



# QF-TEST

## Professional GUI Testing for Java and Web

QF-Test Customer Report of Thales Australia

**What QF-Test means for us regarding ROI**

by Denis Gauthier

July 2010

Let's say the automation of our tests is successful and this is a big step forward. I am not an expert on all the other available tools but from what I know, I don't see how I could have interfaced other tools than QF-Test with our system, moreover in such a short period of time.

## The few points that make QF-Test the killer application :-) :

- **A mature and complete HMI to design the Test Suite.**
- **A synthetic report generated automatically in HTML.**
- **No need to hack into the JVM or the application under test to make things testable.**
- **Jython scripting: You can do anything you need with this.**
- **Extension API: This allows to make QF-Test aware of internal objects inside the GUI application.**

In terms of how far we went into the integration of QF-Test into our process: Our Non-Regression test have been almost fully implemented under QF-Test. It takes roughly three hours to run and hence saves me heaps of time.

In terms of ROI, there are several aspects: Regarding the time saving aspect, it usually took me a full day to perform the Non-Regression Tests. It has now been reduced to three hours. The big difference is that I can do something else in the time saved! The direct benefit is that I can now run some tests on different versions of our System (i.e. different branches) in parallel, when it was impossible before (not enough time).

Another less visible aspect is that it improves the communication with the Software Development. There are no 'feelings' or personal judgement in the appraisal of the results. It happened previously that depending on who was running the tests, the results were slightly different. Indeed, as I mentioned this is a real time System and depending on what action you were doing at what time, the behaviour of the System might not always be the same. Having automated the tests removes (or decreases) these timing issues.

Software Development cannot say anymore that you stuffed up or did not use the System properly during the tests (and asking you to redo them and hence gain some time ...). It is always the same tests that are reproduced and if the results changes, it is because the System has changed. Sometimes good changes (improved functionality, bug fixes, faster response times ....) but the automation picks that up, which is the whole point of the NRG Testing: detecting changes.

There are some drawbacks though in automating the tests:

First, tests need to be maintained! The SUT is evolving and the components need to be updated. New functionalities are also introduced and they need to be added into the scope of the tests.

Then the tests are not perfect: they are not bullet proof and sometimes an issue can generate a domino effect and void the complete result. If you are not in front of the machine during the test, you will not see it and possibly loose three hours. So every time it happens, I fix the test and add more checks and protections. But once in a while I need to relaunch the test.

Then QF-Test will only detect issues that you taught it to recognize. In comparison, a human tester would be able to detect odd behaviours of the System. Again, it is real time and with a lot of algorithms. Not everything is predictable. That means that issues might not be detected by QF-Test (or more accurately the Test Suite I wrote). This is usually detected further down the test chain (IBB, V&V Testing ...) but then it costs a bit more to correct. Every time it happens, I try to improve the Test Suite, which is a bit time consuming.

Well, you may wonder then if there is a real gain in automating tests :-\ My conclusion would still be yes, but the ROI is not on the time. The time you do not spend on the testing anymore needs to be spent on different topics: maintenance, improvements, analysis of what you want to test and how you want to test it (not always as straightforward as one could think). The gain you obtain is on the quality and consistency.

**Denis Gauthier**

Software Integration, Thales Australia, Melbourne