

Survey on Test Coverage Analysis in software testing to Improve Quality Assurance and Reliability

¹Reetesh Gupta, ²Neelabh Sao

¹MTech Scholar, ²Assistant professor

¹Computer Science Department,

¹Rungta College of engineering and technology, Bhilai, India

Abstract— Test Coverage is a vital pointer of programming quality and a key piece of programming support. It helps in assessing the feasibility of testing by giving information on diverse coverage things. even though much research exertion has been put on the most proficient method to get possibility data by either code based testing or precondition based testing, very little has been paid to gauge and examine the scope by covering most extreme number of possibility things. This paper gives an analysis of the momentum test coverage explores led by other specialists for test scope in programming testing. By taking a gander at the current methodologies; holes and uncovered estimation of Test Coverage can be observed additional.

Index Term — Code Coverage, Coverage Measurement, Software Testing, Test Coverage, software quality.

I. INTRODUCTION

Programming testing is viewed as now as a vital progress in programming support life cycle. It is a practice frequently used to focus and enhance programming quality. Testing exercises likewise incorporate acquiring the test scope. "Coverage is the degree that a structure has been practiced as a rate of the things being secured. In the event that coverage is not 100%, then more tests may be intended to test those things that were missed and accordingly, expand Coverage"[1]. Test scope can help in observing the superiority of testing, and support in coordinating the test generators to make test case that cover classes and methods that have not been checked before [2].

The yield of coverage estimation can be utilized as a part of a few approaches to enhance the testing methodology. It can additionally furnish the client with data on the status of the check process. It can help to discover openings in the testing, i.e. regions that are not secured [2]. Test scope likewise helps in Regression testing, experiment prioritization, test suite increase and test suite minimization. Albeit much research consideration has been given to test scope estimation and investigation, yet there is a need to cover each of the three granularity levels of test scope things, i.e. fine- grain, medium-grain and course-grain in point of interest.

This paper provides details regarding a study to discover the examination on present methodologies identified with test coverage. The study introduced here is that includes a search for the script to figure out what sorts of studies tending to the orderly assessment analysis have been completed, where they are distributed, in what databases they have been filed, what sorts of results they have evaluated, and in which populaces [3]. This study is pointed at displaying a review of exploration concerning test coverage over the period 2004 – 2014 and recognizing possible holes in exploration about test coverage.

This paper is composed as takes after. Segment II depicts the research method in leading the study. Section III reports the outcome and discourse about study. In segment IV, we have recognized a few threats to legitimacy. At long last area V closes the work.

II. RESEARCH METHOD

Research Objective

The goal of this study is to compress current state of methodologies identified with Test Coverage in view of the add-on principle questions:

- What is the most important research areas in test coverage tended to by different studies?
- What are the current methodologies of test coverage find out?
- What is the quantity of publications every year?

Data Source

This study was directed from May to November 2004. The literature inquiry was completed by looking in a set of preferred storage area. The stores utilized were ACM Digital Library, ScienceDirect, IEEEExplore, SciVerse Scopus, Google Scholar and Springer Link. Despite the fact that the aggregate number of papers was high, the vast majority of the papers were excess. The paper having same title was returned by distinctive vault internet searchers. In this way, we eliminated these research papers.

Exploration or search criteria

Diverse watchwords were characterized as a search string in view of the research questions. These consist of Code coverage, Software Testing, Test Coverage, and Traceability. Utilizing the above decisive words, a search string was made and utilized for different repository web search tools. The year collection 2004 - 2014 was integrated in the search.

III. THE SYSTEMATIC MAPPING STUDY PROCEDURE

The vital procedure steps of our systematic study are demonstrated in Figure 1. Our deliberate survey process begun with characterizing exploration address as expressed in segment II (A). Next step was to characterize the seeking keyword. The keywords are recorded in segment C. Utilizing the keywords, searching was completed on the chosen repository utilizing the given web search tool. A set of consideration and avoidance criteria were characterized for the choice methodology. These criteria were then utilized as a part of the study handle in the wake of searching.

