Software Qual J (2013) 21:445–478
DOI 10.1007/s11219-012-9181-z

Test case prioritization: a systematic mapping study

Cagatay Catal · Deepti Mishra

Published online: 26 July 2012

Abstract Test case prioritization techniques, which are used to improve the cost-effectiveness of regression testing, order test cases in such a way that those cases that are expected to outperform others in detecting software faults are run earlier in the testing phase. The objective of this study is to examine what kind of techniques have been widely used in papers on this subject, determine which aspects of test case prioritization have been studied, provide a basis for the improvement of test case prioritization research, and evaluate the current trends of this research area. We searched for papers in the following five electronic databases: IEEE Explorer, ACM Digital Library, Science Direct, Springer, and Wiley. Initially, the search string retrieved 202 studies, but upon further examination of titles and abstracts, 120 papers were identified as related to test case prioritization. There exists a large variety of prioritization techniques in the literature, with coverage-based prioritization techniques (i.e., prioritization in terms of the number of statements, basic blocks, or methods test cases cover) dominating the field. The proportion of papers on model-based techniques is on the rise, yet the growth rate is still slow. The proportion of papers that use datasets from industrial projects is found to be 64 %, while those that utilize public datasets for validation are only 38 %. On the basis of this study, the following recommendations are provided for researchers: (1) Give preference to public datasets rather than proprietary datasets; (2) develop more model-based prioritization methods; (3) conduct more studies on the comparison of prioritization methods; (4) always evaluate the effectiveness of the proposed technique with well-known evaluation metrics and compare the performance with the existing methods; (5) publish surveys and systematic review papers on test case prioritization; and (6) use datasets from industrial projects that represent real industrial problems.

C. Catal (✉)
Department of Computer Engineering, Istanbul Kultur University, 34156 Istanbul, Turkey
e-mail: c.catal@iku.edu.tr

D. Mishra
Department of Computer Engineering, Atilim University, 06836 Ankara, Turkey
e-mail: deepti@atilim.edu.tr

Keywords Systematic literature review · Systematic mapping study · Test case prioritization · Regression testing

1 Introduction

Some of the new features coming in new versions of software may adversely affect the existing and unchanged components of the software. Therefore, it is crucial to revalidate the software by running existing test cases in addition to the newly developed test cases. This type of testing due to the software modifications is called regression testing, and it may be very expensive in some circumstances. An industrial collaborator of a researcher (Elbaum et al. 2003)[1] stated that a test suite of software having 20,000 lines of code requires 7 weeks to run. Another industrial collaborator who runs tests continuously in a test cycle requires over 30 days to complete the testing activity (Elbaum et al. 2003). Even if the testing duration is shorter, the cost of testing may still be expensive due to the required human effort to set up, execute, and monitor the tests. Hence, researchers have developed diverse techniques to improve the cost-effectiveness of regression testing. These techniques are discussed and formally described by Yoo and Harman (2012):

- *Test Suite Minimization* (*TSM*)/*Test Suite Reduction* (*TSR*): These techniques remove redundant test cases permanently to reduce the size of the test suite. However, the fault detection capability of the test suite may decrease due to the reduction in the number of test cases.
- *Test Case Selection* (*TCS*)/*Regression Test Selection* (*RTS*): These techniques select some of the test cases and focus on the ones that test the changed parts of the software. Contrary to the TSR techniques, RTS does not remove test cases but selects test cases that are related to the changed portion of the source code.
- *Test Case Prioritization* (*TCP*): This type of technique identifies the efficient ordering of test cases to maximize certain properties, such as the rate of fault detection or coverage rate.

While both the TSM and TCS techniques reduce testing time, they can omit some significant test cases that can detect certain types of faults and hence cause an increase in the software cost (Do et al. 2010). However, TCP techniques use the entire test suite and reduce testing cost by achieving parallelization of the debugging and testing activities (Do et al. 2010). As the main theme of this paper is TCP, we will not discuss the details of TSM and TCS approaches here. Readers may examine the work of Yoo and Harman (2012) in order to learn more about these approaches. In this section, we will provide introductory information on TCP approaches.

Test case prioritization techniques improve the cost-effectiveness of regression testing and provide several benefits, such as earlier detection of faults and earlier feedback to testers. Many papers provide clear evidence that these prioritization techniques can be beneficial for regression testing (Do et al. 2006, 2010; Elbaum et al. 2000; Korel and Koutsogiannakis 2009). Most of these techniques are code coverage-based, and there is a limited number of prioritization techniques based on design models. Test case prioritization techniques can be distinguished based on the following dimensions (Do et al. 2006):

[1] [[ref#]] refers to papers in Appendix A.

- *The Type of Code Coverage Information:* Prioritization can be performed in terms of the number of statements, basic blocks ("single-entry, single-exit sequences of statements" [[33]]), or methods test cases cover. For example, total method coverage prioritization sorts the test cases in order by the number of methods they cover (Do et al. 2006).
- *The Use of Feedback:* For example, additional method coverage prioritization first selects a test case that provides the greatest method coverage and then successively adds test cases that cover the most as-yet-uncovered parts (Do et al. 2006).
- *The Use of Information about Code Modifications:* The amount of change that has been made in a method or basic block can be incorporated into the prioritization technique (Do et al. 2006).
- *Other Dimensions:* Researchers suggest other dimensions such as test cost estimates, fault severity estimates, estimates of fault propagation probability, test history information, and usage statistics (Do et al. 2006).

