CUSTOMER CASE STUDY: **LARGE EUROPEAN BANK**

Seeing is Achieving

How Visualization Helps a Major European Bank Manage More Mainframe Code Better, Faster and with Fewer People

BUSINESS CHALLENGE

As the Lead Software Configuration Manager for one of Europe's largest private banks, R.C.* faces a common IT management challenge. The bank stopped developing new mainframe applications and allowed its mainframe development staff to shrink around 30% by attrition. But the bank's mainframe development needs are increasing.

For one thing, use of the bank's mainframe applications and data by new and existing distributed, web and mobile applications is increasing. At the same time, developers no longer build new COBOL applications but must constantly modify mainframe code to assist non-mainframe applications.

For another, the bank can't wait forever for coding changes to be completed. Competition in the financial services market is intense, making time-to-benefit important, whether for customers' mobile experiences or new capabilities for staff, and putting pressures on the mainframe software development lifecycle.

That lifecycle has to produce defect-free code. The bank's mainframe applications are critical to everything it does and everything customers and regulators expect. Mainframe test/QA must be rigorous and fast. If the testing process uncovers a problem, it must be rapidly and accurately diagnosed, fixed and tested again.

Another issue is the more the bank's non-mainframe applications tap into its mainframe environment, the more vigilant R.C. and his team must be about performance and capacity utilization issues. Every new query and transaction originating outside the mainframe adds to the mainframe workload. If a non-mainframe/mainframe interface is badly designed, it can monopolize a mainframe resource, adversely impacting other systems.

"It's not easy when you're stuck between disinvestment in the mainframe and increasing pressures to make a growing number of relatively small but potentially high-impact changes across all your mainframe programs," says R.C. "That obviously forces mainframe managers like me to do things a lot differently than we have in the past."

SOLUTION

On R.C.'s team, disinvestment affects the mapping of mainframe developers to mainframe applications. When staffing levels were higher, a lead developer took ownership of one or two applications and maintained familiarity with the evolving inner complexities and idiosyncrasies of each. Today, as headcounts decline, long-term mapping is gone and developers must work on unfamiliar, poorly-documented applications.

CUSTOMER

Large European Bank

INDUSTRY

Banking

CHALLENGE

The bank's need for fast, defect-free mainframe development continues to escalate, even as it reduces its mainframe development staff by attrition.

SOLUTION

Compuware Topaz™

Compuware Xpediter®

Compuware File-AID®

Compuware Abend-AID®

RESULTS

- Reduced time to discover and resolve issues in application relationships, runtime behaviors and program logic flows
- Clarified impact of code changes in advance to avoid re-coding and re-testing
- Reduced batch processing time after visualization of application's runtime activity in high-level production
- Consolidated mainframe and non-mainframe environments on single interface to facilitate collaboration between developers of all skill levels

Compuware®

Initials are being used due to the bank's attribution policies.

Lack of familiarity makes it difficult and time-consuming for developers to understand how coding changes impact primary applications and applications dependent on primary applications. In test/QA, lack of familiarity with applications makes it difficult and time-consuming for developers to diagnose root-causes of functional failures or poor performance.

Given the complexity-accumulated inter-dependencies between applications and databases, developers even struggle to understand coding and performance issues within applications they know relatively well.

The solution, according to R.C., has been embracing visualization with Compuware tools like Topaz, Xpediter, File-AID and Abend-AID, providing developers with visually intuitive insight into application relationships, runtime behaviors and problems in programming logic flows.

Issues that previously took hours to find now take developers minutes with these visualization tools, which also help developers understand in advance the implications of coding changes to avoid re-coding and re-testing.

“I know some people view visualization as a mere ‘nice-to-have’ that just sort of makes life a little easier for developers without necessarily creating major advantages for the business,” says R.C. “But visualization is actually a powerful and essential enabling technology for any enterprise—especially a world-class financial institution likes ours—that has to leverage its mainframe applications and data in the context of its total digital business effort.”

RESULTS

In one example, given the ability to graphically visualize a mainframe application’s runtime activity in high-level production and “drill down” to a detailed view, R.C.’s team noticed a batch process was completing later and later. If the trend continued, the process would not complete before the open of business, spelling trouble for the bank.

Using Compuware’s Topaz for Program Analysis, R.C.’s team zeroed in on a sort-program call repeating hundreds of thousands of times a night. A simple code fix reduced that egregious performance bottleneck to a single call.

R.C. also points out that a well-visualized dev/test workspace with the right tool integrations makes it easier for developers to move between mainframe and non-mainframe environments, regardless of their “home field.”

This benefit is vital for two reasons. First, as mainframe veterans retire from the bank, Java/x86 developers must pick up the slack, but not with dated green-screen tools. Instead, they use highly intuitive visualizations to navigate unfamiliar, poorly documented, complex mainframe applications. In one case, rapid visualization helped R.C.’s team avoid a potentially disastrous technology snafu, saving IT hours of frantic troubleshooting.

Second, as R.C.’s sort-program call example demonstrates, mainframe and non-mainframe code must increasingly work in tandem, and keeping them in silos is counter-productive. Visualization gives all developers on all platforms a single, common basis for collaborating on great applications.

“Many people don’t realize how often the success of a non-mainframe application rollout or update depends on a seemingly small but critical piece of mainframe code,” says R.C. “Given this reality, it’s a very good idea for IT to invest in new visual mainframe DevOps tools that make it as easy as possible for development and test/QA staff at all skill levels to modify mainframe code as required—quickly and with the highest degree of confidence.”

To learn more, please visit: **compuware.com.**

The Mainframe Software Partner For The Next 50 Years

Compuware empowers the world’s largest companies to excel in the digital economy by fully leveraging their high-value mainframe investments. We do this by delivering highly innovative solutions that uniquely enable IT professionals with mainstream skills to manage mainframe applications, data and platform operations.

Learn more at Compuware.com.

© 2016 Compuware Corporation. Compuware products and services listed within are trademarks or registered trademarks of Compuware Corporation.