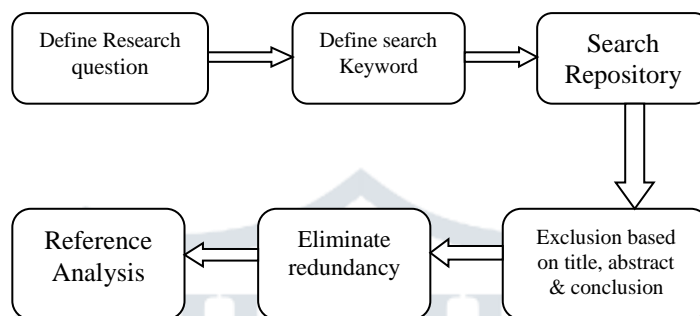


Figure 1 Mapping study process

When the list of exploration papers was gotten, the papers not matching our keywords were expelled after the title, theoretical and conclusion. In the meantime, the papers having same title returned by diverse repository were uprooted to dispose of excess. Last analysis was directed on the base of references to guarantee that referenced papers were not passed up a major opportunity. Reference analysis is vital as a portion of the publications may have been passed up a major opportunity along with the keywords based searching utilizing web engine and at some point in exclusion based on title and abstract. By taking a gander at the reference toward the end of every paper, applicable papers that are missed prior can be integrated in the mapping study.

IV. RESULT AND DISCUSSION

Lots of research papers identified with Test Coverage were returned by this study looking procedure. The research papers were then ordered and arranged into 6 grouped which are review and analysis of test scope, structures for test scope, test coverage things, algorithms/calculations, programming software quality, reliability demonstrating for test coverage, and others. Figure 2 delineates the breakdown of research papers as indicated by the six classes.

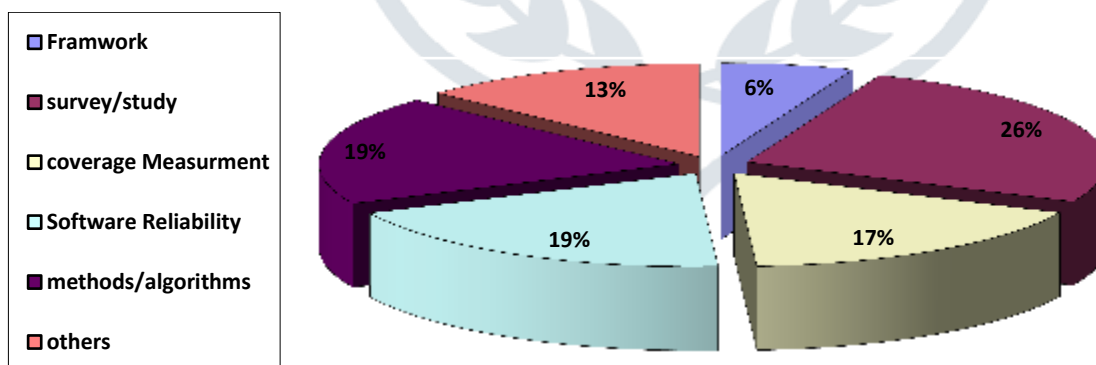


Figure 2 Breakdown of Research Papers Category

In view of Figure 2, coverage estimation, programming dependability issues and studies and overviews are contributing about the equivalent offer. Offer for system advancement is just 6%.

Figure 3 demonstrates the quantity of research publication made every year. Research distributions on test coverage became quickly after year 2004. As of November 2010, there are 5 distributed papers in not long from now. This number may increment, following the literature search was led between May to November 2010 and that's just the beginning.