Even though there are many TCP methods in the literature, there is no up-to-date systematic review, specifically a systematic mapping study, which shows the current state of the TCP research area. Systematic mapping is a methodology that is used to categorize the primary research papers in a research area. The availability of a systematic mapping study on the TCP problem can change the research perspective in this area, and little researched areas can be identified (Kitchenham et al. 2009). We believe that a mapping study of this area is timely due to the growing body of knowledge on TCP problems. The findings of this paper can be used by both academics and practitioners. Practitioners and tool vendors can identify the state-of-the-art techniques to prioritize test cases and implement them in their tools. There is one review Yoo and Harman (2012) in this area, but our study differs from this review in the following ways:

- *Scope:* While Yoo and Harman's (2012) study is a traditional review study, our paper is a systematic mapping study.
- *Time frame:* We evaluated papers published in 2001–2011, but Yoo and Harman's study only covers papers published until 2009. Therefore, this study is more contemporary.
- *Comprehensiveness:* Yoo and Harman's (2012) study consisted of 47 papers, which were not selected systematically. In contrast, this systematic mapping study used 120 papers in total and is therefore more comprehensive.

Also, the discrepancy between this review study and the study of Yoo and Harman (2012) is partly due to the fact that we included papers for which TCP is a secondary study or a tool (those belonging in "Other" category in Sect. 3.3.2).

The objective of this systematic mapping study is to examine which aspects of the test case prioritization have been studied in the most detail, determine what kind of techniques have been widely used, provide a basis for the improvement of test case prioritization research, and evaluate the current trends of this research area. For this reason, we conducted a systematic mapping study, a technique used to identify, select, synthesize, summarize, and assess all relevant studies related to a research topic. Systematic mapping studies and systematic literature reviews are widely used in medicine but are also common in other sciences such as sociology, psychology, and, recently, software engineering (Kitchenham and Charters 2007). These types of studies are the main methods of synthesis for Evidence-Based Software Engineering (EBSE), which applies an evidence-based approach to software engineering.

This paper systematically reviews 120 journal articles and conference papers on test case prioritization to evaluate the current progress of the research and to direct future research on this problem. This systematic mapping study does not describe all the test case prioritization techniques for practitioners in detail. The difference between a systematic mapping study and a systematic literature review is discussed in Sect. 2. We posed the nine research questions shown in Table 1, and these questions helped us to collect the necessary information from papers in our review process. Research questions from Jorgensen and Shepperd's (2007) paper are generic enough for software engineering problems, and therefore, we adapted these questions while designing research questions for this paper. We also used some of Jorgensen and Shepperd's (2007) research questions in our previous systematic review study (Catal and Diri 2009).

The contribution of this paper is twofold. First, to the best of our knowledge, this is the first systematic mapping study on test case prioritization research. Second, we classified TCP papers with respect to research topic, test case prioritization approach, research approach, study context, and dataset as performed in Jorgensen and Shepperd's (2007) study, using the classification of Yoo and Harman (2012). The classification mechanism used in this paper is not restricted to the prioritization techniques, as we also consider study context, research approach, and dataset. The classification of papers based on several factors is crucial for a good systematic mapping study.

This paper is organized as follows: Sect. 2 describes the background and related work. Section 3 reports the systematic mapping process and results. Section 4 provides conclusions and suggests issues for future research on test case prioritization. While Section References shows the papers used throughout this paper, "Appendix 1" lists the 120 papers included in this systematic mapping study. In referring to a paper in "Appendix 1" by citation number, the reference style [[ref#]] is used. "Appendix 2" reports the number of

Table 1 Research questions and main motivation

Research questions	Main motivation
RQ1: What are the most investigated prioritization methods?	Identify trends and opportunities for TCP prioritization methods.
RQ2: What are the most investigated TCP research topics?	Identify which research topics are widely used in TCP papers.
RQ3: What are the most frequently applied research methods?	Identify possible shortcomings of research methods used in the TCP papers.
RQ4: What is the distribution of study context in TCP papers?	Provide recommendations, if necessary, for the change of use of the projects' data.
RQ5: How many researchers are there who have a long-term interest in test case prioritization?	Assess the vulnerability of test case prioritization research; for example, having few researchers on particular topics may increase vulnerability.
RQ6: Which journal is the dominant test case prioritization journal?	Identify the most important test case prioritization journal.
RQ7: Which journals include papers on test case prioritization?	Support test case prioritization researchers with a list of journals with potentially relevant papers.
RQ8: What kind of evaluation metrics are the most frequently used for test case prioritization?	Identify the most widely used evaluation metrics for TCP papers.
RQ9: What kind of datasets are the most frequently used for test case prioritization?	Identify whether prioritization techniques are repeatable or not by examining the usage of public datasets.

papers identified in each journal, and "Appendix 3" details the categories of each classification property.

2 Background and related work

In this section, the problem of test case prioritization is briefly discussed and the methodology used, called a systematic mapping study, is introduced. We also discuss the difference between a systematic literature review (SLR) and a systematic mapping study.

2.1 Test case prioritization

Changes in a version of software may obstruct previously existing components of the software. The new components may also interact incorrectly or interact in an unknown manner with existing components. Therefore, regression testing is performed to ensure that changes do not adversely affect the existing unchanged components. TCP is one of the techniques used to improve the cost-effectiveness of regression testing. Elbaum et al. (2000) defined the test case prioritization problem as follows:

Given T, a test suite; PT, the set of permutations of T; and f, a function from PT to the real numbers.

