Open Problems in Software Test Coverage

Dalal Z. Alrmuny

Abstract—Software testing is concerned with the production and maintenance of high quality systems. Test coverage is one major component in software testing. It gives a meaningful description of how much testing is required. The level of test coverage is an indicator of testing thoroughness, which in turn helps testers decide when to stop testing. Several studies addressed coverage criteria effectiveness and adequacy. However, there are several interesting open problems. We identify nine open problems in test coverage and discuss what issues need to be addressed. The objective of this paper is to direct the attention of researchers and testers toward the areas that need to be investigated.

Index Terms—Software testing, test coverage, coverage criteria.

I. INTRODUCTION

Software testing is concerned with the production and maintenance of high quality systems. Test coverage is one major component of software testing, as it gives a meaningful description of how much testing is required for adequacy. The objective of test coverage is to decide when to stop testing, which is one major unanswered critical questions in the testing discipline.

We identified a set of open problems and challenges that face researchers and practitioners in the testing discipline specific to software test coverage by looking into the existing published work. The objective of the identification of the open problems is to direct the attention of the testing community to the areas that need to further investigation.

Some of the discussed problems span other issues in software testing other than test coverage. The reason behind that it is hard to isolate test coverage from some other components. Hence, we had to discuss some open problems that are pertinent to other related issues in the testing process, such as test oracles.

The rest of the paper is organized as follows. Section II provides a background on software testing and test coverage. We discuss the set of identified open problems in test coverage in Section III. Conclusions and recommendations are provided in Section IV.

II. BACKGROUND

A. Software Testing

Nowadays, it is not deniable that software testing is an

essential activity in the development process to produce reliable software. Moreover, it is expected that software testing will remain in the future as one of the best available tools that we have to ensure software reliability [1].

Unfortunately, Software testing has been described as difficult, time consuming and inadequate [1], [2]. Several testing techniques are available now but there is no consensus on the absolute or relative applicability or usability of these techniques. Testing techniques are inherently heuristic as their application does not guarantee the detection of faults [2]. In addition to that, some researchers characterize software testing as tricky; as the evaluation of some testing techniques generated counter-intuitive results [3].

Practitioners in the testing field are challenged by the lack of sufficient knowledge that can guide them in the selection among the different testing techniques [4]. Software testing does not meet the maturity level it is expected to have as an engineering discipline [4], [5]. Due to the lack of proper fact-based guidance, developers tend to rely on intuition, fashion or market-speak [4].

Several studies have been carried out to evaluate and compare the different testing techniques [6]-[9]. Many of these studies suffer from validity issues that make their results questionable. One vital question that is still pending an answer from researchers and practitioners is “which testing technique to use and when to use it?” Several researchers pointed out that more research is needed to advance the maturity level of software testing [2], [5], [10]. Integration of the existing techniques and test automation are two areas that need more attention [2]. Bertolino suggests having a broader perspective on software testing that is concerned with generating complete test solutions instead of focusing on the narrow problem of test selection [5].

B. Test Coverage

Test coverage has gained the attention of researchers as one mean of assuring the adequacy of the testing process. Test coverage is used to measure of how much a program has been exercised with tests. The level of test coverage is an indicator of testing thoroughness, which in turn helps testers decide when to stop testing. Hence, several coverage criteria have been developed in the past years.

The immaturity of software testing as an engineering discipline stands as a major obstacle in the way of adopting test coverage criteria in the industrial field. There is lots of necessary knowledge that practitioners, like managers and testers, need to know to be able to use coverage criteria in an efficient and effective manner. Taking a look at which testing techniques and coverage criteria are being used in industry indicates the large gap between industry and research community [5], [10], [11]. The efforts of testing community are focused on generating more coverage criteria. Many of

Manuscript received May 21, 2013; revised September 11, 2013.

Dalal Z. Alrmuny is with the Department of Computer Science, Colorado State University, Fort Collins, CO 80523 USA (e-mail: alrmuny@cs.colostate.edu).

which were not subject to sufficient effectiveness evaluation and context identification. The available criteria at our disposal need to be studied and compared in relative to one another in more details [11].

To help testers decide which coverage criteria to use and when, several evaluation studies have been carried out by researchers [12]-[15] and more studies are being carried out at the moment. Despite the importance of these studies in shedding a light on the importance of analyzing coverage criteria, neither of these studies provides a final answer on which criteria to use in some certain context. It is understandable that no one single study is capable of answering the question of what criteria to use and when, but a series of well-designed studies can. The scattered efforts of researchers, if gathered, and using a properly designed study can provide an acceptable answer to this question.

