Proceedings of the 5th National Conference; INDIACom-2011
Computing For Nation Development, March 10 – 11, 2011
Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi

Testing Techniques Selection: A Systematic Approach

S. M. K. Quadri[1] and *Sheikh Umar Farooq[2]
[1]Head, P. G. Department of Computer Sciences, University of Kashmir
[2]Research Scholar, P. G. Department of Computer Sciences, University of Kashmir
[1]quadrismk@hotmail.com, [2]shiekh.umar.farooq@gmail.com

ABSTRACT
Software testing is vital and challenging activity in SDLC. The complexity and size of the software is increasing multi-fold. A multitude of techniques have been proposed for software testing, but the key problem in software testing is the selection of most effective technique. A great deal of research has been carried out to evaluate the effectiveness and efficiency of various testing techniques. However, up to now no study was able to present a universally acceptable solution. As no testing technique provides a single, comprehensive solution; the selection must be done according to a given state. We present here a decision support approach for selecting the most suitable testing technique on the basis of number of identified factors that influence the selection of appropriate techniques. Selection of testing techniques based on factors identified in this paper can improve the effectiveness of testing process significantly.

KEYWORDS
Software Testing, Testing Techniques, Technique Selection.

INTRODUCTION

Software Testing is the process of executing a program or system with the intent of finding errors [1] 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 [2]. 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 [3]. 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 so that the end result is a high-quality product. At present mostly selection of testing techniques is done neither systematically, nor following well-established guidelines. 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 [4] 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 [5].

The problem we face is how to make tuned selection of testing techniques so as to perform more effective and efficient testing. Solving this problem would help testers to choose the best suited testing techniques for every project. We suggest here a set of factors for selecting testing techniques. Selecting testing techniques considering these factors will yield better results.

TESTING TECHNIQUES AND THEIR USAGE

Software testing techniques are diverse methods to do software testing and authenticate them. Testing techniques refer to different methods of testing particular features a computer program, system or product. Testing techniques means what methods or ways would be applied or calculations would be done to test a particular feature of software. One of testing aims is to expose failures (as many as possible), and many popular test techniques have been developed for this objective. These techniques variously attempt to "break" the program, by running one [or more] test(s) drawn from identified classes of (deemed equivalent) executions [6]. Test techniques should find greatest possible number of errors with manageable amount of efforts applied over a realistic time span with a finite number of test cases. Some techniques are easy; others require a little experience to really employ effectively. The beauty of software testing techniques is that the more you use each the greater insight you will gain. You will understand when to use them, how to implement them, how to execute them, and will have a clear knowledge of which ones to use in any given situation. Most projects will require you to use several software testing techniques to ensure thorough test coverage.

CHARACTERISTICS OF GOOD TESTING TECHNIQUES

Which testing technique is best? Each technique is good for certain things, and not as good for other things. For example, one of the benefits of the benefits of structure-based techniques is that they can find things in the codes that aren't supposed to be there, such as "Trojan horses" or other malicious code. However, if there are parts of the specification that are missing from the code, only specific-based techniques will find that structure-based techniques can only test what is there. If there are things missing from the specification and from the code, then only experience-based techniques would find them. Each individual technique is aimed at particular types of defect as well. For example, state transition testing is unlikely to find boundary defects.

Corresponding Author

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 [7]. Nevertheless, tester's face the question, which are the best-suited techniques every time they have to test a system [8]. 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

PROBLEM IN MAKING TESTING TECHNIQUES SELECTION

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 [9]. The decisions made regarding technique selection are so much unsystematic. The problem of testing technique selection is due to following reasons:

- ➢ We have wide variety of testing techniques [10] [11].
- ➢ In the context of testing technique selection, the term best has different meanings depending on the person making comparisons [12][13];
- ➢ 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 [14], and this also applies to testing techniques, which are not equally applicable for validating the system.

Therefore, the problem we face is how to make selection of testing techniques in effective way.

FACTORS FOR SELECTING TESTING TECHNIQUES

With so many testing techniques to choose from how are testers to decide which ones to use? So how can we choose the most appropriate testing techniques to use? Selection of testing techniques should be done taking into account different aspects/factors(both internal and external) which include but are not limited to: the system being tested, the types of defects they contain, who's doing the detection, how it's done, for what purpose, and in which activities, resources available for testing, time allotted for testing, available expertise, experience. Factors also include which criteria govern the evaluation.

The factors that influence the decisions about which technique to use are [15]:

Type of system

The type of system (e.g., graphical, embedded, financial, etc.) will influence the choice of techniques. For example, a financial application involving many calculations would benefit from boundary value analysis and GUI system will be more benefitted by a GUI testing technique.

Life cycle model

Life cycle model used in software development also impinge on the testing technique selection. For example, a sequential life cycle model will lend itself to the use of more formal techniques whereas an iterative life cycle model may be better suited to using an exploratory testing approach.

Aspect

Which aspect of the software development are we assessing? Requirements? Design? Code? For Example, reviews are more beneficial in design as compared to code Similarly if code is checked for consistency, code reading technique will be beneficial, whereas for checking its behavior functional techniques are more appropriate.

Models used

Since testing techniques are based on models, the models available (i.e. developed and used during the specification, design and implementation of the system) will to some degree govern which testing techniques can be used. For example, if the specification contains a state transition diagram, state transition testing would be a good technique to use.

Tester knowledge/experience

Tester's knowledge about the system and about testing techniques will clearly influence their choice of testing techniques. This knowledge will in itself be influenced by their experience of testing and of the system under test.

