Functional Software Testing: A Systematic Mapping Study

Gilmar Ferreira Arantes
Instituto de Informática
Universidade Federal de Goiás, UFG
Goiânia-GO, Brazil
e-mail: gilmar@inf.ufg.br

Plínio de Sá Leitão-Júnior
Instituto de Informática
Universidade Federal de Goiás, UFG
Goiânia-GO, Brazil
e-mail: plinio@inf.ufg.br

Auri Marcelo Rizzo Vincenzi
Instituto de Informática
Universidade Federal de Goiás, UFG
Goiânia-GO, Brazil
e-mail: auri@inf.ufg.br

Fábio Nogueira de Lucena
Instituto de Informática
Universidade Federal de Goiás, UFG
Goiânia-GO, Brazil
e-mail: fabio@inf.ufg.br

***Abstract*—Software testing is part of a set of activities that ensure high quality software. It primarily aims at revealing defects that have been inserted into a software at various stages of its development. In functional testing, test requirements are derived from software specifications. This paper proposes a systematic map (SM). Its planning and execution were based on questions formulated to investigate functional criteria/techniques related to: i) assessment methods, which have an effect on cost and efficacy; and ii) application scenarios, which define the type of software in which they are used. Furthermore, we assess the strength of evidence and threats to SM validity.**

Keywords-software testing; testing techniques and criteria; functional testing; systematic mapping.

I. Introduction

Software testing is a knowledge area within the field of software engineering, which strives for quality and continually contributes to process and product improvement. The test's main objective is to reveal defects in the software so these may be solved prior to any damage. Ideally, the testing activity must be systematic, and the techniques used must balance cost reduction and increase the levels of defect detection, should any exist. Each technique has a set of test criteria, which may be used during the conception, selection, and evaluation of a test set.

Among the different types of testing techniques, functional testing has an important role for software quality improvement as it complements other methods. Thus, it is relevant to: (i) know how functional testing criteria are employed; (ii) identify weak and strong points; and (iii) describe scenarios in which they are used.

This paper's contributions are obtained through a systematic mapping study. According to Wohlin et al. [1], it follows the same processes and principles used in systematic literature reviews, although it has different criteria for quality assessment and inclusion/exclusion of studies. Due to its wider and more varied range, both the collected data and the literature review are mainly qualitative. The research questions avoid any tendencies; instead, they are more specific and often relate to empirical studies.

The systematic map aims at answering the following questions pertaining to functional software testing:

- **Primary research question**: Which comparisons have been made between test criteria?
- **Secondary research question**: What is the application scenario for each functional testing criterion?

The purpose of the primary research question is to find weak and strong points of functional testing criteria through comparisons made between them. Many aspects are observed, i.e., application costs and ability to detect defects. This question is considered primary because it: (i) provides information on the type of application and limitations; (ii) determines factors influencing efficiency and efficacy; and (ii) contributes to the proposal of other approaches to functional testing.

The secondary research question aims to identify the type of software in which functional criteria are used. It establishes criteria range and determines its application and restricted use in some areas.

The rest of our paper is thus organized: Section II presents the systematic mapping protocol and how it was conducted. Section III shows the results as they relate to our research questions. Section IV discusses the strength of evidence and threats to validity of the primary studies selected. Finally, Section V is made up of final considerations and research implications.

 ISBN: 978-1-61208-304-9

II. Mapping Planning

The systematic mapping protocol was planned according to the model presented by Biolchini et al. [2]. This section explores the main points of the elaborated plan.

A. Scope of studies

The protocol identified the scope of the studies by considering:

1) **Population** – Scientific publications on software testing;
2) **Intervention**: Functional testing criteria.
3) **Results**:
    a) Properties, characteristics and comparisons between functional testing criteria;
    b) Application context of each functional testing criterion.
4) **Application** – association among functional testing criteria to help detect defects; support for an effective use of each criterion, in isolation or as a set; assistance for the proposal of new functional criteria.