We encourage the testing community to focus their efforts toward establishing a foundation for the usability of the existing criteria and their useful extensions. The major benefit of doing that is to give practitioners enough knowledge on which criteria is the best-fit in their development context. Also, this will meet the demands of testing software in newly evolved domains. Ammann and Offutt express similar opinion on this issue [11]. The purpose of this paper is to identify and address the set of open problems that we believe are important to be addressed by the research community.

III. OPEN PROBLEMS IN TEST COVERAGE

Several open problems stand as obstacles in the way of using and applying test coverage in an efficient and effective manner. Based on our survey of existing literature, we identified nine open problems presented in the following subsections.

A. The Gap between Theory and Practice

Currently, literature in test coverage is rich with many research papers that propose new criteria or evaluate and compare some existing criteria [6], [12]-[15]. In addition to that, some work has been done to study the possible combinations of coverage criteria in an effective manner [8].

Theoretical work is well-appreciated as it contributes to creating a solid base to the test the coverage field [10]. Alternatives to theoretical studies are analytical and empirical studies. Analytical studies help identify the suitable context of each coverage criteria, while empirical studies help evaluating the practical use of the coverage criteria [5]. Both theoretical and analytical results are less practical than empirical results [5].

Unfortunately, a gap between theory of coverage and industry still exists. What researchers need to focus on today is how to overcome the barriers to adopting existing coverage criteria in industry [5], [16]. For the industry to adopt a certain criteria, they need a guideline to the selection among the different criteria that is based on facts. The best candidate to overcome this gap is empirical studies as they are closer to practitioner's mindset [5]. The execution of an empirical study is can be viewed as a smaller scale of what takes place in the industrial context. Nevertheless, the value of the

empirical study to the industry is highly-tied to the design and settings of the study. Generally, directly observed results are easier to understand and interpret by industry than mathematical formulas generated from pure theoretical studies. Moreover, empirical studies are viewed as a necessary mean to advance the maturity level of the "testing" discipline [5].

Two major obstacles prevent empirical studies from matching the industrial settings, namely the high cost and the unavailability of industrial data. For example, obtaining real fault data from industry is hindered by two facts. First, the industrial owner of the data is hesitant to share this data as it falls under the "private" classification. Second, fault data in many real systems can be insufficient statistically, which poses a threat to the validity of the results of the empirical study. Briand and Labiche suggest starting with academic settings, and based on the results, the decision can be made whether to apply in industrial settings or not [17].

B. Undetermined Effectiveness of Coverage Criteria

Despite the long time that has elapsed since many coverage criteria have been proposed, there is still no proper means of accurately measuring the effectiveness of different criteria [2], [15]. Although several researchers addressed the evaluation of effectiveness of coverage criteria and comparing them relatively, the results are not sufficient to guide practitioners in the industrial field [18].

A major contribution to solving this open problem was achieved by establishing the subsume hierarchy between comparable criteria. The subsume relation indicates the relative thoroughness between the subsumed and the subsuming criteria, which can help in the selection of the "most" thorough criteria [10], [19].

Nevertheless, the subsume relation is not the only factor that matters in the selection of coverage criteria. The cost, when moving up in the subsumption hierarchy, is an important factor in the adoption of a criterion. What matters is how much more cost we need to pay to achieve coverage of a certain criterion with a higher level of subsumption [11].

Other factors to consider are time; how fast a criterion is in revealing faults, and fault types; what classes of faults a certain criteria reveals better than other criteria [3], [10]. The relative effectiveness of non-comparable criteria using the subsumption relation is another issue that needs to be resolved. In conclusion, additional research is needed to provide scientific evidence of the effectiveness of coverage criteria [4], [10], [11], [15].

C. Dependency on the Context

To address the question of which coverage criteria is more effectiveness than others, several analytical and empirical studies have been performed [7], [8], [10], [12]-[16]. One remarkable finding of several empirical studies is that the behavior of the coverage criteria is dependent on the context in which it is studied [7], [8], [10], [13], [14], [16]. The context of the study can be defined in terms of the characteristics of the study components. One component is the studied system(s) that can be characterized by the system structure and the types of existing faults.