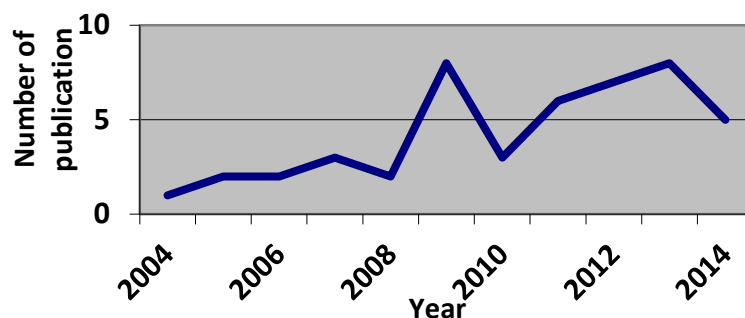


Figure 3 Number of Research Publications per Year

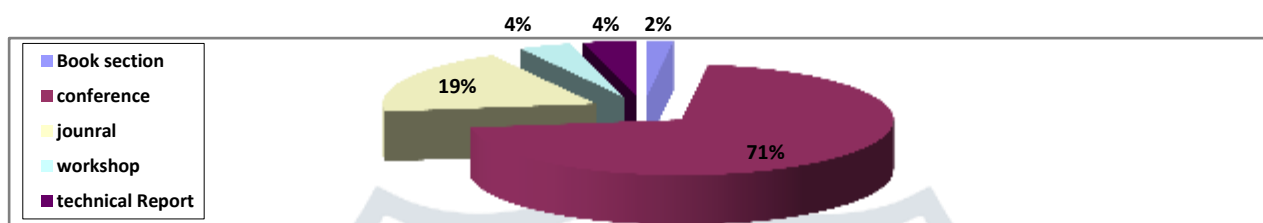


Figure 4 Breakdown of Publication's Forum Type

The publications were additionally arranged in different gathering types. The argument types are workshops, conference and symposiums, diaries, specialized reports and book sections. A substantial rate of the publication originate from meetings and symposiums (70 %), took after by journals (19 %). Workshops and specialized reports imparted 4 % each. This is shown in Figure 4.

The accompanying area will discuss the findings of the study for every focus area as ordered in Figure 2.

Survey/study

There are three study type literature identified with test/code coverage. The scientists concentrated on code coverage as a ceasing rule, analyzed diverse coverage built tools and looked on different technique available for changed methods accessible for producing experiments or test cases to fulfill test coverage criteria.

A few specialists examined test coverage analysis identified with test viability, test suit change and programming software quality estimation. Some accept that visualizations diminished the variability in the quantity of test cases; engineers composed by changing the standard designers used to assess their test adequacy. The whole above specialists are organized in Table 1.

Table1 Survey and study table

Type	Author	Year	Related Work
Survey	Pachawan Augsornsri, Taratip Suwannasart	2014	An Integration Testing Coverage Tool for Object-Oriented Software coverage for object-oriented software
	Pachawan Augsornsri, Taratip Suwannasart	2013	Design of a tool for checking integration testing coverage of object-oriented software
	Ben Smith, Laurie Williams [4]	2008	Code Coverage as a stopping criterion for unit testing
	Qian Yang et al. [5]	2007	Coverage-based testing tools
	Prasanna et al. [6]	2005	Test case generation techniques to satisfy test coverage criteria
Study	Audris Mockus et al.[7]	2009	Test effectiveness
	Silva et al. [8]	2009	Code coverage analysis
	Y. Wei [9]	2008	Coverage as Testing effectiveness
	Anna Derezińska[10]	2008	Test suits improvement
	Jun-Ru et al.[11]	2007	MC / DC coverage criterion
	Stefan Berner et al. [12]	2007	Impact on testing

Framework

Three authors have proposed structures for test coverage estimation and analysis, one by Sakamoto et al. (2010), the other by Matteo Bordin et al. (2009), and third by Misurda et al. (2005). Sakamoto [16] proposed a structure for predictable and adaptable estimation of test coverage, called the Open Code Coverage Framework (OCCF) that backings numerous programming language. Their system gives rules to help a few test coverage criteria. Besides, OCCF let clients increase quality to include user defined test coverage and new programming lang. Matteo Bordin [17] proposed a system.

Couverture, which gives a virtualized execution stage to cross-gathered application on the host machine. Couverture has the capacity measure structural coverage of entity and source code without obliging any type of use instrumentation with a solitary execution of the cross-arranged application and test suites. Misurda, [18] portrayed another versatile and adaptable structure for testing projects with a novel interest driven methodology given that execution path to implement test coverage. He utilized dynamic instrumentation procedure on the binary code that can be embedded and uprooted on-the-fly to keep execution and memory overheads low.

