

eCube Case Study

Cross Functional Development Increases Productivity on OpenVMS

▶ Challenge: Lower the cost and increase the speed of developing batch programs supporting critical business operations	▶ Tools: Java Eclipse IDE NXTware Remote for OpenVMS
▶ Applications: Core business operations	▶ Language Platform: Solution transitioned from C language to Java
▶ Target Platform: OpenVMS Integrity and Alpha Servers	▶ Client: Mutual Insurance Company

OVERVIEW

- The business provides diversified insurance services such as life, salary, health. With over one million customers, the company is a leading mutual insurance company (MIC) and a known leader in the group insurance industry and investment and retirement sectors.
- Solution lowered MIC's cost of operation by enabling an under-utilized pool of Java developers to take over the programming of batch processing applications on OpenVMS. These batch applications were historically written in the C language on HP VMS servers. In the process, the solution improved MIC's business performance, reduced information technology (IT) operations costs and eliminated the high costs and risk associated with moving away from the OpenVMS platform.

SITUATION

Batch processes still play a very important role in IT services at MIC, where real-time applications are updated by batch processes on a regular basis. Batch processes are used because they are still the most economical way to automate computer interactions and access expensive computing time. It is not necessary or efficient to update every aspect of a system and its data after each transaction. Additionally, new compliance rules and changes in underlying IT infrastructure make it important to have access to batch-oriented workflows that can be scheduled or triggered by business conditions.

Historically, MIC's batch programs running on OpenVMS platforms have been written in COBOL and the C language. However, over the years, the number of COBOL and C programmers in the organization has diminished. Moreover, as these skills become scarcer, it became more expensive to develop, locate and/or hire resources with these skills.

Concurrently, the company had found it easy to staff and develop projects based on the Java language. In fact, within the IT organization, there were under-utilized software engineers with these skills. These developers were very successful engineering new applications using Java and the Eclipse Integrated Development Environment on Windows and Linux-based computers. However, these engineering resources did not have the skills to work on the OpenVMS platform and its command line and non-graphical environments, nor were they interested in learning to do so.

SOLUTION

The MIC had three basic choices:

- Replace the OpenVMS systems
- Hire new C and COBOL programmers
- Cross-functionalize: Find a solution that enabled their Java engineers to work on OpenVMS

After conducting an assessment of software assets and transition options, it was deemed too expensive, risky and time consuming to replace and port 20 years of work on OpenVMS. The assessment process determined that it would cost nearly a million dollars to port and test the existing code -- and even more to

implement the replacement infrastructure and servers. This did not take into account parallel costs of running and maintaining the production systems.

Hiring new programmers presented other problems. In addition to the cost and difficulty of locating and hiring new C/COBOL developers, there was the fact that it would take time for new programmers to learn the business and become productive. Most importantly, the company at the time was trying to reduce head count, not add new developers with specialized legacy skills.

The approach preferred by the Director of Infrastructure and Integration and recommended by the assessment was the third option: Find a solution that enabled their Java engineers to work on OpenVMS. It was agreed that the ideal solution would combine modern tools that their developers already used. In their environment Eclipse was the IDE tool of choice. The final requirement was the ability to develop remotely on the OpenVMS Alpha/Integrity servers.

PROCESS

The IT leadership at MIC was determined to implement the most cost-effective and functional solution. After speaking with HP, they were introduced to eCube and its NXTware Remote Product. NXTware provides Eclipse-based Java developers a means to develop code on local workstations (Windows or LINUX) then remotely compile, run and debug code on OpenVMS Alpha/Integrity servers.

There was some skepticism among the OpenVMS administrators and caution within management, so a Proof-of-Concept was arranged. A team of three Java developers were provided with NXTware Remote Studio, the Eclipse plug-in, while an Alpha server used for development and testing purposes was installed with a NXTware Remote Server component.

Management then specified new batch program business logic to be developed. Without any training in OpenVMS development languages, operating systems, nor the use of telnet, FTP or ANT, these Java developers were able to develop, compile, execute and debug business logic on OpenVMS. The developers and the OpenVMS administrators both gave the product high marks. Some of the evaluators suggested small modification to better suit their process. Before the completion of the PoC, these changes were implemented by eCube, tested and accepted by the client.

Soon afterwards, the MIC agreed with eCube to a site license that would allow them to expand their use of OpenVMS to Java developers. At the same time, it identified and closed new internal business based on these capabilities.

ROI

The MIC reaped benefits in improved development productivity through the use of cross functional development teams, while eliminating the need to hire new developers skilled in OpenVMS.

Productivity:

- increased development output by a factor of three in initial tests and projects; The team immediately went from two developers writing batch programs to a flexible team that could expand to more than six on demand.

Business operations:

- reduced operational risk by eliminating unnecessary system and architectural change
- were able to increase productivity without increasing head count
- improved ability to meet business needs due to a more agile and responsive development team

Implementation information and contact

www.ecubesystems.com — transformationinfo@ecubesystems.com — US Phones: 617 628 7112

TECHNOLOGY

Application Platforms NXTware Remote
Operating System Windows/LINUX and OpenVMS
Development Language Java

IMPLEMENTATION

Services Provided Executive Assessment
Other options evaluated Hiring new C/OpenVMS developers;
re-building existing capabilities and porting of logic
Actual Costs software and consulting 7 -10% cost of
other alternatives
Implementation Team size One
Implementation time One week, in addition to a three-week PoC

Implementation information and contact

www.ecubesystems.com — transformationinfo@ecubesystems.com — US Phones: 617 628 7112