Improving Software Consistency by Evaluating Efficiency of Software Testing Techniques

Chanaveni Chaitanya
Asst Professor
Department of CSE,
Siddhartha Institute of technology & Sciences, Narapalli, TS, India

ABSTRACT

We have a great number of software testing techniques at our disposal for testing a software product. Although the utilization of these techniques is growing, we do have a very inadequate knowledge about their relative quantitative and qualitative statistics. The choice of a software testing technique in software testing influences both process and product quality. So it is imperative for us to find a testing technique which is effective as well as efficient. However it is not sufficient if testing techniques are only compared on fault detecting ability. They should also be evaluated to check which among them enhances reliability most. To establish a useful theory for testing, we need to evaluate existing and novel testing techniques not only for effectiveness and efficiency but also for their ability of enhancing software reliability.

Key words: *Effectiveness Evaluation; Software Testing; Software Testing Techniques; Software Reliability*

1. INTRODUCTION

Software testing constitutes a major part of software development lifecycle. Lack of testing has resulted in many software related problems in past, and have actually brought social problems and financial losses. Despite all the efforts people put in the quality of the software, effectiveness of testing remains lower than expectations. By one estimate USA incur approximately US$50B in losses from defective software each year [1]. This estimate suggests industry-wide deficiency in testing. According to [2] testing is a widespread validation approach in industry, but it is still largely ad hoc, expensive, and unpredictably effective. So a solution to this problem is to test a system completely. However we are often faced with lack of time and resources, which can limit our ability to effectively complete testing efforts, thus ruling out exhaustive testing. Therefore we have to select proper and effective testing techniques which will increase test effectiveness to maximum extent that too very efficiently keeping in view the limited resources available for testing. Unfortunately, it is not known which testing technique to select as we do not have adequate information about relative effectiveness, efficiency and cost of testing techniques. To obtain this kind of information is not easy given the variability of their operation, which depends on the subject that applies it, the programming language, software under test, the type of faults etc. Some advances have been made in evaluating effectiveness, efficiency of testing techniques but there is still a long way to go as results are very inconclusive. So we should further evaluate and compare software testing techniques in a way that would yield fruitful results. However, one important thing to note is that most experiments conducted so far are focused only on evaluating software testing technique's fault finding effectiveness and efficiency and not much attention is paid to evaluate software testing techniques on software reliability improvement criterion as producing dependable software is also one of the main objectives of software testing. So there is a need to evaluate software testing techniques not only for effectiveness and efficiency of finding faults but also for the ability of enhancing software reliability.

2. SOFTWARE TESTING AND RELIABILITY

Software testing and software reliability have traditionally belonged to two separate communities. However at present there is a strong bond between software testing and software reliability. An important aspect of testing is to make quality and its characteristics visible which include the reliability of the software. The reliability attribute is not directly measurable and must therefore be derived from other measurements such as failure data collected during testing. Software testing is an effective method for estimating the present reliability and predicting future reliability and also to improve it. The difficulty of the reliability attribute is that it only has a meaning if it is related to a specific user of the system. Different users

experience different reliability, because they use the system in different ways. If we are to estimate, predict or certify the reliability, we must relate this to the usage of the system. One way of relating the reliability to the usage is to apply usage-based testing [3].

On the other hand software reliability can be used to measure how much progress has been made in system-level testing [4] [5]. The size of a maintenance work can be determined by the amount of system reliability that can be sacrificed for a while [6]. The concept of reliability also allows us to quantify the failure-related quality aspect of a software system. Quantification of the quality aspect of software systems gives developers and managers a better insight into the process of software development. In future, it is important to bring these two groups more closer, so that on one hand, software testing can be effectively conducted, while on the other hand, software reliability can be accurately measured and improved.