B. Search strategy for selecting primary studies

The strategy for searching and selecting primary studies was defined according to the research sources, keywords, language, and types of primary studies selected for mapping:

1) **Criteria for source selection** – Electronic indexing databases and internet search engines.
2) **Search methods** – Manually and web search engine.
3) **Source listing** – Conferences, journals and technical reports indexed by IEEExplore, ACM Digital Library and Google Scholar.
4) **Language of primary studies** – English, due to its widespread use in scientific writing.

C. Pilot search execution

A search string was defined for each indexed database considering the research questions, their respective quality and amplitude traits, as well as the search strategy for selecting primary studies.

D. Criteria and procedure for selecting studies

1) *Inclusion criteria*:
    a) $\mathbf{IC}_1$ – Papers mentioning any features of a functional testing criterion;
    b) $\mathbf{IC}_2$ – Papers comparing functional properties;
    c) $\mathbf{IC}_3$ – Papers comparing properties of functional and structural testing criteria, as well as those of the random testing technique.
2) *Exclusion criteria*:
    a) $\mathbf{EC}_1$ – Papers in which software testing is only mentioned and is not the main topic;
    b) $\mathbf{EC}_2$ – Papers discussing software testing, but whose focus is not on functional or random testing techniques;
    c) $\mathbf{EC}_3$ – Papers discussing functional testing criteria, which are not in any of the criteria groups previously defined for analysis;
    d) $\mathbf{EC}_4$ – Papers discussing functional testing criteria, although its focus is not mentioned in any of the categories previously defined for analysis;
    e) $\mathbf{EC}_5$ – Papers describing systematic procedures for test criteria assessment, frameworks, benchmarks for the comparison of testing methods, but which do not actually make any comparisons;
    f) $\mathbf{EC}_6$ – Papers comparing test methods, which do not include functional testing;
    g) $\mathbf{EC}_7$ – Papers discussing functional testing related to formal specifications;
    h) $\mathbf{EC}_8$ – Papers focusing on theoretical analysis with no practical examples of the approach.

E. Selection process of primary studies

1) *Preliminary selection process* – Retrieved papers were analysed by reviewers, who were responsible for reading titles and abstracts. Once a paper was considered relevant by the reviewers, it would be fully read.
2) *Final selection process* – All papers selected were fully read by at least one reviewer, who then elaborated a document including abstracts, methodologies and testing methods mentioned in each paper, as well as other related concepts.
3) *Quality assessment of primary studies* – Researchers assessed the selected papers according to the quality criteria defined by Ali et al. [3].

F. Final selection

The final selection was carried out through four phases. Phase 1 refers to the primary studies retrieved from the electronic databases after the application of search strings. Phase 2 corresponds to the studies resulting from the preliminary selection process. Some studies were excluded because their titles and abstracts did not pertain to our research questions. Phase 3 refers to the studies obtained from the final selection process. Some studies were also excluded once they were fully read for the same reason stated above. In Phase 4, some studies were excluded for their low quality according to the quality criteria defined during the planning stage of the systematic map. In summary, a total of 27 primary studies were selected, of which:

 ISBN: 978-1-61208-304-9

- 14 are from the IEEE database;
- 7 are from the ACM database;
- 4 are from Google Scholar;
- and 2 are directly from Universidade Federal de Goiás (UFG).

Figure 1 shows a distribution of studies spanning from 1978 to 2011. This time span corresponds to the publishing year of the oldest study retrieved from the search string and the year the mapping ended, respectively. The graph shows that the highest number of publications on this subject occurred in 2006 (a total of 6). Furthermore, between 2008 and 2011 there were fewer studies, but a continued interest for research in this area.

G. Digraph of internal citations

To illustrate primary studies that refer to one or more studies from the selected set, we constructed a directed graph (digraph) to identify entry and exit points. Figure 1 shows a representation of the digraph.