Problem Find $T' \in PT$ such that

$$(\forall T'')(T'' \in PT)(T'' \neq T')[f(T') \geq f(T'')]. \tag{1}$$

Elbaum et al. (2000) introduced the average percentage of faults detected (APFD) metric to evaluate the effectiveness of prioritization techniques. It ranges from 0 to 100, with a higher APFD metric implying a better fault detection rate. Yoo and Harman (2012) published a survey on regression test minimization, selection, and prioritization. They investigated 47 TCP papers published between 1999 and 2009 and classified the existing research on prioritization into the following groups:

- *Coverage-based prioritization:* The assumption is that the maximization of structural coverage will increase the chance of the maximization of fault detection. For example, if a test case (test case A) covers more statements, basic blocks, or methods compared to another test case (test case B), more faults can be detected with test case A because of the greater number of statements, basic blocks, or methods exercised.
- *Distribution-based approach:* It prioritizes test cases based on the distribution of the profiles of test cases. Clustering test cases show that similar test cases are redundant and isolated clusters may cause failures.
- *Human-based approach:* A human tester is used in this kind of approach. Prioritization is based on the comparisons made by the human tester.
- *Probabilistic approach:* Probabilistic theory is applied. A selection probability is assigned to each test case in a test suite, and the execution history of each test case is taken into account. Bayesian network-based approaches are shown in this category.
- *History-based approach:* Historical data such as execution history or change information are used in this approach.
- *Requirement-based approach:* Requirement properties are used. For example, factors such as customer-assigned priority on requirements, requirement volatility,

and developer-perceived implementation complexity of requirements can be applied [[85]].
- *Model-based approach:* Different models such as UML (Unified Modeling Language) sequence or activity diagrams are used instead of code blocks. The UML is the standard visual language for modeling the software systems.
- *Cost-aware approach:* The costs of test cases are taken into account because the costs of all the test cases cannot be equal.
- *Other approaches:* In this category, the authors showed other techniques such as interface-contract mutation, relevant slices, and call-tree paths.

While Yoo and Harman's survey is based on 47 TCP papers, our study is derived from the analysis of a systematic search of journal papers and conference proceedings, which led to the identification of 120 TCP papers. The survey in Yoo and Harman (2012) did not describe a selection process or state inclusion/exclusion criteria. Therefore, our paper is more comprehensive and systematic.

Engström et al. (2008) performed a systematic review on Regression Test Selection (RTS) techniques. They reported that there are 32 empirically evaluated techniques on RTS, that there is no approach for selecting one superior technique, and that the empirical evidence for differences between the techniques is sometimes contradictory. Also, Engström et al. (2010) evaluated 28 techniques for RTS and concluded, based on many varying factors, that no technique is superior. Engström's (2010) first study is a subset of the second (Engström et al. 2008).

2.2 Systematic mapping study

Evidence-Based Software Engineering (EBSE) collects and evaluates empirical evidence based on the research questions from primary studies in order to make better decisions related to software engineering and software development. This concept was introduced in 2004 by Kitchenham et al. (2004). The aggregation of empirical results is mostly performed with systematic literature reviews (SLRs) (Kitchenham et al. 2010). In addition to the SLRs, researchers can also use mapping studies (Petersen et al. 2008) in the context of EBSE. The difference between a mapping study and an SLR is that mapping studies identify all research on a specific topic, while SLRs address specific research questions (Kitchenham et al. 2010). Mapping studies classify other studies with respect to several properties and categories. Kitchenham et al. (2010) describe the differences between mapping studies and SLRS as follows (Kitchenham et al. 2010):

- *Research question:* The research question (RQ) of an SLR is specific and related to the outcomes of empirical studies. The RQ of a mapping study is general and related to research trends.
- *Search process:* While the search process of an SLR is defined by the research question, the search process of a mapping study is defined by the research topic.
- *Search strategy requirements:* While the search strategy requirements of an SLR are extremely stringent, the search strategy requirements of a mapping study are less stringent. All previous studies must be found for SLRs.
- *Quality evaluation:* Quality evaluation is crucial for SLRs, but it is not essential for mapping studies.
- *Results:* Results of SLRs are answers to specific research questions. Results of mapping studies are categories of papers related to a research topic.

3 Research method

We conducted this systematic mapping study on the test case prioritization topic, identified papers related to the prioritization, and investigated papers from February 1, 2011, to April 1, 2011.

The review process used in this study included the following phases:

- Planning,
- Execution, and
- Reporting.

3.1 Phase 1—planning

At this phase, we specified research questions, search strategy, inclusion and exclusion criteria, and data extraction and synthesis method.

3.1.1 Research questions (RQ)

We established nine RQs to identify the primary studies that explore test case prioritization. Research questions and main motivation are shown in Table 1.

3.1.2 Search strategy

We identified the following keywords and their synonyms for searching the relevant papers from the electronic databases: *test, tests, testing, case, cases, suite, suites, prioritization, and prioritizing*. We used the Boolean operator OR to link the synonyms of the main terms and then combined the main terms using a Boolean operator AND. The final search string is given as follows:

Search string:

[("test" OR "tests" OR "testing") **AND** ("case" OR "cases" OR "suite" OR "suites") **AND** ("prioritization" OR "prioritizing")].