3. CORRECTNESS TESTING VS. RELIABILITY TESTING

Software testing is a generic term which includes attempting to verify and improve all aspects of software quality. Software testing techniques serves multiple purposes in software testing life cycle. Figure 1 shows the test information flow reflecting different purposes of software testing. Correctness is the minimum requirement of software; therefore determination of correctness of final program with respect to its requirements is the essential purpose of testing. Most common approaches of testing are focused on finding faults as much as possible. Thus, the main objective is fault detection i.e. correctness testing. Software testers test software without referring to how software will operate in the field, as often the environment cannot be fully represented in the laboratory [7]. As a result they spend more time trying to break the software than conducting normal operations and mostly design test cases for exceptional and boundary conditions only. Fault detection does not necessarily inspire confidence. [8] States one important goal of software testing should be to measure the dependability of tested software and also to increase it. If failures are more important than faults, the goal pursued during the test phase may also change. Alternatively, another main objective of testing could be to increase our confidence in failure-free operation of the software i.e. reliability testing. In that case, we will not pursue the discovery of as many faults as possible but will strive for a high reliability. In order to obtain confidence in the daily operation of a software system, we have to mimic that situation. Software should be tested according to its operational profile (that represent typical usage scenarios) in order to allow accurate reliability estimation and prediction. In reliability testing we are not so much interested in the faults themselves, but rather in their manifestations. A fault which frequently shows itself will in general cause more damage than a fault which seldom shows up. Reliability testing will naturally tend to uncover earlier those failures that are most likely in actual operation, thus directing efforts at fixing the most important faults [9]. So our aim should be to focus our testing efforts on both types of testing so as to develop software which will be correct as well as reliable. The fault-finding effectiveness of a correctness testing method hinges on whether the tester's assumptions about faults represent reality; for reliability testing to deliver on its promise of better use of resources, it is necessary for the testing profile to be truly representative of operational use.

4. SOFTWARE TESTING TECHNIQUES

Software testing is a process of verifying and validating that a software application or program meets the business and technical requirements that guided its design and development and works as expected and also identifies important errors or flaws categorized as per the severity level in the application that must be fixed [10]. We test software by selecting appropriate testing technique and applying them systematically. Software testing techniques are diverse methods to do software testing. Testing techniques refer to different methods of testing particular features a computer program, system or product [11]. We have to make sure that we select technique(s) that will help to ensure the most efficient and effective testing of the system [12]. 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. However, the fundamental question is what would be the techniques that we should adopt for an efficient and effective testing. So is there a need to evaluate software testing techniques?

5. WHY TO EVALUATE SOFTWARE TESTING TECHNIQUES

The evolution of definition and targets of software testing has directed the research on testing techniques. A few articles [2, 13, 14, 15] have discussed about the future research developments in software testing. A lot of research has addressed the evaluation of the relative effectiveness of the various test criteria, and especially of the factors which make one technique better than another at fault finding. "Demonstrating effectiveness of testing techniques" was in fact identified as a fundamental research challenge in FOSE2000 and FOSE2007, and still today this objective calls for further research, whereby the emphasis is now on empirical assessment [2]. Additional research is needed to provide analytical, statistical, or empirical evidence of the effectiveness of the test-selection criteria in revealing faults, in order to understand the classes of faults for which the criteria are useful and other allied aspects. 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. The aim for the tester is not 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 [11]. At present we have multitude of software testing techniques, all of them can reveal faults; but how effectively they do that and what kind of faults they find, how much resources they utilize, by which factor they increase the reliability. Unfortunately, we do not have answer to all such questions till date. So it is quite evident that test techniques are one element of evaluation, we need to know how to demonstrate the effectiveness of testing methods, how much effective are testing techniques in terms of effort and defect finding capability as we always want to select a testing technique that will bring the product to an acceptable level.

6. IS RELIABILITY BEING LEFT OUT?

A quantitative measure that is useful in assessing the quality of software is its reliability [16]. However, most studies carried out so far which evaluate testing techniques are mostly focused towards fault finding ability of the testing techniques. Most previous evaluations have considered all failures to be equivalent to one another; in reality this is not the case. Fault finding ability measure is useful for evaluating testing techniques when the goal of testing is to gain confidence that the program is free from faults. It is not necessary that a software testing method that can find more faults than others can get higher reliability. Testing to find faults may be more effective (provided the intuitions that drive it are realistic), but if it uncovers failures that appear very insignificantly during actual operation, test efforts will be end up in insignificantly improving the software which means wastage of test efforts and resources. If the goal of testing is to improve the reliability of the program then the measure of test effectiveness must distinguish between those faults that are likely to cause failures and those that are unlikely to do so [17]. As testing techniques will be used for this purpose also, it will be interesting to evaluate effectiveness of different testing techniques for reliability enhancement criterion. In other words, testing techniques should be checked for their effectiveness in exploring failures types, and thus yielding reliability increase.

