



Deploying from Large-Scale Development

Case Study



BuildMonkey
<http://www.buildmonkey.com>

Table of Contents

Table of Contents.....	2
Summary	3
The Project	4
The Problem	5
Fixed Price, Fixed Deadline	5
Platform Synchronicity.....	5
3 rd Party Handover	5
The Solution.....	6
Fixed Price, Fixed Deadline	6
Platform Synchronicity.....	6
3 rd Party Handover	6
The Result	7
About BuildMonkey	8

Summary

Sun Microsystems were engaged on a UK Government project to develop a large distributed application. The price, and deadline, were fixed.

Due to the complexity of the application, deployment of all of the components was a time-consuming activity – taking four calendar days. Sun identified this as a risk, and engaged BuildMonkey to address the issue.

Through the use of the Deployment Verification System, BuildMonkey were able to reduce this four-day deployment to **eighteen minutes** – a key factor in delivering the project early.

The Project

The application being developed was a large, mission-critical, distributed J2EE application to run in a Solaris environment. The SCM facilities were provided by CVS running on a Solaris server.

There were a number of teams and organisations involved:

- Developers
- Testers (in-house and customer acceptance)
- Infrastructure (in-house and customer)
- Integration (in-house and customer)
- Production support (in-house and customer)

The project, as is the case with most large projects, had more than one environment to manage with each environment being used for a different purpose:

- Development
- Testing
- Staging
- Production

There were also multiple development tracks, i.e. more than one instance of each of these environments – to support different test environments for bug-fix and new releases for example.

The most common problem encountered in this situation was where the deployed code displays defects in one environment but not in another, and the most common question asked – by developers, testers and project managers alike, was “*why does it work here and not there?*”.

The Problem

Fixed Price, Fixed Deadline

Sun were engaged on the project on a fixed-price basis, and had engaged a number of its suppliers and subcontractors on T&M terms. This caused a considerable pressure on Sun to reduce the time for delivery.

Sun were keen to ensure that their reputation for quality and timely delivery of quality applications was maintained, as well as their profit margin.

The contract contained significant penalty clauses around late delivery, and Sun wanted to ensure that it did not encounter conditions under which such penalties would apply.

Platform Synchronicity

The large number of environments to maintain, plus the number of teams involved in doing so, presented a challenge to Project Management.

Ensuring that environments were identical, so that confidence could be had in test results, was of particular importance. Being able to say with confidence that environments, and code versions in different environments, were the same was a constant issue.

3rd Party Handover

Sun did not own, or have access to, the production infrastructure, or the production staging infrastructure. Deploying into these environments reliably and quickly proved difficult since interoperability problems could not be detected and diagnosed until deployment time – under the watchful eye of the customer.

Sun were very keen to avoid any adverse perception of the quality of their application which was indeed of the highest quality.

In other words, Sun wanted to ensure that environmental issues could very quickly be identified as such

The Solution

Fixed Price, Fixed Deadline

Goal: Sun were keen to ensure that their reputation for quality and timely delivery was maintained, as well as their profit margin.

By engaging specialists, such as Nemean Technology, to manage the deployment activities in the project, Sun successfully de-risked this activity and was able to significantly reduce the cost, and uncertainty, that is normally associated with deployment.

The deployment of this massive distributed application was reduced from **four days** to **eighteen minutes**.

Deployment issues were very quickly able to be demarcated between application code and environment. Sun delivered early, and avoided all penalties.

Platform Synchronicity

Goal: Sun were very keen to ensure that there were no differences between the multiple environments so that utter confidence could be had in test results, and that defects were reproducible in any environment.

By using the Deployment Verification System (DVS) developed by Nemean Technology, Sun were able to determine – instantly – whether two environments were the same, and whether identical versions of the application were deployed there.

3rd Party Handover

Goal: Sun wanted to be able to verify the “*fit for deployment*” status of any target environment – particularly the ones that it did not own and manage. It did not wish to incur late delivery penalties due to delays caused by environmental issues that it had no control over.

By using the Deployment Verification System, Sun were able to provide an automated suite of tests that could verify – instantly – whether or not the target environment was ready for the application.

This ensured that Sun were able to deliver code drops with confidence, and that any interoperability problems were known at a very early stage in the development cycle – when there was still time to react.

BuildMonkey. Because deployment is a thief.

The Result

By using buildMonkey software and services from Nemean Technology, Sun were able to deliver a fixed-price project early, even when they had to deploy into infrastructure that they had no visibility of.

Deployment time was reduced from four days to eighteen minutes, and became a milestone rather than a task.

BuildMonkey. Because deployment is a thief.

About BuildMonkey

BuildMonkey are the market leaders in Build, SCM and Deployment.

Formed in 1999, and with many Fortune 500 and FTSE 100 blue-chip clients, we are the original and the best.

All of the concepts described in this paper have been encapsulated in a suite of off-the-shelf tools and associated processes to facilitate rapid implementation of the Best Practices set out in this paper.

We are passionate about solving the problems which plague software development. We know that, with very little effort, it is possible for software to be delivered on-time, on-budget and free of defects.