We selected the electronic databases shown in Table 2 to execute this search string and searched the metadata that include the abstract, indexing terms, and bibliographic citation information in the electronic databases. The search process was performed as an automatic search instead of a manual search. When we found the same paper in multiple databases, we used one of them for the review. We also inspected the reference sections of these selected papers to determine whether there were any research papers that were suitable for inclusion but did not contain the key strings used in searching the databases. However, we did not find any such papers, which did not contain the key strings, that were suitable for inclusion.

Table 2 Selected databases

Source	Location
IEEE Explore	http://ieeexplore.ieee.org
ACM Digital Library	http://portal.acm.org
ScienceDirect	http://www.sciencedirect.com
SpringerLink	http://www.springerlink.com
Wiley	http://www.wiley.com

3.1.3 Inclusion and exclusion criteria

This review included papers published between 2001 and 2011. Papers from peer-reviewed journals, conferences, and workshops were considered. We excluded studies that were not related to test case prioritization in the context of software engineering. Summaries of tutorials, panels, and poster sessions, as well as editorials and prefaces were also excluded. We did not remove duplicate publications if they have different experimental sections. However, if only a small part of the paper was extended, we did not add these additional papers to the list. The inclusion criteria (IC) and exclusion criteria (EC) of our systematic review are as follows:

IC1 The primary studies on the field of test case prioritization (TCP).

IC2 The secondary studies on the field of TCP.

IC3 The primary studies that compare two or more TCP techniques.

IC4 The primary studies that utilize TCP techniques to develop efficient methods for different software engineering problems such as test suite reduction.

IC5 The primary studies that propose new evaluation metrics ("Appendix 3," Category 6) for the TCP problem.

IC6 The primary studies that explain the development of a software tool that can be used for TCP.

EC1 The studies that do not address the test case prioritization problem.

EC2 The studies that are written in a language other than English.

EC3 The papers that cannot be accessed in full text.

EC4 Summaries of tutorials, panels, and poster sessions, as well as editorials and prefaces.

3.1.4 Classification of papers

Classification was first performed separately by two authors, and then, we checked the classification results. We discussed when there is a different label for the papers, and finally, we decided on the label in consensus. The papers were classified according to the properties and categories listed in Table 3. "Appendix 3" explains the categories shown in this table. The inter-rater percentage agreement was higher than 90 % in all properties. Most of the categories are non-exclusive. For example, one paper may apply more than one prioritization approach.

3.1.5 Threats to validity

We did not search papers based on issue-by-issue, manual reading of titles of all published papers in journals, as Jorgensen and Shepperd (2007) did. The reason why we did not take this approach is related to practical concerns such as workload. Therefore, we may have excluded some TCP papers in some journals or conference proceedings.

We did not exclude conference papers, but Jorgensen and Shepperd (2007) excluded them because their inclusion would increase the workload for the study. They included 304

Table 3 Classification of papers

Property	Categories
Research approach	Theory, survey, experiment, personal experience, review, development of prioritization method
Study context	Datasets from student projects, datasets from industrial projects, not relevant
Dataset type	Public, private, partial, none
Prioritization approach	Coverage-based, distribution-based, human-based, probabilistic, history-based, requirements-based, model-based, cost-aware, other approaches, and none
Research topic	Prioritization method, measures of prioritization performance, comparison of prioritization methods, other methods
Evaluation metrics	APFD (Average Percentage of Faults Detected), ASFD (Average Severity of Faults Detected), TPFD (Total Percentage of Faults Detected), APFD(c) (Average Percentage of Faults Detected per Cost), NAPFD (Normalized APFD), RP (Most Likely Relative Position),CE (Coverage Effectiveness), none, others

journal papers for their systematic review study. However, the history of test case prioritization studies is not as old as that of cost estimation studies.

3.2 Phase 2—execution

In this phase, we performed several searches on electronic databases by using the search string. We initially found 202 primary studies on test case prioritization. Then we read the title and abstract sections of these papers and applied our selection criteria. From this step, we selected 120 studies for further reading. Later, we read these papers in full and applied our inclusion/exclusion criteria. We did not remove any papers after reading them in full. Table 4 shows the number of obtained papers, the number of studies included, the rate index (the ratio between the included studies of a database and the total number of studies included in the systematic review), and database names.

IEEE and ACM sources included 85 % of reviewed papers, and therefore, they seem the most useful sources for the TCP problem. However, 42 % of obtained IEEE papers were not related to the TCP problem, and therefore, we had to remove them from our investigation list.

3.3 Phase 3—reporting

In this phase, the analytical results of the systematic mapping study are discussed based on the research questions. Before providing answers to RQs, some of the statistical results of the mapping study are explained. Twenty-one journal papers and 99 papers in conference

Table 4 Sources and obtained and included papers

Database	Obtained	Included	Rate index (%)
IEEE explore	130	76	64
ACM digital	32	25	21
Science direct	15	5	4
Springer link	19	10	8
Wiley	6	4	3

proceedings have been evaluated in this study. Figure 1 is a curve that plots the publication year on the horizontal axis and the number of papers published in that year on the vertical axis for papers in review. It can be seen that, after 2006, the number of papers doubled (i.e., 7 papers in 2006 compared to 16 papers in 2007). According to this figure, the interest of software engineering researchers in test case prioritization seems to be increasing. Since the searches pertaining to this paper were performed at the beginning of 2011, the number of papers published in 2011 has been limited.