We can use software reliability as a criterion for evaluating techniques in terms of its effectiveness to produce reliable software. For example, consider two testing techniques T1 and T2 which are applied on a software S separately. Then by monitoring the reliability level of the software, we can observe which technique is more effective in producing software systems of higher reliability. We would like to be able to compare testing strategies in a way that allows us to say that if a system has been tested using technique T1, it is likely to have less risk (more reliable) associated with its use than if it has been tested using technique T2.

Many initiatives have been taken in past to compare testing techniques this way. Frankl, Hamlet, Littlewood and Strigini introduced different approach for comparing testing techniques; the notion of delivered reliability as a means of comparing testing techniques [18]. Delivered reliability describes the idea that for a given program, requirement, and operational distribution, test cases produced according to a particular testing technique may or may not detect various faults and that correction of those faults that are detected by that testing technique will increase the reliability of the program by an amount governed by the operational distribution. Since different test cases produced according to the particular testing technique will detect different sets of faults, different improvements in reliability can result from applying different testing technique. A probabilistic notion of delivered reliability is used to account for this [17]. Therefore, instead of basing the comparison on the probability of finding one or more faults or failures, or the number of faults detected, their goal is to base their assessment of a testing criterion on the reliability of the program under test after it has been tested using a given strategy. Although the intuition on which this assessment is based is interesting, there is still a long way to go ahead, as a lot of work is needed in this direction.

7. WHAT MAY BE THE WAY OUT?

Two types of studies can be carried out to evaluate the relative effectiveness of software testing techniques and provide information for selecting among them; analytical or empirical. An analytical solution would describe conditions under which one technique is guaranteed to be more effective than another, or describe in statistical terms relative effectiveness of software testing techniques. Analytical studies can produce more generalized results, i.e., results which are not tied to a particular experimental perspective. However, analytical studies remain so far quite theoretical in nature: they are highly useful to enlarge our knowledge behind testing techniques but provide little practical guidance in selecting a test technique. The reason is that the conclusions provided by such analytical comparisons are based on assumptions that are far beyond what one can reasonably expect to know or even hypothesize about a program under test [19].

Empirical solutions are closer to a practitioner's mindset in which measures from the observed experimental outcomes are taken. An empirical solution would be based on extensive studies of the effectiveness of different testing techniques in industrial practice, including controlled studies to determine whether the relative effectiveness of different testing methods depends on the software type, subject who

test it, the type of faults in the software, the kind of organization in which the software is tested, and a myriad of other potential confounding factors [20]. However, empirical evidence available falls short of providing such clear-cut answers. Empirical approaches to measuring and comparing effectiveness of testing techniques are still at its infancy. A major open problem is to determine when, and to what extent, the results of an empirical assessment can be expected to generalize beyond the particular programs and test suites used in the investigation. Easier to say than to do, empirical comparisons of test techniques are very difficult and expensive for a series of obvious reasons. Empirical studies are quite challenging as several uncertainties against both the generality of the experimental results, and the over simplification of the experiment settings can be raised after an experiment is concluded.

However, at present the trend in research is toward empirical, rather than theoretical, comparison of the effectiveness of testing techniques. Directly observed results can sound more realistic and convincing than mathematical formulas built on top of theoretical assumptions. While empirical studies have to a large extent displaced theoretical investigation of test effectiveness, in the longer run useful empirical investigation may require its own theoretical framework. We need both analytical and empirical research to close the circle: analytical studies give us hints on which conditions make a test technique more convenient than another, and by means of empirical studies we then can check when it is that such conditions are met for real programs. Such kind of empirical studies could also be used to put the conclusions from analytical comparisons within a practical context.

8. WHY FURTHER RESEARCH IS NEEDED?

A lot of studies have been conducted to evaluate different testing techniques in past years; but there are no clear-cut results yet. The complete listing and summary can found in [21, 22]. The authors have also reached the same conclusion after studying various experiments on software testing. Also, they found that it is really difficult to compare different experiments; however, they do not present any solution to it.

Studies carried out so far do not examine the conditions of applicability of a technique at length or assess the relevant attributes for each technique. There is no study which can tell us about the relative effectiveness of testing techniques because of the difference between parameters they have taken into consideration and the results also, do not reveal much information and show a lot of contradiction. Comparison criterion for testing techniques is usually not well defined. Most of the studies do not take all the parameters necessary for comparison into consideration, as a result of that one technique do not supersede other techniques on all fronts; thereby creating ambiguity in test technique selection.