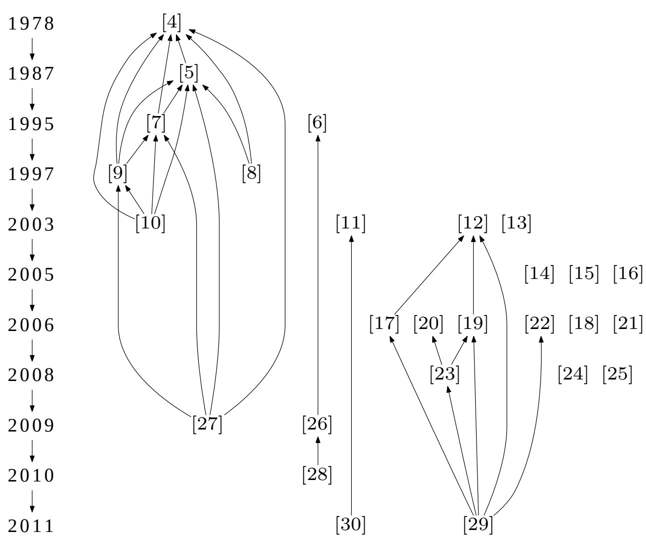

Figure 1. Citations among studies classified by year

.

Figure 1 reveals some areas of concentrated citations among primary studies. For instance, we identified an area of citations in which study [4] has the highest number of entries. This is due to the fact that it was one of the first published studies that approached the comparison of testing techniques. Another identified region includes study [29] with the highest number of exits. It is a survey, therefore it refers to many other primary studies. Finally, another region contains studies [6], [26] and [28], all of which use the same criteria for functional testing: Decision Table and Cause and Effect Graph.

III. Results

Table I presents testing criteria and techniques that were identified in the primary studies. The inspection approach is also used in these studies. The first column lists the criteria/techniques; in some cases, test approaches are not necessarily identified as a criterion, as stated in the literature. The second column shows the number of primary studies that use such criterion/technique. The third column lists the references used in the primary studies, and the last column indicates whether the criterion/technique is relevant to mapping. Thus, Table I shows that: (i) studies in general use more than one test criterion/technique; (ii) in many cases, functional, structural and other testing or code inspection techniques are compared in the same study; (iii) the following criteria are most used: Boundary Value Analysis, Equivalence Class Partitioning, and Decision Table.

Table I. Test criteria, techniques and approaches discussed in the studies analysed

Test Criteria/Techniques and Approaches	# Refs	References
Boundary Value Analysis	12	[5], [7], [8], [9], [10], [11], [13], [16], [18], [25], [27], [28]
Path Coverage	1	[27]
Statement Coverage	1	[5]
Condition Coverage	4	[7], [9], [10], [27]
Inspection/Code Review	6	[4], [5], [7], [9], [10], [27]
Cause-Effect Graph	3	[6], [26], [28]
Random Partitioning	1	[14]
Dynamic Partitioning	1	[14]
Equivalence Class Partitioning	11	[5], [7], [8], [9], [10], [11], [15], [16], [18], [27], [28]
Decision Table	6	[6], [15], [24], [26], [27], [28]
Test using Collaboration Diagram	1	[21]
Test using Object-Z	1	[21]
Test using OCL	1	[21]
Random Testing	2	[8], [14]
Use Case Test	6	[12], [19], [20], [21], [22], [23]
Extended Use Case Test	1	[21]
Structural Testing (without a specific criterion)	1	[4]
Functional Testing (without a specific criterion)	3	[4], [17], [29]
Systematic Functional Testing	2	[11], [30]
Extended Systematic Functional Testing	1	[30]

A. Results of the primary question: Which comparisons have been made between test criteria?

This question aimed at identifying primary studies that carried out comparisons between functional test criteria from any perspective. Results revealed few studies with such an objective. Among the studies analysed, only [21] and [27] make comparisons. The former compares criteria applied to object-oriented systems, whereas the latter uses both Boundary Value Analysis and Equivalence Class Partitioning (also known as Equivalence Partitioning) and compares them to other test criteria, i.e., Decision Table.