3.3.1 The most investigated prioritization methods (RQ1)

By using this research question, the objective was to document and categorize the available prioritization techniques. If one technique is used in less than three papers, that technique is shown in the category called *others*. Also, one paper may apply more than one prioritization technique. Figure 2 shows the distribution of papers based on different test case prioritization methods. We used the same classification scheme, which was explained in Sect. 2.1, suggested by Yoo and Harman (2012).

This distribution suggests that:

- Coverage-based prioritization methods dominate. Most of the papers introduced coverage-based techniques or compared the proposed technique with the coverage-based methods. In addition, greedy-based and model-based approaches were generally preferred, yet the use of coverage-based prioritization methods is still approximately four times higher than that of model-based methods.
- There exists a large variety of prioritization techniques used to rank the test cases.
- The proportion of papers on model-based approaches is increasing, but at a slow rate. Between 2001 and 2009, only four papers proposed or used model-based techniques. However, three papers were published in 2009 on model-based approaches, and three more in 2010.

3.3.2 The most investigated TCP research topics (RQ2)

We identified four research topics that are related to test case prioritization:

- *Prioritization method,*
- *Comparison of prioritization methods,*
- *Measures of prioritization performance,* and

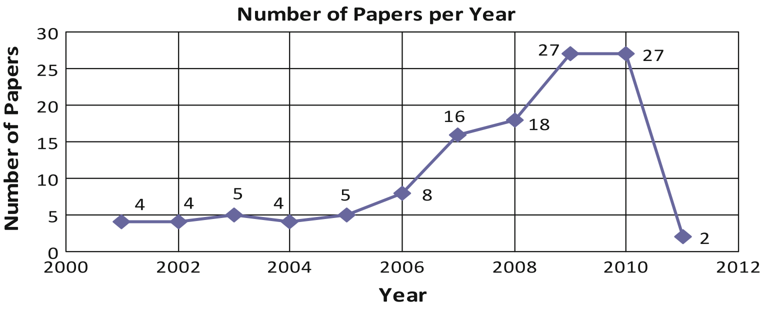

Fig. 1 Number of papers per year in review [incomplete data for year 2011]

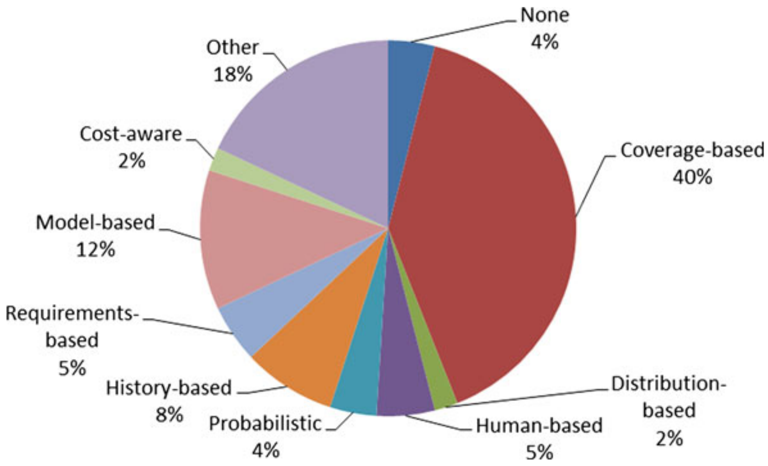

Fig. 2 Distribution of prioritization methods

- *Others*: It includes the following topics: change effect identification, combination of prioritization and reduction methods, tool development, use of prioritization method for test case selection, use of prioritization method for software fault localization, time constraint effects, selecting the most effective prioritization method, test suite generation, test plan generation, survey on prioritization methods, review of prioritization methods, residual defects identification, test adequacy criterion, measuring the sources of variation in the prioritization, and effects of test suite granularity.

Figure 3 shows the distribution of research topics suggesting that:

- Even though 18 different types of research topics were identified during the analysis, only three of them were dominant. The other topics in the group have been categorized as "others." The percentage of these three research topics is 82 %. The most common research topic, with 59 % of the papers, is the development of prioritization methods.

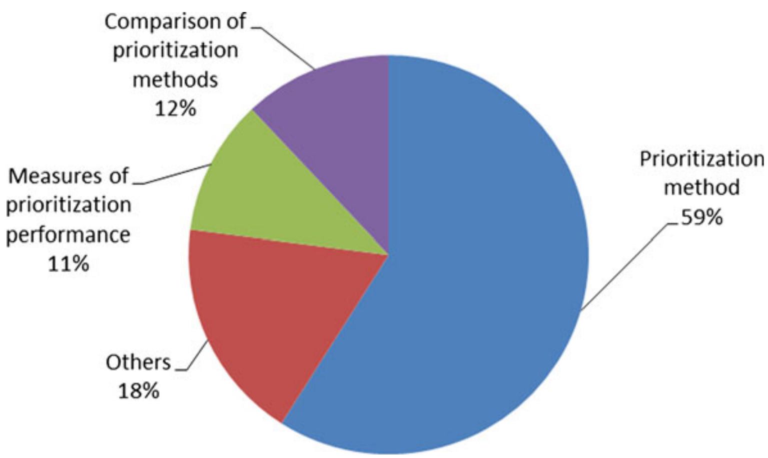

Fig. 3 Distribution of research topics

- 11 % of papers suggested and validated new evaluation metrics. These criteria are discussed in "Appendix 3"—Category 6.
- At only 12 %, the proportion of papers on the *comparison of prioritization methods* is less than expected, implying that more studies on this issue can be conducted. The authors advocate for more comparative studies since more research is needed to improve convergence across studies, and a framework for comparative TCP experiments can also be beneficial.