This implies that we need to look for test patterns rather than absolute effectiveness of coverage criteria. A good

description of a test pattern is presented by Bertolino, where she defines a test pattern as a 3-part rule that expresses a relation between a certain context, a problem, and a solution [5]. Unfortunately, the results of the different studies that address the effectiveness of coverage criteria are not expressed in terms of the context in which the study was performed, but rather researchers attempt to discuss the findings as context-independent. This explains why the validity of the empirical studies in this field is usually being questioned. Isolating the findings from the context degrades from the value of the study. This is because the findings may not hold in other different contexts.

D. The Lack of Well-Formed Empirical Studies

As we mentioned before, empirical studies are the best candidate to evaluate the effectiveness of testing techniques [1], [3], [5], [13]. Empirical studies in testing are characterized by being very difficult and expensive in terms of time consumption and labor work [5], [17]. Also, empirical studies specific to testing suffer from several limitations that affect the validity of their results [5], [13]. Considering a study of coverage criteria effectiveness, for example, several factors come to play that are hard to control and, at the same time, can divert the results of the study in different directions. We list here some of these factors:

- System(s) under study: in the ideal case, the system under study has to be representative of real industrial systems, which usually are not available for researchers to experiment with. Some alternative options to industrial systems are academically-developed systems and open source systems. The selected systems can vary in their size and domain. Due to the increasing effort associated with larger systems, the size is one limitation to empirical studies [5], [13]
- Fault data: the type of faults in the studied system is considered one of the main factors to affect the effectiveness of a coverage criterion [10], [11]. Based on its availability, fault data can be obtained from the reported bugs log or can be seeded in some way. Since many real systems do not have large enough number of reported bugs, researchers use fault seeding as an alternative to obtain fault data, which may not be representative of real faults. Fault seeding poses a threat to external validity of the observed results [17].
- Test case generation: in case of the absence of automatic test case generation, manual generation is the alternative option [13]. The efficiency of the studied coverage criteria will be dependent on the selected way of test case generation. The quantification of the effort required to generate the test cases is most likely impossible in the case of manual generation.
- The human factor: the difference in the experience and the background of testers may suggest different results of the same study [5], [17]. The selection of test cases, for example, one tester make is based on her experience and are not expected to be similar to what other testers select. This selection can either lead to higher or lower overall effectiveness.

The aforementioned factors distract researchers from carrying an objective assessment in a reproducible testing

process, which is one of the current demanding needs in test coverage [3]. Due to the prohibited cost of more realistic studies and the lack of real systems and fault data, researches tend to simplify the study settings and use surrogate measures. For example, test suite size is commonly used as a surrogate measure for test cost. Some studies in test coverage were criticized for the oversimplification of the experiment settings and the generality of the observed results [5], [13].

E. Software Evolution and Emerging Development Paradigms

The fast pace of the evolution in software development and the emerging new development paradigms pose another challenge to software testing as it has to cope with these changes. Testing techniques need to address the requirements of modern systems that are characterized by a growing complexity. Strategies that work for regression testing in average systems may not scale up in large composite systems [10]. Systems that evolve dynamically have different requirements of testing and test coverage, which makes quality control a real challenge [10]. Also, coverage criteria are needed for aspect-oriented systems, which are getting an increasing attention recently.

The need to have domain-specific testing techniques and domain-specific coverage criteria is emerging [11]. Service-oriented computing is one recent paradigm which has many specific requirements of testing. One example of these special requirements is testing the composition of services as well as each individual service. The challenges of testing web services are driven by the lack of information and control that testers need. Information about the system artifacts is typically not available. Also, testers have no control over service execution [20]. How can we guarantee adequate testing of web services and what coverage criteria is suitable to be used are some questions that need to be answered. Another example is GUI testing where traditional coverage criteria do not work well for GUIs [21]. Little work has been done to define domain-specific coverage criteria and hence more attention needs to be paid to this area.

F. Is a Combination Better?

Coverage criteria are considered to be alternative to each other in the different studies that attempted to compare their relative effectiveness. Recently, several researchers highlighted the fact that coverage criteria can be viewed as complimentary to one another rather than alternative.

The rationale behind this conclusion is that empirical studies show that individual criterion performs better based on certain context. This suggests that two different criteria that target different types of systems or fault types complement each other [10]. The combination of these criteria is expected to outperform either one as more coverage will be achieved. Even the most effective criteria that target certain type(s) of faults will eventually suffer from saturation at certain time [10].

Empirical results supports using a combination of coverage criteria to achieve better coverage and researchers are pointing out the need to investigate the effective combination of coverage criteria and testing techniques [2], [5], [7], [8], [10], [11], [16]. Some studies even suggest that the combination is significantly more effective than the

individual criteria or techniques [7], [8], [11].

