



# The enemy within

Sarah Salzman, Compuware's Technology Manager for Automated Software Quality, on the dangers of dormant code and how to find it

Over the past few years security has become one of the most talked about topics in the IT sector. Businesses have made huge investments in technology to try and ensure hackers cannot penetrate their networks and cause serious financial damage. However, the enemy can also lurk from within and steal millions without the company even knowing. For example it cost the Jasper State Bank \$2.7million before they realised that two former employees were engaged in fraudulent activities.

Organisations are tackling the issue of fraud by carefully vetting and monitoring employees (including following up on all references) that have access to sensitive financial systems. Employee vetting is a critical part of the recruitment process because typically with development projects there is a high turnover of staff. People tend to move around with high regularity in this industry. The predictable churn of staff within a project, in addition to project managers hiring contractors (to address the peaks and troughs or to utilise staff with a particular skills set required at that point) is a cause for concern from a security perspective. Having a high turnover means that there is a larger potential/opportunity for people to place malicious code into your applications. The problem is that even vetting and monitoring employees does not always protect against former staff that may still be in a position to commit fraud.

Most organisations do not view current or former employees as an immediate threat, because they are very diligent about managing passwords and access rights when people move on. However, many may be unaware that systems or applications can be developed in such a way that features are built in to enable someone to steal millions of pounds from company accounts or destroy applications without leaving a trace. Fraudulent application developers can insert lines of code that remain dormant for several years, then when/if they leave the company the code may become active. This code is often carefully hidden away within the application and therefore will only be found if people are specifically looking for it.

Presently most businesses do not have the controls or processes in place to protect against criminals who have the technical knowledge to insert fraudulent code into IT systems. Most security managers would not be able to find this type of threat because very specialised and technical knowledge is generally required to discover this malicious code. Methods can however be employed within the specialist application testing or quality assurance teams to trap fraudulent code.

By adopting testing practices that work in combination with code coverage analysis tools, the test teams determine:

- 1 the functional completeness of test procedures being used
- 2 test results based upon code coverage measurements and metrics.

In taking your test procedures further by drilling down to assess the lines of code that have been run during a test, you can visually and easily determine which elements of code have and have not been tested. For example, in figure 1 you can

see what percentage of the code has been covered during the test run.

Next it's important to be able to drill down further to examine exactly which components have or haven't been tested. The diagram below illustrates this easily. The red lines of code have not been tested, whereas the green sections have. This kind of detailed information can stand as tangible evidence (at both summary and detailed level) to prove which code has, and more importantly, has not been tested (figure 2).

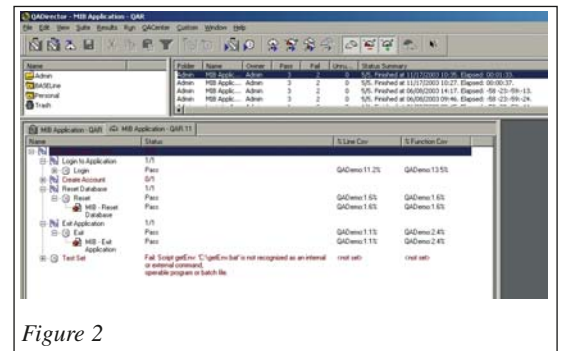


Figure 2

Companies need to turn to the testing teams and look at the testing processes they have in place. It tends to be the norm for people to test only active code (code that actually contributes to the running of an application), but this needs to change if businesses are to protect themselves from rogue developers. They need to make sure that more of the code is tested, so that dormant code is identified, examined and analysed for fraudulent attributes. Embedding code coverage analysis as an element to the testing strategy will ensure that the testing teams can highlight easily lines of code that may not have been tested, prompting a revision in test procedures to generate more testing scenarios aimed at uncovering fraudulent code. Obviously increasing the amount of testing will cost money, but if you consider that the Association of Insurers recently said that fraud costs the UK economy upwards of £15 billion per year, you can see why it is crucial that businesses act to ensure that their application development and testing processes are watertight, guaranteeing fraudsters cannot use dormant code to steal money from the business.

