

## Identifying some problems with selection of software testing techniques

SHEIKH UMAR FAROOQ and S.M.K. QUADRI

Department of Computer Sciences, University of Kashmir, Srinagar (India).

(Received: August 12, 2010; Accepted: September 04, 2010)

### ABSTRACT

Testing techniques refer to different methods or ways of testing particular features of a computer program, system or product. Presently there are so many different software testing techniques that we can use. Whether we decide to automate or just execute tests manually, there is a selection of testing techniques to choose from. We have to make sure that we select technique(s) that will help to ensure the most efficient and effective testing of the system. The fundamental problem in software testing thus throws an open question, as to what would be the techniques that we should adopt for an efficient and effective testing. Thus, the selection of right testing techniques at the right time for right problem will make the software testing efficient and effective. In this paper we discuss how should testing techniques be compared with one another and why do we face a problem in making appropriate testing technique selection.

**Key words:** Software testing; Testing techniques; Testing techniques selection.

### INTRODUCTION

Software testing is the process of executing a program with the intent of finding errors<sup>1</sup>. Or, it involves any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results<sup>2</sup>. Software testing is also used to test the software for other software quality factors like reliability, usability, integrity, security, capability, efficiency, portability, maintainability, compatibility etc<sup>3</sup>. Successful testing is a critical concern for most of the software majors. There are a variety of types of tests that are performed at different stages of the software development lifecycle such as unit tests, integration tests, systems tests and acceptance tests. Each of these can be further divided into types of testing such as functional, structural performance, regression, or usability tests, just to name a few. In each testing type there are many software testing techniques that are used to test a system. At present

mostly selection of testing techniques is done neither systematically, nor following well-established guidelines. The problem we face is how to make tuned selection of testing techniques so as to perform more effective and efficient testing and we do not have a standard comparison criterion for comparing testing techniques. Solving these problems would help testers to choose the best suited testing techniques for every project.

### Characteristics of a Good Testing Technique

Which testing technique is best? Each technique is good for certain things, and not as good for other things. Each individual technique is aimed at particular types of defect as well. For example, state transition testing is unlikely to find boundary defects. Testing Techniques should assure maximum effectiveness with the least possible number of test cases. The "right technique" is the one that lets you achieve your goal, and which you can accomplish in your current situation.

Nevertheless, tester's face the question, which are the best-suited techniques every time they have to test a system<sup>4</sup>. Some techniques are more efficient in finding failures than others and some are easier to apply than others are. Some techniques are more applicable to certain: situations and test levels; others are applicable to all test levels. Each testing technique meant for testing has its own dimensions i.e. for what purpose it is used, what aspect it will test, what will be its deliverables etc. It is imperative to find the most effective and efficient testing technique but that should not be practically impossible.

Focusing on selection of testing techniques, there are still many decisions to be made about which techniques are the best. Characteristics of good testing technique are:

- ' High probability of finding errors (effectiveness).
- ' Probability of finding undiscovered errors.
- ' Achieves its desired goal in the least amount of time and budget.
- ' Non-redundant.
- ' Right level of complexity.

#### **Comparison Criteria for Testing Techniques**

A basic question is that how can we compare testing techniques with each other. A general approach is to compare the effectiveness of testing techniques. Test effectiveness is a measure of bug finding ability of the testing technique<sup>5</sup>.

Test Effectiveness = Errors reported by Testers / Total Errors reported.

Where Total Errors = Tester reported + User reported Errors.

This approach of evaluating the impact of a particular testing technique can be adjusted in several ways. For example, failures can be assigned a severity level, and test effectiveness can be calculated by a level. In fact the number of errors found in test effort is not meaningful as a measure until we combine it with the severity of errors, type of errors found and so on. In this way, we can say testing technique A might be 50% effective at finding faults that cause critical failures, but 80% effective at finding faults that cause minor failures.

#### **Effectiveness of testing can be adjusted using several parameters**

- ' Increase in software reliability.
- ' Software type.
- ' Error detection effectiveness (detection of most errors).
- ' Error detection cost (#errors/effort).
- ' Error type (Class of errors found: Critical, Serious, Medium and Low).

It is not easy to compare the effectiveness of the different techniques. The effectiveness of a technique for testing particular software will, in general, depend on the type of errors that exist in the software. However, based on the nature of the techniques, one can make some general observations about the effectiveness for different types of errors. For comparison, it is best to first classify the errors into different categories. One such comparison is given in by<sup>6</sup>. Another way of measuring effectiveness is to consider the "cost effectiveness" of different strategies, that is, the cost of detecting an error by using a particular strategy. And the cost includes all the effort required to plan, test and evaluate. A classification on the basis of the strengths & weaknesses of each technique (theoretical, technical & pragmatics aspects) would be much more useful than classifications based on mechanical or operational considerations.