Test Coverage objects or classes

A lots of study papers concentrated on diverse coverage things to gauge and examine test coverage. There are around 12 scope thing sorts like articulation (statement), extension (Branch), piece (block), choice (decision), condition, strategy (Method), class, bundle (Package), requirement, and information flow coverage. In view of the papers accumulated, it can be watched that distinctive scientists have focused on distinctive coverage types yet just two have utilized necessity coverage for test scope analysis. This sort of coverage can be investigated further to look profoundly into scope analysis and estimation. Table 2 orders the methodologies in view of coverage estimation or measurement.

Table 2 Classification of Approaches based on Test Coverage Measurement

Author	Year	Test Coverage objects
Koochakzadeh et al. [19]	2010	Method, Class, Package
Faizah and Suhaimi [20]	2009	Method, Class, Package, Requirement
Angeletti et al. [21]	2009	Branch
Kapfhammer et al. [22]	2008	Code Coverage, Date Flow Coverage
Lingampally et al. [23]	2007	Branch, Block, Method, Predicate
Mehdi et al. [24]	2005	Line, condition, Method
Lormans et al. [25]	2005	Requirement
Diaz et al. [26]	2004	Branch

Methods/ Algorithms

A few analysts grew new techniques, models and algorithms to build and enhance the nature of the code coverage. Tsai et al. (2007) characterized a coverage relationship model for test case determination and positioning for multi-adaptation programming software. He proposed a model based versatile test (MAT) case strategy. Li and Asaf utilized their models to enhance the nature of the coverage analysis. In his study, Gao proposed element test scope analysis answer for screen API-based part approval. James and Mary displayed two new algorithms for test-suite diminishment and one new calculation for test-suite prioritization that can account for MC/DC when reducing and organizing test suites. We have demonstrated these creators, year and the issues they examined in their strategies/algorithms in Table 3.

Table3 Methods/Algorithms

Author	Year	Approaches
Krishnamoorthi et al. [27]	2009	Test case prioritization
Chen et al.[28]	2009	Test data generation for branch coverage
Tsai et al. [29]	2007	Test case selection
J. Jenny Li[30]	2005	Increase code coverage Reduce testing cost
Gao et al. [31]	2005	Test coverage analysis for Component validation
Asaf et al. [32]	2004	Improvement for coverage analysis
James and Mary [33]	2003	Test-suite reduction Test-suite prioritization
Tikir et al. [34]	2002	Dynamic instrumentation
Atif Memon et al. [35]	2001	GUI coverage

Software Reliability Modeling

The study in regards to programming software quality utilizing test coverage has increased much consideration as of late. Various new models have been proposed for programming unwavering quality estimation. A large portion of the analysts accept that time is by all account not the only component that influences the failure behavior of the software product yet other imperative variables must be added to foresee the accuracy of programming software reliability quality models. Test coverage is accepted as a crucial variable to upgrade these models. Table 4 lists down the authors included, year, and issues tended to in building up the software product quality models utilizing test coverage. All studies utilized test coverage with time and different variables to characterize programming software quality reliability models.

Table 4 Software Reliability Modeling Study

Author	Year	Approaches
Jinxia et al. [36]	2010	Integrated test coverage
Smidts et al. [37]	2009	Software fault content and location
Haifeng et al. [38]	2008	Fault detection
Xia et al. [39]] 2007	Estimation improvement
Xia [40]	2006	Failure intensity function
J.-Y. Park et al. [41]	2006	Coverage growth function
H Pham et al. [42]	2003	Integrated test coverage
Malaiya et al. [43]	2002	Relation between testing time, coverage and reliability
M. Chen et al. [44]	2001	Testing time reduction

Others

Other researcher works identifying with test coverage estimation and analysis includes a procedure via Ricardo et al. (2010) [45] for test coverage study of UML state machines which delivers a hued UML model demonstrating the components secured. Rauf et al. (2010) [46] utilized genetic algorithm as a part of MATLAB to investigate automated GUI test suits. Takahashi et al. (2008) [47] utilized suits criteria for simultaneous programming testing. The simultaneous coverage criteria expect to discover simultaneous programming particular defects, for example, race conditions. Gupta and Jalote (2008) [48] utilized transformation (Mutation) analysis to tentatively assessing viability furthermore proficiency of coverage criteria for testing.