3.3.3 The most frequently applied research methods (RQ3)

The papers were classified according to the applied research method, which included theory, survey, experiment, personal experience, review, and development of prioritization method. These research methods are explained in "Appendix 3"—Category 1. Figure 4 presents the distribution of papers applying the different research approaches. This figure suggests that:

- The majority of the papers (62 %) reported the development of a new prioritization method, with the next-highest proportion being the experiments (27 %). This shows that researchers are primarily trying to focus on the development of new prioritization methods. However, around 25 % of papers (19 out of 75) reporting the development of prioritization methods did not use any evaluation metrics to check the effectiveness of their proposed prioritization method.
- There is a need to conduct surveys that can report the current situation of test case prioritization in software development organizations.

3.3.4 Distribution of study context in TCP papers (RQ4)

The papers were categorized based on the following study contexts: datasets from student projects, datasets from industrial projects, and not relevant. These study contexts are defined in "Appendix 3"—Category 2. The term "project" refers to the code and test cases

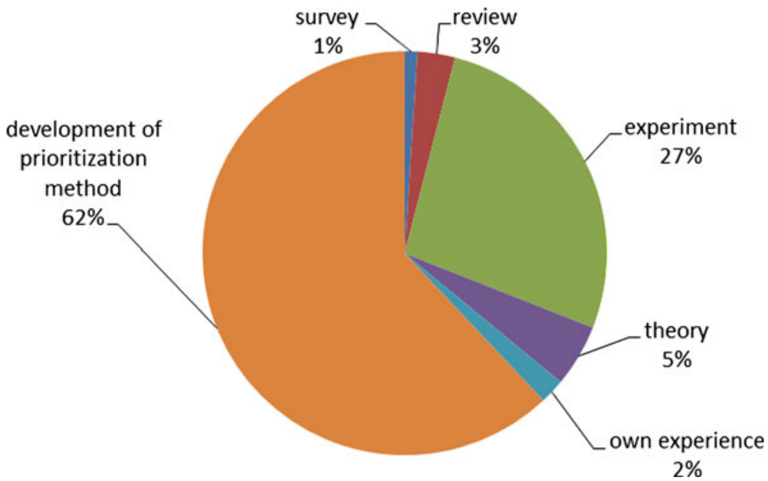

Fig. 4 Distribution of research methods

developed in projects. Figure 5 shows the distribution of papers based on the study context. This table suggests that:

- The proportion of papers that use datasets from industrial projects to evaluate the performance of the proposed techniques is 64 %, similar to the ratio obtained in Jorgensen and Shepperd's study (2007) that focused on the software cost estimation problem. They stated that 71 % of cost estimation papers used datasets from industrial projects and that this ratio shows the realism of the studies. We observed a similar ratio in test case prioritization papers as well. However, most of these datasets from industrial projects are related to the programs located in SIR repository (Software-artifacts Infrastructure Repository) (Do et al. 2005), and even though they were written by professional programmers, their size and complexity may not represent the real industrial problems. Therefore, researchers should not only take into account the type of the projects (i.e., datasets from industrial projects or datasets from student projects), but also the size and complexity of the projects.
- Twenty-one percentage of the papers (25 papers) did not use any project data. However, three of them are review papers, one is a survey paper, and two discuss the development of a software tool. Therefore, 19 papers need to be validated. We suggest that researchers should not only propose a new technique, but also validate it with project data.

3.3.5 Researchers who have a long-term interest in TCP (RQ5)

The present study includes 120 research papers, and it has found that there were 219 authors involved in writing all these papers. According to Table 5, the highest number of active researchers involved in publishing journal papers was recorded in 2009, with a steady increase every year in the number of active researchers who published papers in the conference proceedings from the year 2007 onward. It seems that the number of active authors in this area has gradually increased in the last 5 years, especially in the papers of conference proceedings.

However, during the analysis it was found that there are very few authors with a long-term focus on test case prioritization. The majority of the authors (152 out of 219 authors, i.e.,