#### **Problems with Selection of Software Testing Techniques**

The aim is not for the tester to design every possible test case, but rather that he selects a specific technique in relation to the selected test strategy - aiming to achieve the highest possible 'defect-finding chance' with the least possible number of test cases. When choosing a testing technique, practitioners want to know which one will detect the faults that matter most to them in the programs that they plan to test<sup>7</sup>. Claims that it remains an open issue. One sure thing is that while there are comparative studies between techniques, there are no studies that examine the conditions of applicability of a technique at length or assess the relevant attributes for each technique. Additionally, existing studies show contradictory results<sup>8</sup>. Although a large number of software testing techniques have been proposed, we are basically

ignorant of their respective powers. There is still not adequate proof to indicate which of these techniques are effective. It is difficult to perform meaningful comparisons of the effectiveness of testing techniques. The problem with testing techniques in research is that they are mostly made on a small sample, and often with the demonstration that they either perform better than random, or another specific technique. The problem with testing techniques in industry is that they are not known (many testers have no training in testing techniques), not utilized, since often there is a belief of their efficiency and effectiveness, and it is seldom proven for larger complex system<sup>9</sup>. The main reason for this is the difficulty for research is to get enough large amount of real data to perform research on (code sample selection is in majority below 2K) and comparing techniques between them. The actual research setting of creating reasonable comparative models have not been totally explored. The main focus is often to invent a special technique, and compare its effectiveness' with one already known (often a similar technique). The knowledge for selecting testing techniques should come from studies that empirically justify the benefits and application conditions of the different techniques. However, as authors like<sup>10</sup> have noted, formal and practical studies of this kind do not abound, as:

- (i) It is difficult to compare testing techniques, because they do not have a solid theoretical foundation;
- (ii) It is difficult to determine what testing techniques variables are of interest in these studies<sup>11</sup>.

Vegas<sup>12</sup> cite two main reasons why developers do not make good choices. Both refer to their knowledge about existing techniques and their properties.

The information available about the techniques is normally distributed across different sources of information (books, articles and even people). This means that developers do not have an overall idea of what techniques are available and of all the information of interest about each testing technique.

There is no access to the pragmatic information concerning each testing technique unless they used it before. Developers do not tend to share the knowledge they acquire by using a testing technique with others. This means that they miss out on the chance of learning about the experiences of others.

In general, the problem of software testing technique selection is due to following reasons:

- ' We have wide variety of testing techniques<sup>13</sup>.
- ' In the context of testing technique selection, the term best has different meanings depending on the person making comparisons<sup>14</sup>;
- ' We do not have an overall idea of what techniques are available and of all the information of interest about every testing technique.
- ' We have no access to practical information pertaining to testing techniques unless we have used it before. We do not tend to share the knowledge we acquire by using testing techniques with others.

The processes, techniques and tools used in the development of software systems are not universally applicable<sup>15</sup>, and this also applies to testing techniques, which are not equally applicable for validating the system.

## CONCLUSION

Perhaps the single most important thing to understand is that the best testing technique is no single testing technique. Different techniques have different strengths and weaknesses. No technique is good at detecting all types of errors, and hence no one technique can suffice for proper verification and validation. To be able to evaluate techniques to find out which one is best among them in terms of effectiveness, efficiency and applicability, one needs to carry out experiments on a large scale, under a common standardized framework. But creating a framework for defining, and juxtaposing techniques is no simple matter in present testing scenario.

## REFERENCES

1. Myers, Glenford J., "*The Art of Software Testing*", New York: Wiley, c1979. ISBN: 0471043281.
2. Hetzel, William C. "*The Complete Guide to Software Testing*", 2nd ed. Publication info: Wellesley, Mass.: QED Information Sciences, 1988. ISBN: 0894352423.
3. S.M.K Quadri and Sheikh Umar Farooq, "*Software Testing – Goals, Principles, and Limitations*", *International Journal of Computer Applications*. (0975 – 8887) 6(9): (2010).
4. S. Vegas, "*Identifying the Relevant Information for Software Testing Technique Selection*", *iseese*, 39-48, 2004 *International Symposium on Empirical Software Engineering (ISESE'04)*, (2004).
5. Sheikh Umar Farooq and S.M.K. Quadri, "*Effectiveness of Software Testing Techniques on a Measurement Scale*", *Oriental Journal of Computer Science & Technology*, 3(1): 109-113 (2010).
6. H. Dunn, "*Software defect removal*", McGraw-Hill Inc., (1984).
7. A. Bertolino, "*Guide to the knowledge area of software testing*". Software Engineering Body of Knowledge, February 2001.
8. J. Rowland and Y. Zuyuan, "*Experimental comparison of three system strategies, preliminary report*", In *International Symposium on Software Testing and Analysis*, Pages 141-149, Key West, Florida, USA, ACM (1989).
9. Weyuker, E.J.; "*Can we measure software testing effectiveness*", *Software Metrics Symposium*, 1993. *Proceedings, First International*, 21-22(s):100 – 107 (1993).
10. Hamlet, R., "*Theoretical Comparison of Testing Methods*". In *Proceedings of the ACM SIGSOFT '89 Third Symposium on Testing, Analysis and Verification*. 28-37, Key West, Florida, ACM.
11. Natalia Juristo, Ana M. Moreno and Sira Vegas, "*State of the Empirical Knowledge on Testing Techniques*", ACM (2001).
12. S. Vegas, "*Which information is relevant when selecting testing techniques*", In *Knowledge Systems Institute, editor, Proceedings of the 13<sup>th</sup> International Conference on Software Engineering and Knowledge Engineering*. 45-52, Buenos Aires, Argentina (2001).
13. T.Y. Chen and Y. T. Yu, "*On the expected number of failures detected by sub domain testing and random testing*", *IEEE Transactions on Software Engineering*. 22(2): 109-119 (1996).
14. S. Ntafos, "*A comparison of some structural testing strategies*." *IEEE transactions on software engineering*, 14(6): 368-874 (1988).
15. V.R. Basili and H.D. Rombach. "*Support for Comprehensive reuse*." *Software Engineering Journal*, 303-316 (1991).