In our third inclusion criterion, which includes studies comparing structural and random testing techniques, nine studies were added to the previous two. Therefore, a total of 11 studies were selected for the primary question. Among the criteria considered of interest to our systematic mapping, Vallespir and Herbert [27] concluded that Equivalence Partitioning obtained better results than Decision Table regarding three comparative features: (i) number of defects, (ii) detection time and (iii) efficiency (quantity/time). Seo and Choi [21] concluded that Extended Use Case Test and Test Derived from Formal OCL Specifications are the most effective and suggested the combined use of them.

All studies presented in [4], [5], [7], [9], and [10] stated that, in general, Boundary Value Analysis and Equivalence Class Partitioning showed the best results regarding the number of defects detected in a short period of time. However, almost all of them agree that results depended on program type, tester experience and type of defect detected.

Similarly to studies [5] and [7], study [9] noted that up until 1997: (i) there was no consistent evidence to support that one technique for defect detection was better than another; on the contrary, current evidence suggests that every technique has its own merits; (ii) current evidence shows that functional, structural and code review testing techniques complement one another, and should be used in combination.

In summary, comparative features relevant to the research question were applied to the selected studies. However, the results obtained from the application of these features are not definitive for two main reasons: (a) tested programs are very small and simple, and (b) defects are inserted by the tester. We consider our results as contributions to knowledge pertaining to test criteria/techniques. Thus, results may be analysed as tendencies and not as conclusions, because they cannot be generalized.

B. Results of the secondary question: What is the application scenario for each functional testing criterion?

Table II shows the studies selected to answer this research question. They were classified according to study type (experiment, theoretical analysis, simulation, case study, survey) and scope. Such perspective is relevant to assess the strength of evidence, which will be discussed in Subsection IV-A.

Table III presents application scenarios for each test criterion. It lists criteria according to the number of scenarios in which they are applied. Results revealed recurring scenarios in various criteria, which shows multiplicity of scenarios and criteria (n:n – "many for many"). In other words, the studies do not identify exclusiveness between Scenario A and Criterion B. This may be regarded as positive because criteria application scope is non-restricted within the scenarios identified.

Table II. Identified test scenarios in primary studies selected

Reference	Study Type	Scope of Study
[19]	Case study	Industry
[24]	Simulation	Industry
[25]	Simulation	Laboratory
[4]	Experiment	Academy
[6]	Theoretical analysis	Laboratory
[7]	Experiment	Academy
[8]	Experiment	Industry
[9]	Experiment	Academy
[10]	Experiment	Academy
[11]	Case study	Laboratory
[12]	Case study	Laboratory
[13]	Theoretical analysis	Industry
[14]	Experiment	Industry
[15]	Simulation	Laboratory
[17]	Survey	Laboratory
[18]	Theoretical analysis	Laboratory
[20]	Case study	Industry
[21]	Experiment	Laboratory
[22]	Simulation	Industry
[23]	Case study	Industry
[26]	Theoretical analysis	Laboratory
[27]	Experiment	Academy
[28]	Simulation	Laboratory
[30]	Case study	Academy

Table III. Test criteria/technique and scenarios

Test Criterion/Technique	Test Scenario
Boundary Value Analysis	Academic/didactic system, Non safety-critical commercial information system, Aircraft operational system, Operating system utility and Embedded commercial systems
Equivalence Class Partitioning	Academic/didactic system, Non safety-critical commercial information system, Aircraft operational system and Operating system utility
Decision Table	Academic/didactic system, Non safety-critical commercial information systems and web service
Use Case Test	Video conference, Safety-critical embedded aviation system, Safety-critical commercial information system, Safety-critical financial system, Safety-critical web system and Academic/didactic system
Cause and Effect Graph	Academic/didactic system
Extended systematic functional testing	Strategic management system and Critical commercial information system
Dynamic Partitioning	Air traffic control
Extended Use Case Test	Critical financial system
Systematic Functional Testing	Operating system utility