Likely defects

Knowledge of the likely defects will be very helpful in choosing testing techniques (since each technique is good at finding a particular type of defect). This knowledge could be

gained through experience of testing a previous version of the system and previous levels of testing on the current version. What types of defects do the artifacts contain? There's a big difference between grammatical errors in code and missing requirements in a requirements specification. Testing and inspection methods might be better or worse for different types of defects. We classify defects on the basis of their origin: requirements, design, or code. Second, many empirical studies categorize defects along two dimensions. The first dimension classifies defects as either an omission (something is missing) or a commission (something is incorrect), while the second dimension defines defect classes according to their technical content [16, 17], other classifications focus on the defect's severity in terms of its impact for the user: unimportant, important, or crucial [18].

Test objective

If the test objective is simply to gain confidence that the software will cope with typical operational tasks then routine techniques can be employed. If the objective is for very thorough testing or high reliability (e.g. for safety-critical systems) then more rigorous and detailed techniques should be selected.

Documentation

Whether or not documentation (e.g. a requirements specification or design specification) exists and whether or not it is made up to date will affect the choice of testing techniques. The content and style of the documentation will also influence the choice of techniques.

Evaluation criteria

What are the criteria for selecting techniques? Should you choose the most effective or the most efficient method? Efficiency in this context means the number of defects found per time unit spent on verification, and effectiveness means the share of the existing defects found.

Risk

The greater the risk (e.g. safety-critical systems), the greater the need for more thorough and more formal testing. Commercial risk may be influenced by quality issues (so more thorough testing would be appropriate) or by time-to market issues (so exploratory testing would be a more appropriate choice).

Customer/contractual requisites

A contract usually specifies the main objective of the system. Possible objectives can be performance, dependability etc. Accordingly we will choose a testing technique to meet that objective.

Regulatory requirements

Some industries have regulatory standards or guidelines that govern the testing techniques used. For example, the aircraft industry requires the use of equivalence partitioning, boundary value analysis and state transition testing for high integrity systems together with statement, decision or modified condition decision coverage depending on the level of software integrity required.

Time and budget

Ultimately how much time there is available will always affect the choice of testing techniques. When more time is available we can afford to select more techniques and when time is severely limited we will be limited to those that we know have a good chance of helping us find just the most important defects.

CONCLUSION

Testing techniques are valuable. We have to select testing techniques anyway, so why not do it systematically. The benefit is: If somebody asks how you did it, you are able to describe it, plus your reasoning behind it. Your test will be accountable. And you may be able to improve over time. Perhaps the single most important thing to understand is that the best testing technique is no single testing technique. Because each testing technique is good at finding one specific class of defect, using just one technique will help ensure that many (perhaps most but not all) defects of that particular class are found. Unfortunately, it may also help to ensure that many defects of other classes are missed! Using a variety of techniques will therefore help ensure that a variety of defects are found, resulting in more effective testing. The factors listed above show that many variations must be taken into account for selection of software testing techniques.

FUTURE SCOPE

The factors listed can be further explored in the software testing industry to determine if (i) any factor is missing or (ii) any factor defined is not essential.

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."*, 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) Volume 6– No.9, September 2010.

[4] Hamlet, R.,"*Theoretical Comparison of Testing Methods*". In Proceedings of the ACM SIGSOFT '89 Third Symposium on Testing, Analysis and Verification. Pages 28-37, Key West, Florida, ACM.

[5] Natalia Juristo, Ana M. Morena, Sira Vegas," *State of the Empirical Knowledge on Testing Techniques*",

[6] A. Bertolino,"*Guide to the knowledge area of software testing"*. Software Engineering Body of Knowledge, February 2001.

[7] Whitney Quesenbery, *"Choosing the Right Usability Technique*", www.WQusability.com

[8] S. Vegas, "*Identifying the Relevant Information for Software Testing Technique Selection*", isese, pp.39-48, 2004 International Symposium on Empirical Software Engineering (ISESE'04), 2004.

[9] Jaymie Strecker, Atif M Memon, "*Faults' context matters*", ACM New York, NY, USA. Pages: 112 - 115,2007, ISBN:978-1-59593-724-7

[10] 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).

[11] P. G. Frankl and E. J. Weyuker, "*A formal analysis of the fault-detecting ability of testing methods*", IEEE Transactions on Software Engineering 19(3):202–213 (1993).

[12] S. Ntafos, "*A comparison of some structural testing strategies."* IEEE transactions on software engineering, 14(6):368-874, June 1988

[13] S.N. Weiss. "*Methods of comparing test data adequacy criteria*." In COMPSAC'90. Proceedings of the 14th Annual International Computer Software and Application Conference, pages 1-6, Chicago, Illinois, October 1990.

[14] V.R. Basili and H.D. Rombach. "*Support for Comprehensive reuse*." Software Engineering Journal, pages 303-316, September 1991.

[15] Graham, D. and Van Veenendaal, E. "*Foundations of Software Testing: ISTQB Certificatio*n", Cengage Learning Emea, 2008.

[16] V.R. Basili and R. Selby, "*Comparing the Effectiveness of Software Testing Strategies*" IEEE Trans. Software Eng., Dec. 1987, pp. 1278–1296.

[17] T. Berling and T. Thelin, "*An Industrial Case Study of the Verification and Validation Activities*" Proc. 9th Int'l Software Metrics Symp, IEEE CS Press, 2003, pp. 226–238.

[18] T. Thelin, P. Runeson, and C. Wohlin, "*Prioritized Use Cases as a Vehicle for Software Inspections*" IEEE Software, vol. 20, no. 4, 2003, pp. 30–33.