The big question still remains there: Which are the techniques which is most effective and efficient. At present, we are still without an answer. Presently, we select testing techniques neither systematically, nor following well-defined guidelines. Some testing techniques are never considered for use at all and others are used over again in different software projects without even examining, after use, whether or not they were really suited [23]. Current studies suggest it is just not sufficient to rely on a single method for catching all errors or problems in a program. Perhaps the single most important thing to understand is that the best testing technique is no single testing technique. Using a combination of software testing techniques will help us to ensure that diverse faults are found, resulting in more effective testing. But how long will we use numerous techniques to carry out testing. Going this way means excessive use of resources (less efficiency), as using many testing techniques clearly implies more test cases, more time and more resources. So it is argued that more experimental efforts are required to find out testing techniques which will make our testing both effective and efficient. So there is a need to review and evaluate software testing techniques to compare their effectiveness but the experimentation should be carried out in such a way so that results can be realistic and with very less contradictions. Recent surveys on comparisons of various software testing techniques also concludes that further empirical research in software testing is needed, and that much more replication has to be conducted before general results can be stated [24].

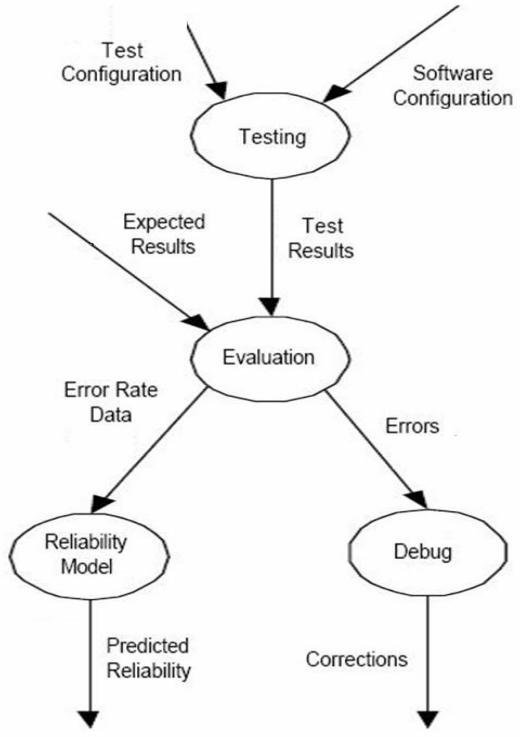

Figure 1: Test Information Flow

9. CONCLUSION AND FUTURE WORK

Presently we are unaware about the relative ordering of software testing techniques and if we are to make software testing more effective by selecting effective testing techniques then we need to place existing software testing techniques at least on an ordinal scale [8]. To do so we need to carry out experimentation on large scale but that needs to in a way that can be compared and will have no contradictions. For that we also need to establish common and standard parameters so that there are little variations in experimentation goals. However the actual research settings of creating reasonable comparative models have not been totally explored. We need a common schema according to which we should carry out our experiments. We also have to find out dimensions on the basis of which we can all agree that if one testing method A is more effective than another testing method. The possible dimensions can be:

1. If technique A finds more faults than technique B.
2. If technique A finds more critical faults than technique B.
3. If technique A produces more reliable software than technique B.

Rather than evaluating testing techniques for fault finding ability only, we should take other factors into consideration also like fault severity, cost, efficiency etc. We should also evaluate techniques for reliability dimension. As we know it is difficult to make meaningful comparison of testing techniques. We cannot really expect one testing technique to supersede all other techniques. There is often a specific interest or purpose of evaluating a particular test technique, based on the assumption that the technique will be more effective. Regardless of our technique, it could be wise to try to understand what types of failures and faults a particular technique can be expected to find and at what cost. We have to check whether testing technique effectiveness and efficiency depends on program to which it is applied, subject who applies it, the number of faults in the program or the type of faults in the program

A lot of such questions need to be answered. Present situation call for replication and further work on evaluation of software testing techniques so as to acquire the basic knowledge about the relative effectiveness and efficiency of software testing techniques for both fault finding and reliability criterion. Most of the research that has been performed is very academic and not very useful in the real testing world. We should validate the software testing technique effectiveness and efficiency for real world software, and then only we can put these testing techniques into practice.