In conclusion, no single criterion provides a magic solution and combining the different criteria is necessary to achieve adequate coverage. There is an obvious need to explore the different combinations of coverage to see whether they perform better than using single ones, which requires more empirical studies to be performed.

In order to decide on which criteria to combine for better coverage, we need to first to identify when each individual criterion performs better, then we can use that information to define combinations of criteria to get maximum effectiveness. Unfortunately, researchers jumped to the combination stage before completing the first step of evaluating individual criteria.

G. Test Oracles

A testing technique can be viewed as a coverage criterion and an oracle [1]. The general belief is that the choice of a test oracle is generally orthogonal to that of test criterion. Nevertheless, some recent studies show that test oracles have an impact on the effectiveness of testing. Some oracles tend to be more suitable with certain coverage criteria. For example, state invariants are more suitable as test oracles with state-based test techniques [1].

Memom et al. show that the test oracle used during testing (beside the type and the number of test cases executed on the software) contributed significantly to test effectiveness and cost [22]. Gupta and Jalote claim that the ability of a test case to find a fault in a program is greatly affected by the strength of the test oracle [14]. Also, knowing which test oracles to choose can contribute to easier automation [21]. More studies on the impact of test oracles on test effectiveness are needed as they are currently scarce and in limited domains [23].

H. Test Automation

High cost has been an obstacle in way of adopting several testing techniques and coverage criteria in the industrial field. For example, unit testing is often skipped because it is quite expensive [10]. Test automation is one solution that can reduce the cost of testing and hence make it more feasible.

There is a need to increase the amount of automation in the testing process by finding cost-reduced methods and techniques to perform testing [10].

To improve our knowledge about the different coverage criteria, we need to exercise them in different contexts. Automation can make the process of comparing different criteria achievable under acceptable time and effort requirements. Also, automation can reduce the effect of the confounding factors, like human intervention, and can make the results of empirical studies more objective. Until now, testers lack enough automation in test coverage.

A related issue that is worth mentioning here is the misuse of coverage tools. Coverage tools must be used properly and their results need to be interpreted correctly. Achieving high coverage does not mean that testing is exhaustive [12]. The parts that were not covered in the system under test may be the place where faults with high impact reside.

I. Infeasible Test Requirements

To meet a given coverage criterion, the set of corresponding test requirements need to be satisfied using test data. Test requirements that can't be satisfied by any test

data are called infeasible. Infeasible test requirements have an impact on the actual coverage level that can be achieved using certain coverage criterion. The amount of testing required to satisfy certain level of coverage can't be accurately determined with the existence of infeasible requirements [24].

The detection of infeasible test requirements is not solvable theoretically [24]. Several researchers addressed the problem of infeasible requirements detection. Unfortunately, most of the approaches used are capable of detecting only a limited number of feasible requirements.

The infeasibility issue is an obstacle in the way to test automation as the automatic detection of infeasible paths is generally difficult if not impossible [2]. Efficient techniques that can detect infeasible requirements as early as possible are needed.

IV. CONCLUSIONS AND RECOMMENDATIONS

Test coverage is an important component of software testing field that enables the production of reliable software. The existence of the aforementioned open problems in test coverage highlights how challenging this field can be. On the positive side, they create opportunities for improvement. In the current time, we have a rich database of published work that can be used as the base for new researchers who are interested in addressing one or more test coverage open problems.

In the following we provide a set of recommendations to help in addressing open problems in test coverage.

- One way to overcome the limitations in the existing empirical studies is to move from small-scale scattered studies to large-scale projects that involve different researchers and experts in the testing field. Carrying out cross-cutting comparison studies that are based on well-defined contexts leads to better valid results.
- To reduce the gap between theory and industry, we suggest that the inclusion of industrial partners in the empirical studies will offer more realistic results that are valid in the industrial context. The research community is required to provide a more clear distinction on which criteria works in which context. This implies adjusting the settings of empirical studies to the industrial context as we mentioned before.
- In order to encourage researchers and practitioners to use criteria that are proven to be more effective, there is a need to produce coverage tools that are easy to use in an efficient manner. Whenever more automation is available, it will contribute to bringing the cost of testing down, which is considered a major obstacle in the way of applying adequate testing. We also encourage researchers to focus on more research in domain-specific testing to provide complete exact solutions.

