for code management. It can help drive business agility and collaboration that limit waste. IDC research indicates that savvy companies increasingly are coordinating software change control, configuration, and version management with build management and deployment. This can aid communication and cut expenses, given the increased complexity of geographically dispersed software development projects.

As organizations implement process change for consistent code management and performance across dispersed teams, many are evaluating and adopting automated software change and configuration management (SCM) and version control solutions. IDC analysts encourage users to perform a maturity assessment and gap analysis — as part of the evaluation process — to determine their current state of organizational and process maturity before and after implementation to assess the impact of these efforts. This approach can facilitate adoption by teams that might otherwise be reluctant to change behavior. Clear evidence indicates change management can save costs and time, and this evidence helps advocates in their effort to combat the often encountered human resistance to change.

The adoption of appropriate automated tools for change management, version control, and build management, coupled with organizational and process improvements, is a foundation for success. Tool implementations can help focus teams with the collaboration required for process evolution. And automation, once accomplished, can facilitate visibility and management of incoming demand, changes, build management, and provisioning across geographically dispersed resources.

IN THIS CASE STUDY

Organizations using broadly distributed teams and complex sourcing for software development projects are increasingly bringing in consistent automation for software change and configuration management and version control. Back in the late 1990s, Nikon evaluated and installed Perforce's solutions as a platform for communication and collaboration across distributed development teams in North America and Asia. Benefits included increased visibility and improved performance and execution times, which resulted in qualitative development improvements and cost benefit returns (resulting from better performance and increased developer efficiency). This case study assesses Nikon's adoption of Perforce's SCM products and the company's organizational and process evolution to a successful deployment for IT and emerging support for the business.

SITUATION OVERVIEW

Organization Overview

Nikon is well-known for manufacturing cameras, lenses, microscopes, and other related optical and imaging products. Nikon has around 500 U.S. employees with 22,000+ staff worldwide. The particular division on which this study focuses supports Nikon's imaging software, cameras and accessories, and related businesses. The teams in this division include a small number of IT engineers who lead software development in the United States in conjunction with distributed developers in Japan who also package releases for teams in Germany. A total of around 61 users in New York and Tokyo must be able to manage code across their geographically dispersed environment to help drive Nikon's businesses in those areas. These teams needed a software change management and version control product that could be easily distributed across North America and Asia and that could perform and scale well. Currently, Nikon is seeking to evolve this solution to be used by nontechnical, non-IT users as well as by IT development teams to be able to more easily track and find application changes related to dynamically changing business needs and specifications (along with the demands of collaborative software creation).

Challenges and Solution

The main challenges faced by the Nikon teams when they began their evaluation nine years ago included poor global performance of their existing software version control system, lack of visibility into the work being done by other team members, and the need to backtrack to find out where problems had occurred in order to fix them. With distributed development staff in New York and Tokyo, these inefficiencies cost the teams time and money. In terms of the performance problems, the Nikon teams were using Microsoft Visual SourceSafe over a very slow ISDN connection, which created significant issues in terms of managing their version control systems collaboratively. Lack of management and visibility led to a chaotic development process when they created code. This approach created delays when the teams needed to backtrack to see exactly what problems had occurred and how to remedy the resulting issues. These problems cost the teams additional development effort. In order to be able to collaborate and to address some of these problems at that time, the teams would periodically take code and burn it onto CDs to share release information. This approach was inefficient and cumbersome and hindered fast and effective communication across these geographically dispersed groups.

To respond to these collaboration, expense, and code management issues, the Senior Manager of Software Development at Nikon evaluated Perforce, CVS, and Voodoo (a Macintosh-specific system). The criteria that were of focus during the evaluation included the speed and performance of the solution, product architecture, integration with the company's development tools, usability, and cost.

Nikon rejected the other products at that time for a variety of reasons — Voodoo was platform dependent, SourceSafe's poor distributed performance pushed it out of the equation, and CVS seemed both not as functionally strong and more complex with poor management capabilities.

Nikon opted for Perforce because of its strong architecture, performance speed, user references, good reviews, and reasonable cost. Nikon set up a test system with Perforce, and within a week, the solution was deployed.

The time frames involved for choosing Perforce (from evaluation to deployment) went quickly for Nikon — Perforce was efficient in responding to Nikon in terms of licensing, and the agreement was completed within a matter of two to three weeks.

The initial deployment itself also went quickly — it took about a week to get the solution up and running and to bring over Nikon's source code into Perforce. The product implementation involved only one person from Nikon (with no outside integrator support). One of the benefits of Perforce described by Nikon is ease of implementation.

While the deployment was straightforward and went according to plan, Nikon's hardest challenge was getting the team to use the tool, given that process and cultural change do not happen easily. Users initially resisted the new tool; lack of certainty about benefits slowed adoption. The Senior Manager of Software Development at Nikon leveraged a variety of approaches to gain buy-in, including focused marketing efforts, such as educational sessions for the new users about what the tool would provide, along with quick demonstrations, as well as having key developers reiterate the need and the costs of the preexisting approach. These efforts ultimately won over the broader Nikon team to consistent use and adoption of Perforce.

The Perforce server resides in New York, and the development teams from Tokyo connect via the WAN to the Perforce server. These teams use the product for delivering both daily and longer-term software releases as frequently as two to three times per month. They also perform automated "smoke tests" in conjunction with Perforce — doing whatever tests they need to make certain that the release is usable and functions appropriately. Nikon has created a number of automated scripts that connect to Perforce; pull down the latest release; and test, package, and automatically deliver the release packages to specified locations. If a full application is being delivered, then the process is more manual, and more work is done in terms of assembling the package. Small components like a library are normally handled in an automated way.

