

No. 01/08

March 15th, 2008

# te testing experience

The Magazine for Professional Testers

printed in Germany

free exemplar

[www.testingexperience.com](http://www.testingexperience.com)



## Testing & People



## Why can't testers code?

By James Lyndsay

Perhaps it's the circles I move in, but it seems that I can't get through a conference without someone telling me, in some fashion: "Coders don't test". If it's not hearing the latest coding-idiot bug, it's the refrain that testers need to be independent - even isolated - from the bug-forgiving influence of the design teams, or someone whining that their coding colleagues aren't remotely interested in all the good work testers put in. I've heard testers gasp in horror on discovering that the coders are writing their own tests, without benefit of ISTQB qualification or the most basic grounding in test design. Something clearly needs to be done.

On the other hand, I've worked in organisations where testing has been an innovation-free zone. Some teams are still writing test scripts in Word, and printing them out before ticking the boxes. Some teams manually configure their test environment, every time, before laboriously hand-crafting their data. Some check reports by eye, and rely on their memory to remind them about whatever they did just before finding their most recent bug. When I teach exploratory testing, I'm always surprised to find people in teams who hardly use tools, don't share the tools they might use, and certainly don't have the wherewithall even to chuck together a quick script to check that the parameters of the test environment are much the same this morning as they were last night. Curiously, it's these testers who seem most disturbed that coders are writing their own tests.

Here, then, is my take on the matter. It is not that coders don't test, it is that testers can't code. This is a skills gap - whereas coders can test, but sometimes don't, it's uncommon to find a tester who can read code, control a debugger, and fix the bugs they find. Is it any wonder that testers don't get no respect, that testers can't communicate with coders, that testers simply don't understand the range of bugs that using = rather than == can trigger? Is it any wonder that other people on the team are starting to take over the tester's job with vast numbers of unit tests, and business acceptance tests that actually make sense to the business?

Some of you reading this will be quite rightly offended by my blanket characterisation of technology-adverse testers. Some of you may already be excellent coders, whipping up a bug-tracking system or database loading script in your lunchbreak. For the rest of you, perhaps, this will be a call to arms: your colleagues are doing the interesting parts of your job, and you're rolling over to let them. Ignore, for now, the proprietary tools that let you test your whole application with a single button press - don't be tempted by their scriptless attractions. Turn instead to your library, bookshop or web browser, get hold of a beginners guide to a language you can code on your own laptop, and get moving. Write small, and write useful - something to spot changes or load data. Don't try to write automated tests. If you fancy, use a language that your coders use, get them to show you their development environment, the way they build their code and make their living. Perhaps you'd prefer to turn to the powerful and simple unix scripting tools, or their equivalents on any operating system worth a mention. Perhaps you'll pick up Python, Perl or PHP. Any way you choose, engage with the technology that powers our business, get over the hump, and become a tester who codes.



### Biography

Based in London, James Lyndsay has been testing since 1986. He formed Workroom Productions in 1994, and has been an independent test strategist ever since. Having started coding in 1981, James has forgotten most of what he once knew, but currently codes rather poorly in combinations of PHP, unix scripts, and Action-Script/Flash. He feels this gives him many powerful test techniques, primarily empathy with the frustrations of professional coders. James was an internal irritant to the ISEB exam process for five years, is a regular speaker and occasional teacher, and runs LEWT (the London Exploratory Workshop in Testing). Visit <http://www.workroom-productions.com> for papers, recorded talks and tools.

### Future public engagements:

UK: London SIGiST, special session on Diagnosis, keynote on "Testing in an Agile Environment" 18 March

Denmark: Exploratory Testing Seminar, 29 April

USA: STAREast Diagnosis tutorial May 6