

## **‘Don’t be thorough, be effective.’**

*... sample your way to test success!*

Some published descriptions of popular software testing techniques recommend, or at least imply:

- ✘ **That we should test all the coverage items (test conditions) that the techniques identify.**

*EG. When using ‘equivalence partitioning’, descriptions of this technique imply that we should design and develop tests that: ‘collectively exercise **all** the identified partitions, both valid and invalid’.*

*EG. With boundary value analysis (BVA), the impression is given that: we’re not done until we have a set of tests that exercise every identified boundary value. Not necessarily!*



### **THE PROBLEM WITH THIS IS**

- **Some testers follow this guidance unquestioningly and consequently waste significant time and effort** implementing and executing tests that achieve this supposed ideal 100% coverage of the identified coverage items.
- **Some testers choose not to use test techniques at all** (as they see the effort required to achieve 100% coverage).

### **Be careful where you spend your time**

*... ‘being thorough with any single technique’ is at the cost of ‘not using a richer, more diverse approach’.*

It’s all well and good if your thoroughness and dedication to test techniques pays off in terms of an improvement in detection rates, but often it doesn’t. Instead, the one technique approach detracts from the overall effectiveness of the testing because the effort spent being focussed on one technique is at the cost of not using a richer, more diverse approach.

### **Complementary non-systematic techniques is almost always a good thing**

Although I have no empirical evidence of this to hand, I perceive (*from my testing and consultancy experience*) that: **the effectiveness of testing is most often achieved through variety, rather than thoroughness.**

Using a diverse set of test techniques quickly can be much more effective than using just one (*or two*) techniques thoroughly. Indeed, using techniques in conjunction with complementary non-systematic techniques will almost always be a good thing to do.

(Each technique is good at finding defects of a particular type. Software generally has defects of many different types, so it is important to cover a variety of test techniques, rather than just one or two.)

For example, applying the boundary value analysis (BVA) technique thoroughly can easily identify 10's of boundary values for even relatively simple software. It is unlikely that 'all boundary values will reveal unique defects'.

### **Identify areas for exploration by trialling subsets of conditions.**

For any testing technique, selecting a representative subset of the test conditions identified by the technique as a sample will make a good start. We can then use the results of these tests to help gauge the value of testing a larger sample. For example when using BVA, if the first sample finds any defects, testing further boundary values may be worthwhile, whereas if no defects are found, testing further boundaries is less likely to be fruitful.

### **THE RESULT: Free up time to target different 'types' of defect**

Your time and effort saved by not achieving 100% coverage of 'resulting identified test conditions' can be better spent using a variety of different test techniques.

Mark Fewster.

Consultant, Author, Speaker.

Director and Owner of Grove Software Testing Limited.