REFERENCES

- [1] L. Briand, "A critical analysis of empirical research in software testing," *First International Symposium on Empirical Software Engineering and Measurement*, 2007.
- [2] W. Adrion, M. Branstad, J. Cherniavsky, and Validation, "Verification and testing of computer software," *Computing Surveys*, vol. 14, no. 2, June 1982.

- [3] B. Meyer, "Seven principles of software testing," *Computer Magazine*, vol. 4, issue 8, August 2008.
- [4] N. Juristo, A. Moreno, and S. Vegas, "Reviewing 25 years of testing technique experiments," *Empirical Software Engineering*, vol. 9, pp. 7-44, 2004.
- [5] A. Bertolino, "The (im)maturity level of software testing," *WERST Proceedings, ACM SIGSOFT Software Engineering Notes*, vol. 29, issue 5, September 2004.
- [6] H. Zhu, "A formal analysis of the subsume relation between software test adequacy criteria," *IEEE Transactions Software Engineering*, April 1996, vol. 22, no. 4.
- [7] P. Frankle and E. Weyuker, "A formal analysis of the fault-detecting ability of testing methods," *IEEE Transactions on Software Engineering*, March 1993, vol. 19, no. 3.
- [8] M. Wood, M. Roper, A. Brooks, and J. Miller, "Comparing and combining software defect detection techniques: a replicated empirical study," in *Proc. the 6th European Conference/ 5th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, New York: Springer, 1997, pp. 262-277.
- [9] V. Basili and R. Selby, "Comparing the effectiveness of software testing strategies," *IEEE Transactions on Software Engineering*, December 1987, vol. 13, no. 12.
- [10] A. Bertolino, "Software testing research: achievements, challenges, dreams," *Future of Software Engineering*, IEEE-CS Press, 2007.
- [11] P. Amman and J. Offutt, *Introduction to Software Testing*, Cambridge University Press, 2008.
- [12] S. Berner, R. Weber, and R. Keller, "Enhancing software testing by judicious use of code coverage information," *ICSE 2007*, pp. 612-620.
- [13] J. Andrews, L. Briand, Y. Labiche, and A. Namin, "Using mutation analysis for assessing and comparing testing coverage criteria," *IEEE Transactions on Software Engineering*, vol. 32 no. 8, pp. 608-624, 2006.
- [14] A. Gupta and P. Jalote, "An approach for experimentally evaluating effectiveness and efficiency of coverage criteria for software testing," *International Journal on Software Tools and Technology Transfer* 10, pp. 145-160, 2008.
- [15] N. Li, U. Praphamontrirong, and J. Offutt, "An experimental comparison of four unit test criteria: mutation, edge-pair, all-uses and prime path coverage," *IEEE Int'l Conference on Software Testing, Verification and Validation Workshops*, 2009.
- [16] A. Bertolino, "Software testing practice and research," in *Proc. the 10th Int. Workshop on Abstract State Machines*, March 2003.
- [17] L. Briand and Y. Labiche, "Empirical studies of software testing techniques: challenges, practical strategies, and future research," *WERST Proceedings / ACM SIGSOFT Software Engineering Notes*, vol. 29, no. 5, pp. 1-3, September 2004.
- [18] B. Zhang *et al.*, "Measurement of effectiveness of software testing," *Fifth International Conference on Machine Vision (ICMV'12), International Society for Optics and Photonics*, 2013.
- [19] E. Weyuker, "The Complexity of Data Flow Criteria for Test Data Selection," *Information Processing Letters*, vol. 19, issue 2, August 1984.
- [20] H. Zhu, "A framework for service-oriented testing of web services," *Computer Software and Applications Conference*, 2006.
- [21] A. Memon, "GUI Testing: Pitfalls and Process," *Computer Magazine*, August 2002.
- [22] A. Memon, I. Banerjee, and A. Nagarajan, "What test oracle should i use for effective gui testing?" in *Proc. the 18th IEEE International Conference on Automated Software Engineering (ASE'03)*, 2003.
- [23] J. A. Whittaker, "What is software testing? and why is it so hard?" *IEEE Software*, vol. 17, no.1, pp.70-79, 2000.
- [24] M. Ngo and H. Tan, "Detecting large number of infeasible paths through recognizing their patterns," in *Proc. the ACM SIGSOFT symposium on The Foundations of Software Engineering*, 2007.

Dalal Alrmuny received her master's degree in computer science from the University of Jordan, Jordan in 2002. She is currently a PhD student at Colorado State University, USA in the Computer Science Department. Her research is focused on object oriented software testability improvement. Mrs. Alrmuny is a recipient of a Fulbright Scholarship in 2006.