Results regarding scenarios showed that systems were mainly tested in academic/didactic environments, to which a total of six test criteria were applied. Next, four test criteria were used in non safety-critical commercial information systems. This is due to the fact that most studies analysed (70.38%) were developed in academic environments or laboratories. However, criteria were also

 ISBN: 978-1-61208-304-9

applied to real life settings, i.e., safety-critical scenarios, response time, robustness, as shown in studies [19], [20], [24]. Such scenarios involve embedded systems for military aircrafts, web service testing, ticket management systems (for integrated transport systems in large metropolitan areas) and electronic component testing (mobile devices, cell phones, remote controls, television).

Among test criteria, Use Case Test was most frequent in scenarios involving critical systems (five out of three scenarios). In Extended Systematic Information Systems and Random Testing, scenarios were only applied to strategic or critical systems. Cause and Effect Graph was used only in academic/didactic scenarios. The remaining criteria were mainly applied in academic/didactic scenarios or in ones involving non safety-critical systems.

Furthermore, the first five lines in Table III show that the criteria most used in the studies were applied in a variety of scenarios.

IV. Discussion

A. Strength of evidence

Assessment of the strength of evidence is a key factor for assessing the reliability of conclusions and consequent recommendations [3], [7].

There are many systems for assessing strength of evidence. For our research, we used the GRADE system (Grading of Recommendations Assessment, Development and Evaluation) for two reasons: (i) its definitions involve the main weak points of systems that classify evidence based on hierarchy, and (ii) it may be used by other software engineering researchers [3].

The GRADE system identifies four levels of strength of evidence: high, moderate, low and very low. It is determined by a combination of four elements: study characteristics, quality, consistency and directness.

In terms of study characteristics, two thirds of the studies are observational, and one third of them are experimental. Thus, the strength of evidence of the systematic mapping is low according to GRADE definitions [3].

On the topic of study quality, data analysis approaches were moderately explained in terms of study implications, credibility and limitations. In only six out of 27 studies researchers made critical analyses of their role during research. Result credibility was discussed in 85.19% of studies. A total of 88.89% of studies pondered over their limitations. Based on these results, we may conclude that studies showed moderate evidence regarding quality.

The consistency criterion was similar across studies, given that all of them applied functional testing by use of one criterion or more, individually or in a set, in a certain scenario or in comparative experiments using criteria from other testing techniques. Therefore, the strength of evidence related to consistency was high.

Next, the aim was to test objectiveness (directness). Most studies (70.38%) were carried out in academic/laboratory contexts. Regarding intervention, most studies investigated functional testing criteria and techniques, as defined during planning. Results also showed that most studies requires empirical validation through real applications. Thus, the strength of evidence ranges between moderate and low in relation to directness.

The strength of evidence of our proposed systematic map reaches a moderate level when all four aspects are combined. Therefore, future research may alter its reliability estimate.

B. Threats to Validity

According to [31], our proposed systematic map may face two threats to its validity: (i) limitations of research sources; (ii) elaboration of research questions in accordance with works in the scientific community on the same knowledge area under investigation.

Associated with the first threat is the fact that IEEExplore and ACM Digital Library indexed databases were highly used, which may have prevented the identification of relevant primary studies that were not published in any of the two sources. Related to the second threat is the fact that the scope of the primary question includes comparisons among functional criteria as well as comparisons with criteria used in non-functional techniques.

A third threat was identified: there was no evidence of objective comparisons between test criteria. Despite this, criteria were compared in relation to efficacy, cost and efficiency. However, we noted that these factors are dependent on other ones, i.e., tester experience, the type and size of the program being tested, etc.