Whalen et al. (2006) [49] characterized test coverage measurements on high level state formal programming prerequisites to help structural or white box testing. Gupta et al. (2000) [50] displayed another system execution based way to produce input data information for branch coverage.

V. THREATS TO AUTHENTICITY

Initially, keyword looking and reference analysis were utilized to get the obliged research papers. The irrelevant papers were then barred in the wake of reading the title, dynamic, abstract and conclusions. On the other hand, there is a probability that a few papers may be missed because of the above seeking and avoidance technique methods. Second, judgmental errors may bother in characterizing the papers into every classification.

VI. CONCLUSION

This survey paper has portrayed the research methods techniques, examined the results of the study and danger threats to the validity of the study. The orderly process was depicted as far as the research question characterized, seeking keywords utilized, and the avoidance and inclusion criteria. The consequences of the study was arranged into a few classifications and examined. The paper has demonstrated the area of analysis inside Test Coverage that has been carried out by noting the questions that were characterized at first. The vast majority of the research papers are from gathering and paper transactions, which demonstrate that more work needs to be carried out keeping in mind the end goal to enhance the momentum condition of research in test coverage estimation, analysis and measurement.

REFERENCES

- [1] ISTQB, "International Software Testing Qualification Board" version 2.0, 2007 www.istqb.org
- [2] Grinwald, R., Harel, E., Orgad, M., Ur S, Ziv, A., "User Defined Coverage – A tool Supported Methodology for Design Verification", IBM Research Lab, Haifa Dac, San Francisco, CA USA, pp. 158-163, 1998
- [3] Barbara Kitchenham, Stuart Charters, Guidelines for performing Systematic Literature Reviews in Software Engineering, EBSE Technical report 2007, Keel University, UK.
- [4] Williams, B. S. a. L. (2008). "A Survey on Code Coverage as a Stopping Criterion for Unit Testing.", Technical report (North Carolina State University. Dept. of Computer Science), TR-2008-22.
- [5] Qian Yang, J. Jenny Li, David M. Weiss, "A Survey of Coverage-Based Testing Tools", Published in The Computer Journal (2009) , volume 52 (5): pp. 589-597.
- [6] M.Prasanna, S.N. Sivanandam, R.Venkatesan, R.Sundarrajan, "A SURVEY ON AUTOMATIC TEST CASE GENERATION", published in Academic Open Internet Journal (AOIJ), Volume 15, 2005
- [7] Audris Mockus, N. N. (2009). "Test Coverage and Post-Verification Defects: A Multiple Case Study", In the 3rd International Symposium on Empirical Software Engineering and Measurement, 2009. ESEM 2009.
- [8] Silva, L. and S. Soares (2009). "Analyzing structure-based techniques for test coverage on a J2ME software product line", In 10th Latin American Test Workshop, 2009, LATW '09. Pp.1-6
- [9] Y. Wei (2008), "Is Coverage a Good Measure of Testing Effectiveness?", Chair of software engineering ETH Zurich, CH-8092 Zurich, Switzerland.
- [10] Derezińska, A. (2008). "Experiences from an Empirical Study of Programs Code Coverage.", Book section, Advances in Computer and Information Sciences and Engineering 2008, 57-62, DOI: 10.1007/978-1-4020-8741-7_11
- [11] Jun-Ru, C. and H. Chin-Yu (2007). "A Study of Enhanced MC/DC Coverage Criterion for Software Testing", In 31st Annual International Computer Software and Applications Conference, 2007 COMPSAC 2007