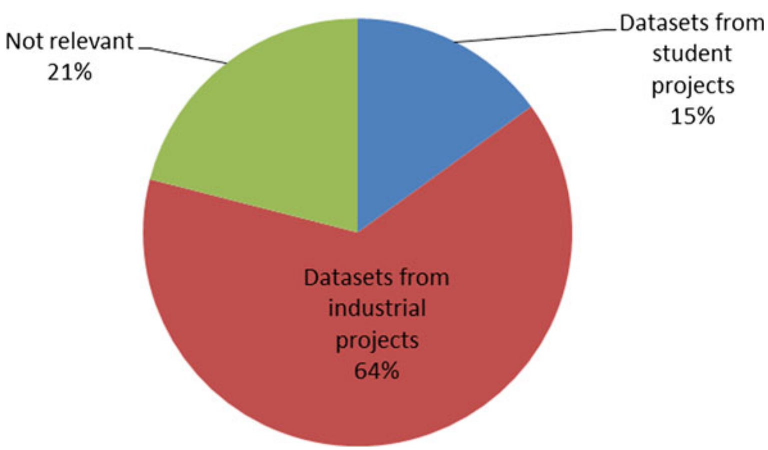

Fig. 5 Distribution of study context

69.4 %) have written only one paper in the last 10 years, as shown in Table 6. This number may be affected by several factors, such as the number of student authors, but we also observed that there are some researchers who have only written one paper. There are few authors (only 8 authors, i.e., 3.6 %) who have written 5 or more research papers on this topic, and only one of these eight authors has been actively involved in writing more than 10 papers. Two of these eight researchers with long-term focus are not currently active, and their last paper was published in the year 2004, which means there are only six researchers with a long-term focus currently active in this area. Naturally, it is imperative for high-quality research in any area that the number of researchers with a long-term focus should be higher as compared to researchers with a short-term focus. Researchers who published more than four papers are shown in Table 7. This table shows that 43 % of journal papers investigated in this systematic mapping study were written by Gregg Rothermel and his colleagues. Also, we showed the number of citations of the papers for each researcher in Table 7. We accessed the Google Scholar website (http://scholar.google.com) on March 11, 2012, to retrieve the number of citations.

3.3.6 The dominant test case prioritization journal (RQ6)

IEEE Transactions on Software Engineering (TSE) is the most dominant test case prioritization journal with respect to the number of studies published (See "Appendix 2"). In addition, we investigated the number of citations made to TSE by other researchers for each journal paper. To identify the number of citations for each paper, once again the Google Scholar website (http://scholar.google.com) was used. On the basis of that information, it is found that four papers from the TSE journal have been cited over 100 times (137, 332, 237, and 135 citations) and none of the papers published in journals other than TSE has over 100 citations. Even though one of the papers from TSE was only published in 2007, it now has 135 citations (accessed on April 1, 2011). According to this citation analysis and the proportion of papers published in this journal, TSE is the dominant TCP journal.

3.3.7 Journals that include papers on TCP (RQ7)

Papers were found on test case prioritization in 10 journals. "Appendix 2" shows all of these journals. Figure 6 shows the journals that published more than one paper. These four

Table 5 Distribution of active authors

Years	# of authors in journals	# of authors in conferences
2001	4	6
2002	3	7
2003	7	5
2004	6	6
2005	0	13
2006	5	12
2007	3	38
2008	2	45
2009	11	56
2010	9	66
2011	7	0

Table 6 Distribution of paper counts

# of authors	# of papers
152	1
35	2
14	3
10	4
1	5
2	6
2	7
2	8
1	16

Table 7 Distribution of paper and citation counts per author

Author name	# of articles	# of proceedings	Total	# of citations
Gregg Rothermel	9	7	16	1346
Sebastian Elbaum	5	3	8	702
Gregory M. Kapfhammer	0	8	8	203
Hyunsook Do	3	4	7	198
Wing-Kwong Chan	1	6	7	71
Alexey Malishevsky	4	2	6	659
T. H. Tse	1	5	6	91
Renee C. Bryce	2	3	5	120

journals include 71 % of all the identified journal papers on test case prioritization. However, only sufficing to these journals may cause one to miss other significant papers as they may only be published in conference proceedings and nowhere else.

3.3.8 *The most frequently used evaluation metrics for TCP (RQ8)*

There are several papers introducing new types of evaluation metrics, such as APFD (Average Percentage of Faults Detected), CE (Coverage Effectiveness), and RP (Most Likely Relative Position). "Appendix 3"—Category 6 explains these evaluation metrics. The distribution of these metrics is presented in Fig. 7, indicating that:

- Thirty-four percentage of the papers have used the APFD evaluation metric making it the most dominant of all.
- Twenty-six percentage of the papers did not employ any evaluation metrics. However, six of them are review, tool, and survey papers. Therefore, 21 % of the papers did not use any evaluation metric even though they have been written on the subject of TCP and are not review papers. It can be argued that researchers should evaluate the effectiveness of their approaches with some of the evaluation metrics shown in "Appendix 3"—Category 6.
- Twenty-three different types of evaluation metrics exist in the group called *others*. These metrics were used two times or less in the papers and consequently have not been considered in the major categories. Some of these metrics include total testing effort