V. Final Considerations

The present work focused on software functional testing to contribute with its assessment and evolution. A detailed study of various functional criteria was carried out through a systematic map.

The systematic map was planned based on the model elaborated by Biolchini et al. [2] and was carried out following these research questions:

- Primary research question: Which comparisons have been made between test criteria?
- Secondary research question: What is the application scenario for each functional testing criterion?

A set of 27 primary studies were investigated. Each of them provided relevant information to support conclusions which were the basis for answering our research questions.

 ISBN: 978-1-61208-304-9

Regarding the primary question, only two studies compared functional testing among them, which little contributed to consolidate functional criteria knowledge and practice. A total of nine studies made comparisons between functional criteria and criteria applied to other testing techniques, i.e., Structural Testing and Random Testing. These studies showed that a certain criterion is more effective in given contexts and scenarios. We may thus conclude that testing techniques and criteria complement each other and should be applied as a set to obtain more effective results during the test process. The results of such comparisons were influenced by factors such as tester experience, type and size of the program under testing and defect types in the program.

Regarding the secondary question, as a contribution to industry and practitioners in the application of testing techniques, Boundary Value Analysis was the most used test criterion because it was analysed in a larger number of scenarios. Many application scenarios of functional test criteria were identified. The academic/learning scenario was present in most of the studies analysed. The Use Case Test was the most used in safety-critical scenarios. No scenario was exclusive to any test criterion. Tester experience and creativity were essential for criteria application, even when they were not recommended in a certain scenario.

After considerations related to the research questions had been made, the primary studies were assessed according to the quality criteria defined by Ali et al. [3] to verify strength of evidence and establish the reliability level of results. We concluded that the strength of evidence of our systematic map was moderate.

Threats to validity were also identified and assessed to verify what effects they would have in our research. Furthermore, we found that there are no similar systematic reviews. However, we identified some reviews with a specific focus, i.e., Model-based testing and concurrent software testing. This study seeks to encourage further research on systematic mapping, which is able to provide more answers to our research questions and help develop their strength of evidence.

As a future work, we intend to perform a deeper analysis of data related to the second research question, trying to provide more evidences to industry and practitioners.

Acknowledgment

The authors would like to thank the Instituto de Informática (INF/UFG), and the Brazilian Funding Agencies – FAPEG, and CAPES – which support this work.

References

[1] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*. New York, NY, USA: Springer Heidelberg, 2012.

[2] J. C. de Almeida Biolchini, P. G. Mian, A. C. C. Natali, T. U. Conte, and G. H. Travassos, "Scientific research ontology to support systematic review in software engineering," *Adv. Eng. Inform.*, vol. 21, no. 2, pp. 133–151, Apr. 2007, [retrieved: Jan., 2012]. [Online]. Available: http://goo.gl/sWxntK

[3] M. S. Ali, M. Ali Babar, L. Chen, and K.-J. Stol, "A systematic review of comparative evidence of aspect-oriented programming," *Inf. Softw. Technol.*, vol. 52, no. 9, pp. 871–887, Sep. 2010, [retrieved: Jan., 2012]. [Online]. Available: http://goo.gl/BWkt4O

[4] G. J. Myers, "A controlled experiment in program testing and code walkthroughs/inspections," *Commun. ACM*, vol. 21, no. 9, pp. 760–768, Sep. 1978, [retrieved: Jan., 2012]. [Online]. Available: http://goo.gl/xuMFHS

[5] V. Basili and R. Selby, "Comparing the effectiveness of software testing strategies," *Software Engineering, IEEE Transactions on*, vol. SE-13, no. 12, pp. 1278–1296, 1987.