PT

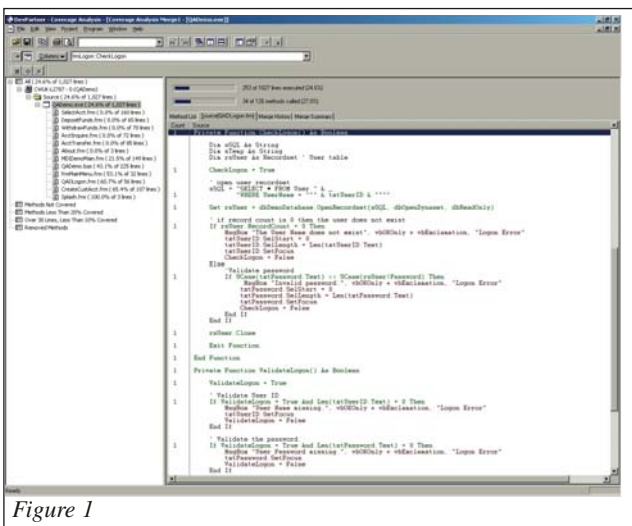
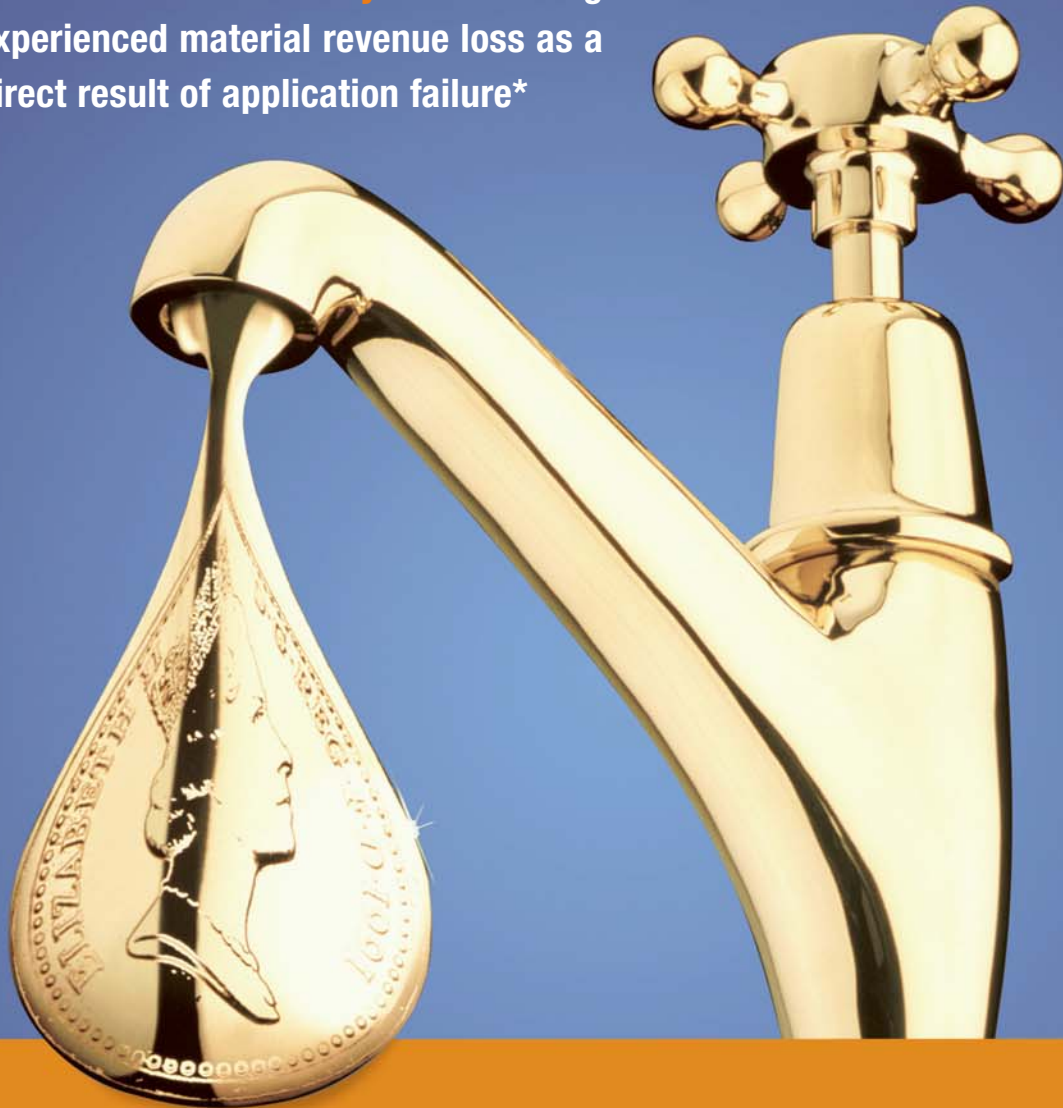


Figure 1

# Are your applications leaking money?

**64% of IT Executives say Yes...** having experienced material revenue loss as a direct result of application failure\*



In today's competitive environment, IT departments need to release increasingly rich feature sets across complex distributed infrastructures. To reduce the risk of costly errors, analysts such as Forrester Research and Patricia Seybold, recommend an Automated Software Quality (ASQ) solution.

To learn how your software projects can be a third less expensive\*\* and to download your ASQ information pack with Patricia Seybold white paper, visit:

**[www.compuware.co.uk/money](http://www.compuware.co.uk/money)**

\* Forrester Research - 2003 \*\* Patricia Seybold Group - 2003

**COMPUWARE**  
[www.compuware.co.uk](http://www.compuware.co.uk)