- [12] Berner, S., R. Weber, et al. (2007). "Enhancing Software Testing by Judicious Use of Code Coverage Information", In 29th International Conference on Software Engineering, 2007. ICSE 2007..
- [13] Lloyd, E. L. (2005). "A Study of Test Coverage Adequacy in the Presence of Stubs.", In the Journal of Object Technology, 2005, volume 4, pp.117--137
- [14] Joseph Lawrance, S. C., Margaret Burnett, Gregg Rothermel (2005). "How Well Do Professional Developers Test with Code Coverage Visualizations? an Empirical study", In the IEEE Symposium on Visual Languages and Human-Centric Computing, 2005 pp.53-60
- [15] Kim, Y. W. (2003). "Efficient Use of Code Coverage in Large-Scale Software Development.", In the Proceedings of the 2003 conference of the Centre for Advanced Studies on Collaborative research CASCON '03
- [16] Sakamoto, K., H. Washizaki, et al. (2010). "Open Code Coverage Framework: A Consistent and Flexible Framework for Measuring Test Coverage Supporting Multiple Programming Languages", In the 10th International Conference on Quality Software, QSIC, 2010, pp. 262-269
- [17] Matteo Bordin, C. C., Tristan Gingold (2009). "Couverture: an Innovative Open Framework for code coverage analysis of safety critical applications.", Couverture Open R/repository at Open-DO.org, <http://forge.opendo.org/projects/couverture>
- [18] Misurda, J., J. A. Clause, et al. (2005). "Demand-driven structural testing with dynamic instrumentation", In Proceedings of 27th International Conference on Software Engineering, 2005 ICSE 2005, pp. 165-165.
- [19] Koochakzadeh, V. G. a. N., "An Empirical Evaluation to Study Benefits of Visual versus Textual Test Coverage Information.", in the fifth conference on The Testing: Academic and Industrial Conference, Practice and Research Techniques (TAIC PART), 2010
- [20] Faizah Omar, S. I. "A Software Traceability Approach to Support Test Coverage Analysis" In the proceedings of Third International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services CENTRIC 2010.
- [21] Angeletti, D., E. Giunchiglia, et al. (2009). "Automatic Test Generation for Coverage Analysis of ERTMS Software", In the International Conference on Software Testing Verification and Validation, 2009. ICST '09.
- [22] Kapfhammer, G.M., Soffa, m.L.(2008) "Database-Aware Test Coverage Monitoring", Proceedings of the 1st India Software engineering conference ISEC'08, Hyderabad, India, ACM.77-86.
- [23] R. Lingampally, A. Gupta, P. Jalote. "A Multipurpose Code Coverage Tool for Java," In Proceedings of the 40th Annual Hawaii International Conference on System Sciences, IEEE Computer Society, 261b, 2007.
- [24] M. Kessiss, Y. Ledru, G. Vandome. "Experiences in Coverage Testing of a Java Middleware", in Proceedings SEM 2005, Lisbon, Portugal. ACM, pp. 3945, 2005.
- [25] Lormans M, D. (2005). "Reconstructing Requirements Coverage Views from Design and Test using Traceability Recovery via LSI", TEFSE, Long Beach, California, USA, ACM 2005
- [26] Eugenia Díaz, J. T., Raquel Blanco (2004). "A Modular Tool for Automated Coverage in Software Testing", In the Eleventh Annual International Workshop on Software Technology and Engineering Practice, 2003.
- [27] R. Krishnamoorthi *, S. A. S. A. M. (2009). "Factor oriented requirement coverage based system test case prioritization of new and regression test cases." ,Published in journal of Information and Software Technology, Volume 51, Issue 4, 2009
- [28] Ciyong, Chen, Xiaofeng, Xu, Yan, Chen, "A new method of test data generation for branch coverage in software testing based on EPDG and Genetic Algorithm". In the 3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication, 2009. ASID 2009
- [29] Tsai, W. T., Z. Xinyu, et al. (2007). "A Coverage Relationship Model for Test Case Selection and Ranking for Multi-version Software", In 10th IEEE High Assurance Systems Engineering Symposium, 2007. HASE '07.
- [30] Li, J. J. (2005). "Prioritize code for testing to improve code coverage of complex software", In the 16th IEEE International Symposium on Software Reliability Engineering, 2005. ISSRE 2005, 10-pp.
- [31] Gao, J., R. Espinoza, et al. (2005). "Testing coverage analysis for software component validation", In the 29th Annual International Computer Software and Applications Conference, 2005. COMPSAC 2005.
- [32] Asaf, S., E. Marcus, et al. (2004). "Defining coverage views to improve functional coverage analysis", In the Proceedings of 41st Design Automation Conference, 2004.
- [33] Jones, J. A. and M. J. Harrold (2003). "Test-suite reduction and prioritization for modified condition/decision coverage", In the journal of IEEE Transactions on Software Engineering, Volume 29 Issue 3, March 2003.
- [34] Tikir, M. M. and J. K. Hollingsworth (2002). "Efficient instrumentation for code coverage testing", In Proceedings of the ACM SIGSOFT 2002 International Symposium on Software Testing and Analysis, pp.86-96.
- [35] Atif Memon, "Coverage Criteria for GUI Testing", In ACM SIGSOFT Software Engineering Notes. Vol. 26, no.5, pp. 256-267. Sept. 2001
- [36] Jinxia, J. Z. (2010). "Software Reliability Modeling with Integrated Test Coverage.", In the Fourth International Conference on Secure Software Integration and Reliability Improvement (SSIRI), 2010 pp. 106
- [37] Smidts, C. (2009). "A Test Coverage-Based Model for Predicting Software Fault Content and Location during Multi-phase Functional Testing", In International Conference on Advanced Software Engineering & Its Applications (ASEA 2009), Korea
- [38] Haifeng, L. Qiuying, et al. (2008). "Software Reliability Modeling with Logistic Test Coverage Function", In 19th International Symposium on Software Reliability Engineering, 2008. ISSRE 2008.
- [39] Xia, C. and M. R. Lyu (2007). Software Reliability Modeling with Test Coverage: Experimentation and Measurement with A Fault-Tolerant Software Project", In the 18th IEEE International Symposium on Software Reliability, 2007. ISSRE '07.
- [40] CAI Xia, (2006), "Coverage Based Testing Strategies and Reliability Modeling for Fault-Tolerant Software Systems", in CSA, Dissertation Abstracts International. Vol. 68, no. 3. 2006
- [41] J. -Y. PARK, (2006), "COVERAGE GROWTH FUNCTIONS FOR SOFTWARE RELIABILITY MODELING", Book article, Advanced Reliability Modeling, II, Reliability Testing and Improvement, (pp. 435-442).
- [42] H Pham, (2003), "NHPP software reliability and cost models with Test coverage", In the European Journal of Operational Research Volume 145, Issue 2, 1 March 2003, Pages 443-454.
- [43] Malaiya, Y. K., M. N. Li, et al. (2002). "Software reliability growth with test coverage", In Journal IEEE Transactions on Reliability. Vol. 51, no. 4, pp. 420-426. Dec. 2002.
- [44] Mei-Hwa Chen, Michael R., (2001), "Effect of Code Coverage on Software Reliability Measurement", in journal IEEE Transactions on Reliability, vol.50, no.2, 2001, pp.165-170.