[6] K. Nursimulu and R. L. Probert, "Cause-effect graphing analysis and validation of requirements," in *Proceedings of the 1995 conference of the Centre for Advanced Studies on Collaborative research*, ser. CASCON '95. IBM Press, 1995, pp. 46–46, [retrieved: Jan., 2012]. [Online]. Available: http://goo.gl/OqMw8U

[7] E. Kamsties and C. M. Lott, "An empirical evaluation of three defect-detection techniques," in *Proceedings of the 5th European Software Engineering Conference.* London, UK, UK: Springer-Verlag, 1995, pp. 362–383, [retrieved: Jan., 2012]. [Online]. Available: http://goo.gl/VaraHr

[8] S. C. Reid, "An empirical analysis of equivalence partitioning, boundary value analysis and random testing," in *Proceedings of the 4th International Symposium on Software Metrics*, ser. METRICS '97. Washington, DC, USA: IEEE Computer Society, 1997, pp. 64–73, [retrieved: Jan., 2012]. [Online]. Available: http://goo.gl/DhMfls

[9] M. Wood, M. Roper, A. Brooks, and J. Miller, "Comparing and combining software defect detection techniques: a replicated empirical study," in *Proceedings of the 6th European SOFTWARE ENGINEERING conference held jointly with the 5th ACM SIGSOFT international symposium on Foundations of software engineering*, ser. ESEC '97/FSE-5. New York, NY, USA: Springer-Verlag New York, Inc., 1997, pp. 262–277, [retrieved: Jan., 2012]. [Online]. Available: http://goo.gl/rGW3aU

[10] N. Juristo and S. Vegas, "Functional testing, structural testing and code reading: What fault type do they each detect?" in *Empirical Methods and Studies in Software Engineering*, ser. Lecture Notes in Computer Science, R. Conradi and A. Wang, Eds. Springer Berlin / Heidelberg, 2003, vol. 2765, pp. 208–232.

[11] S. Linkman, A. M. R. Vincenzi, and J. C. Maldonado, "An evaluation of systematic functional testing using

mutation testing," *7th International Conference on Empirical Assessment in Software Engineering [EASE. [S.l.: s.n.]]*, 2003.

[12] C. Nebut, F. Fleurey, Y. L. Traon, and J.-M. Jézéquel, "Requirements by contracts allow automated system testing," in *Proceedings of the 14th International Symposium on Software Reliability Engineering*, ser. ISSRE '03. Washington, DC, USA: IEEE Computer Society, 2003, pp. 85–96, [retrieved: Jan., 2012]. [Online]. Available: http://goo.gl/r2D0BW

[13] M. Ramachandran, "Testing software components using boundary value analysis," in *Proceedings of the 29th Conference on EUROMICRO*, ser. EUROMICRO '03. Washington, DC, USA: IEEE Computer Society, 2003, pp. 94–98, [retrieved: Jan., 2012]. [Online]. Available: http://goo.gl/2gi7sT

[14] K.-Y. Cai, T. Jing, and C.-G. Bai, "Partition testing with dynamic partitioning," in *Proceedings of the 29th annual international conference on Computer software and applications conference*, ser. COMPSAC-W'05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 113–116, [retrieved: Jan., 2012]. [Online]. Available: http://goo.gl/z82pJ6

[15] E. L. Jones, "Automated support for test-driven specification," Phoenix, Arizona, pp. 218–223, nov. 2005.

[16] T. Murnane, R. Hall, and K. Reed, "Towards describing black-box testing methods as atomic rules," in *Proceedings of the 29th Annual International Computer Software and Applications Conference - Volume 01*, ser. COMPSAC '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 437–442, [retrieved: Jan., 2012]. [Online]. Available: http://goo.gl/qhltH0

[17] J. J. Gutierrez, M. J. Escalona, M. Mejías, and J. Torres, "Generation of test cases from functional requirements. a survey," in *4th Workshop on System Testing and Validation*, Potsdam, Germany, 2006, [retrieved: Jan., 2012]. [Online]. Available: http://goo.gl/Cqn1B0

[18] R. M. Hierons, "Avoiding coincidental correctness in boundary value analysis," *ACM Trans. Softw. Eng. Methodol.*, vol. 15, no. 3, pp. 227–241, Jul. 2006, [retrieved: Jan., 2012]. [Online]. Available: http://goo.gl/dl0JxS

[19] C. Nebut, F. Fleurey, Y. Le-Traon, and J.-M. Jezequel, "Automatic test generation: a use case driven approach," *Software Engineering, IEEE Transactions on*, vol. 32, no. 3, pp. 140–155, 2006.