Results

In the past nine years, the Nikon teams have committed themselves to Perforce and are pleased with the resulting time savings and collaboration benefits. Perforce forms the "backbone" of development for this division at Nikon, which is looking to integrate other services with Perforce. They currently have integrated wikis with Perforce to facilitate cooperation and communication across groups. They are also trying to determine the best way to build their service and configuration management system around Perforce to coordinate source code management and issue tracking for both technical and non-IT users.

In the future, for non-IT users, Nikon would potentially like to use Perforce to handle issue and change management for areas outside development. This would enable business users to create product and business specifications in a wiki or a Microsoft Word document that is then delivered automatically to and managed by Perforce. Nikon's development teams are now leveraging a one-way synchronization between their wikis and Perforce and would like bidirectional synchronization. In regard to functionality, the Nikon teams rely on the Perforce APIs (Ruby, Python, and C++) for custom tools and scripts to tie into the change management system. The custom tools

handle build management. Perforce lets developers label and mark files that are ready to be released, and the build machine does a compilation on that, packages it up, and distributes it back to Perforce. Other custom tools handle branch management as well as client and label management. Perforce has significant features for branch and client management that give users a straightforward way of handling these tasks, which might seem daunting or complex to some. The API gives them a way of creating add-on tools for accomplishing these daily tasks even more easily. The Nikon teams have also leveraged the Perforce APIs for administration tasks. The teams have an automated backup procedure where they do weekly checkpoints and midweek backups. They are then notified that the backup has been completed or that it failed. Because of these tools, Nikon teams have been able to establish a set of processes and have simplified the administration somewhat; recurring tasks have been automated.

The client/server architecture of Perforce, which enables data to reside on a local copy of workspaces, means that the teams don't have to rely on a constant server connection. Perforce's version control and change list functionality lets Nikon specify which files go with which specific changes, and long description provides additional context (along with version control). These capabilities resolved the issue of developers colliding with one another when building code. Because it became easy to see that a certain file was being modified by a specific user, the developers knew exactly who they needed to contact. This enables the teams to catch and prevent versioning and change management conflicts. Perforce also has built in commands to help verify data integrity, which the Nikon teams have found useful.

Although Nikon doesn't have a change advisory board, it has an infrastructure team that manages factors related to change management and Perforce policies and procedures for using tools.

Qualitative benefits to the teams include a sense of security about data accessibility and stability — they test it thoroughly and have backup procedures so that once the information is in Perforce, they feel secure about the integrity of the data. There has been a significant improvement in development speed — enabled by the fact that developers can sit down and start working without network access issues. Perforce also gives the teams the opportunity to do code review processes and to learn from one another in order to help make sure that the code created meets established specifications.

One of the most important factors related to resolving their core problems was Perforce's strong customer support. When there were challenges or issues, Perforce was available and able to resolve them quickly. Nikon found the Perforce support team to be helpful also in terms of providing information about how it would go about scripting Perforce to provide additional, custom integrations.

Looking to the Future

The Nikon teams' biggest plan for upcoming Perforce usage expansion is to incorporate a configuration management system that ties all of their services into Perforce. They hope to leverage CMMI practices, set up a baseline, track change requests, and put in a process so that nondevelopment personnel, executives, and marketing staff in their division can access their information without having to be Perforce users. In this way, they will facilitate communication with those teams,

leveraging what is familiar to those groups for creating specifications. They have begun to do this now. The initial steps included incorporating issue tracking, the build system, smoke testing, and document collaboration, which they have gotten up and running incrementally. Within the next year or so, they expect to implement a more comprehensive, integrated solution.

New or additional features/services that Nikon would like to see in the Perforce solution, although there is nothing major, include better permission editing tools and more involvement with improving Perforce support for commercial IDE products. The teams are also interested in gaining access to processes and approaches that other companies have leveraged to structure their development environments around Perforce. Overall, the teams are very satisfied with Perforce's product capabilities, performance, and flexibility.

ESSENTIAL GUIDANCE

Key takeaways from the Nikon experience include the pressing need to choose automated SCM solutions that address performance and functional needs that match the demand for global development. In an economy where distributed development and complex sourcing are increasingly the norm for major enterprises, organizations must put in place effective code management, change, and versioning. Product infrastructure has visceral consequences for organizations and must be closely evaluated in the context of the demands that will be placed on the SCM system. Coordinating those capabilities with other areas of importance, such as business specifications and testing, also demands product flexibility (and APIs). In addition, the Nikon teams underscored the importance of the excellent, responsive customer support they received from Perforce, saying that it was critical to their success. When putting in place an effective SCM system, efficient, knowledgeable 24 x 7 support sustains global teams. Usability is also a factor; organizations must make sure it's not something that is too difficult for users to grasp and that the capabilities map to demand, process, and organizational maturity.

Global organizations should consider these factors as they evaluate and adopt appropriate SCM tools to enable improved collaboration across global teams — to help facilitate adaptive businesses and responsive software development strategies.

Copyright Notice

External Publication of IDC Information and Data — Any IDC information that is to be used in advertising, press releases, or promotional materials requires prior written approval from the appropriate IDC Vice President or Country Manager. A draft of the proposed document should accompany any such request. IDC reserves the right to deny approval of external usage for any reason.

Copyright 2010 IDC. Reproduction without written permission is completely forbidden.