- [45] Ricardo D. F. Ferreira, J. P. F., Ana C. R. Paiva (2010). "Test Coverage Analysis of UML State Machines", In the International Conference on Software Testing, Verification and Validation Workshops, 2010. ICSTW '10. , pp. 284-289.
- [46] Rauf, A., S. Anwar, (2010). "Automated GUI Test Coverage Analysis Using GA", Proceedings of the 2010 Seventh International Conference on Information Technology: New Generations, pp. 1057-1062.
- [47] Takahashi, J., H. Kojima, et al. (2008). "Coverage Based Testing for Concurrent Software", In 28th International Conference on Distributed Computing Systems Workshops, 2008. ICDCS '08.
- [48] Jalote, A. Gupta. P. (2008). "An approach for experimentally evaluating effectiveness and efficiency of coverage criteria for software testing.", In International Journal on Software Tools for Technology Transfer (STTT), Volume 10, Number 2, 145-160, DOI: 10.1007/s10009-007-0059-5
- [49] Whalen, M. W., M. P. E. Heimdahl, et al. (2006). "Coverage metrics for requirements-based testing." 2006, In Proceedings of the 2006 International Symposium on Software Testing and Analysis, ISSTA 2006, pp. 25-35.
- [50] Sidirolou[‡], P. Akritidis[?], K. Xinidis[?], E. Markatos[?], A.D. Keromytis[‡], "Detecting Targeted Attacks Using Shadow Honeypots", [‡]CIS Department, Univ. of Pennsylvania, [‡]Department of Computer Science, Columbia University, [?]Institute of Computer Science – FORTH.