REFERENCES

[1] NIST-Final Report "The Economic Impacts of Inadequate Infrastructure for Software Testing", *Table 8-1, National Institute of Standards and Technology, May 2002.*

[2] Bertolino, A. (2007). , "Software testing research: Achievements, challenges, dreams.", *In FOSE '07: 2007 Future of Software Engineering, pages 85–103, Washington, DC, USA. IEEE Computer Society.*

[3] C. Wohlin, M. Höst, P. Runeson and A. Wesslén, "Software Reliability", in Encyclopedia of Physical Sciences and Technology (third edition), Vol. 15, *Academic Press, 2001*

[4] S. Dalal and C. Mallows, "When Should One Stop Testing Software", *Journal of the American Statistical Associations*, Vol. 81, 1988, pp. 872–879.

[5] M. C. K. Yang and A. Chao, "Reliability-Estimation and Stopping-Rules for Software Testing, Based on Repeated Appearances of Bugs", *IEEE Transactions on Reliability*, June 1995, pp. 315–321.

[6] Kshirasagar Naik, Priyadarshi Tripathy, "Software testing and quality assurance: theory and practice", *John Wiley & Sons*, 2008.

[7] M. R. Lyu, "Software Reliability Eng : A Roadmap," *Intl. Conf. on* Software Eng. (FOSE '07), May 2007, pp. 153-170.

[8] Sheikh Umar Farooq and S.M.K. Quadri, "Effectiveness of Software Testing Techniques on a Measurement Scale", *Oriental Journal of Computer Science & Technology*, Vol. 3(1), 109-113 (2010).

[9] P. Frankl, R. Hamlet, B. Littlewood, and L. Strigini. Choosing a Testing Method to Deliver Reliability. In *International Conference on Software Engineering, pages* 68–78, 1997.

[10] 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.

[11] S. M. K. Quadri and Sheikh Umar Farooq, "Testing Techniques Selection: A Systematic Approach", *Proceedings of the 5th National Conference;* INDIACom-2011, pp-279-281, March 10 – 11, 2011

[12] Sheikh Umar Farooq, and SMK Quadri, "Identifying some problems with selection of software testing techniques", Oriental Journal of Computer Science & Technology Vol. 3(2), 266-269 (2010)

[13] Harrold, M. J. (2000). Testing: A roadmap. In ICSE '00: Proceedings of the Conference on The Future of Software Engineering, pages 61–72, New York, NY, USA. ACM Press.

[14] Abran, A., Bourque, P., Dupuis, R., and Moore, J.W., editors (2004). Guide to the Software Engineering Body of Knowledge - SWEBOK. IEEE Press, Piscataway, NJ, USA.

[15] Taipale, O., Smolander, K., and Kälviäinen, H. (2005), "Finding and ranking research directions for software testing". In EuroSPI'2005: 12th European Conference on Software Process Improvement, pages 39–48. Springer.

[16] M.R. Lyu (ed.), Handbook of Software Reliability Engineering, IEEE Computer Society Press and McGraw- Hill, 1996.

[17] P.G. Frankl and Y. Deng, "Comparison of Delivered Reliability of Branch, Data Flow and Operational Testing: A Case Study," Proc. ACM Int'l Symp. Software Testing and Analysis, ACM Press, 2000, pp. 124–131.

[18] P. Frankl, D. Hamlet, B. Littlewood and L. Strigini, "Evaluating testing methods by delivered reliability", IEEE Transactions on Software Engineering, 24 (8), pp. 586-601, 1998.

[19] Bertolino, A.: The (Im)maturity Level of Software Testing. SIGSOFT Softw. Eng. Notes 29(5), 1–4 (2004)

[20] Michal Young, "Software Testing and Analysis: Process, Principles, and Techniques", John Wiley & Sons, 28-Jul-2008

[21] Juristo N, Moreno AM, Vegas S (2002), "A survey on testing technique empirical studies: how limited is our knowledge?", 1st Int Symp Empir Softw Eng, pp 161–172

[22] N. Juristo, A.M. Moreno, S. Vegas, M. Solari, "In search of what we experimentally know about unit testing", IEEE Software 23 (6) (2006) 72–80.

[23] 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.

[24] Juristo, N., Moreno, A.M., and Vegas, S. Reviewing 25 Years of Testing Technique Experiments, Empirical Softw. Eng. J., 9, 1/2 (March 2004), 7-44.