[20] S. Roubtsov and P. Heck, "Use case-based acceptance testing of a large industrial system: Approach and experience report," in *Proceedings of the Testing: Academic & Industrial Conference on Practice And Research Techniques*, ser. TAIC-PART '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 211–220, [retrieved: Jan., 2012]. [Online]. Available: http://goo.gl/1M1F5F

[21] K. I. Seo and E. M. Choi, "Comparison of five black-box testing methods for object-oriented software," in *Proceedings of the Fourth International Conference on Software Engineering Research, Management and Applications*, ser. SERA '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 213–220, [retrieved: Jan., 2012]. [Online]. Available: http://goo.gl/1eju7r

[22] P. Zielczynski, "Traceability from use cases to test cases," On-line article, 2006, [retrieved: Jan., 2012]. [Online]. Available: http://goo.gl/RqoGJ3

[23] J. Gutierrez, M. Escalona, M. Mejias, J. Torres, and A. Centeno, "A case study for generating test cases from use cases," in *Research Challenges in Information Science, 2008. RCIS 2008. Second International Conference on*, 2008, pp. 209–214.

[24] S. Noikajana and T. Suwannasart, "Web service test case generation based on decision table (short paper)," in *Quality Software, 2008. QSIC '08. The Eighth International Conference on*, 2008, pp. 321–326.

[25] K. Vij and W. Feng, "Boundary value analysis using divide-and-rule approach," in *Information Technology: New Generations, 2008. ITNG 2008. Fifth International Conference on*, 2008, pp. 70–75.

[26] P. R. Srivastava, P. Patel, and S. Chatrola, "Cause effect graph to decision table generation," *SIGSOFT Softw. Eng. Notes*, vol. 34, no. 2, pp. 1–4, Feb. 2009, [retrieved: Jan., 2012]. [Online]. Available: http://goo.gl/qhYxB0

[27] D. Vallespir and J. Herbert, "Effectiveness and cost of verification techniques: Preliminary conclusions on five techniques," in *Computer Science (ENC), 2009 Mexican International Conference on*, 2009, pp. 264–271.

[28] M. Sharma and B. Chandra, "Automatic generation of test suites from decision table - theory and implementation," in *Software Engineering Advances (ICSEA), 2010 Fifth International Conference on*, 2010, pp. 459–464.

[29] M. J. Escalona, J. J. Gutierrez, M. Mejías, G. Aragón, I. Ramos, J. Torres, and F. J. Domínguez, "An overview on test generation from functional requirements," *J. Syst. Softw.*, vol. 84, no. 8, pp. 1379–1393, Aug. 2011, [retrieved: Jan., 2012]. [Online]. Available: http://goo.gl/Jq63fE

[30] A. R. Vidal, "Extended systematic funcional test: A contribution in the application of black-box testing criteria," Master's thesis, Universidade Federal de Goiás, Goiânia, 2011, (in Portuguese).

[31] D. Budgen, A. J. Burn, O. P. Brereton, B. A. Kitchenham, and R. Pretorius, "Empirical evidence about the uml: a systematic literature review," *Softw. Pract. Exper.*, vol. 41, no. 4, pp. 363–392, Apr. 2011, [retrieved: Jan., 2012]. [Online]. Available: http://goo.gl/qEc82C

 ISBN: 978-1-61208-304-9