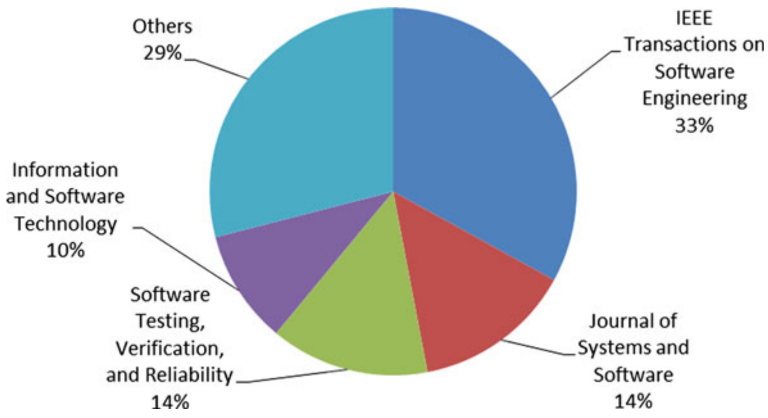

Fig. 6 Journals that include papers on TCP

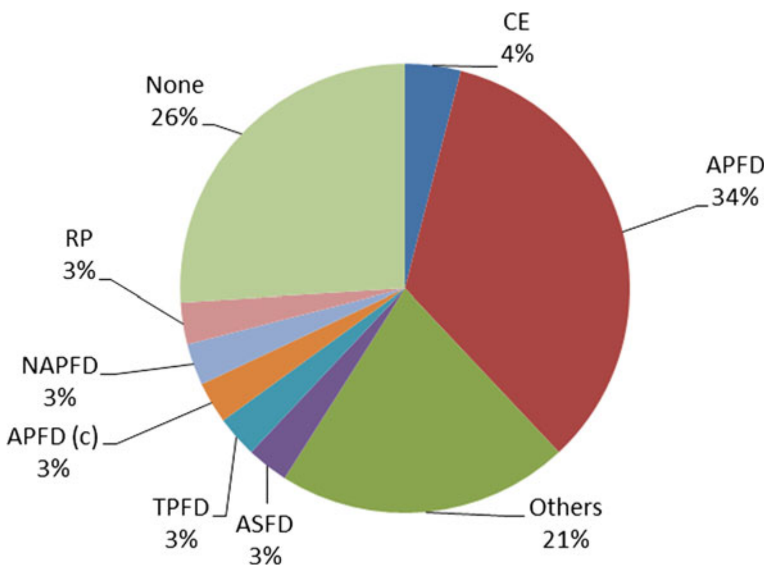

Fig. 7 Distribution of evaluation metrics

and average fault age [[3]], APRC Improved [[6]], APDP (Average Percentage of Damage Prevented) [[9]], cost–benefit model [[12]], faults exposure ratio [[17]], execution time [[18]], and expense [[52]]. None of these items fit into the major groups because they are calculated based on different mathematical formulas.

3.3.9 The most frequently used kind of datasets (RQ9)

It is common knowledge that software engineering experiments are required to be repeatable and verifiable, which means that public datasets are most preferable for evaluation. Otherwise, it is not possible to access the data and to ensure whether the experiment is repeatable or not. With this research question, we are focusing on whether prioritization techniques are repeatable or not, by examining the usage of public datasets. The reproducible research has many benefits such as helping the new researchers enter in the field (Kapfhammer 2011).

While there are certain benefits associated with using this type of dataset, there are some drawbacks as well. For example, an industry-leading company may not prefer to make its dataset public due to confidentiality issues. In this case, there are useful options for data sharing that do not require a public data set. A recent technology called *Dataverse Network* (King 2007) supports restricted access to the dataset if distributing the dataset to the public is not feasible. *Dataverse Network* system not only gives recognition to the author, distributor, and publisher, but also it generates a persistent data citation by using a global persistent identifier and Universal Numerical Fingerprint (UNF) to verify the dataset. Companies that have concerns about making datasets public may investigate the data sharing features of *Dataverse Network* system.

The papers were categorized based on the type of dataset used: *public, private, partial,* and *none*. Publicly available datasets are referred to as public datasets. The private ones are not distributed freely, and they belong to a software company or an academic research that does not release its datasets. Partial datasets are created based on open source projects, but are not distributed to the community as public datasets. If there is no information about the dataset in the paper, this kind of dataset is called *none*. Figure 8 shows the distribution of papers using different types of datasets indicating that:

- The proportion of papers that use public datasets to evaluate the performance of the proposed techniques or to conduct their experiments is only 38 %. The number of studies that used private datasets for evaluation is around the same (36 %)—a trend that is not encouraging and which suggests that more papers should be based on public datasets. The applicability of the studies reporting the result based on evaluation using private datasets is worthy of reconsideration as these studies cannot be repeated by other researchers.
- Almost all the papers that employed public datasets for evaluation also used datasets from industrial projects (45 out of 46). This gives reassurance to people intending to use the results, as usage of public datasets combined with datasets from industrial

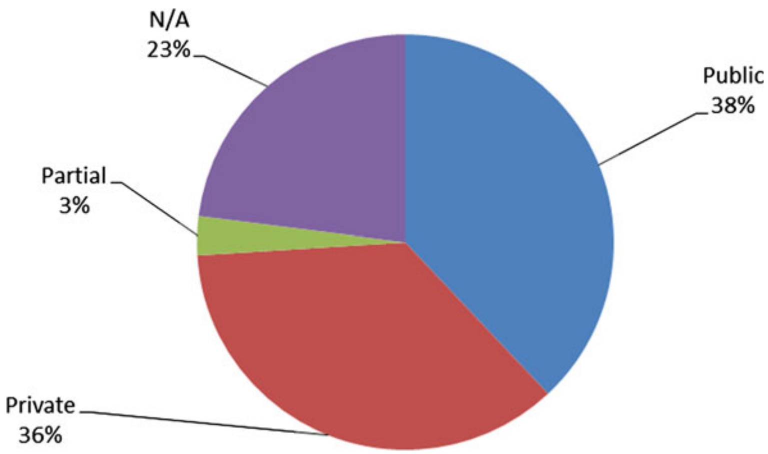

Fig. 8 Distribution of dataset types

projects for evaluation makes the results of the paper more reliable and trustworthy. Only one study that used public datasets employed students or involved simple projects.

- Twenty-three percentage of the papers (27 papers) did not use any datasets. However, three of these are review papers, one is a survey paper, and two discuss the development of a software tool. The actual number of studies that did not use any datasets is 21, which is over 17 %. One can infer from these figures that the usefulness of the results presented by the papers that are not validated can be disputed and that, in light of this observation, professionals may feel apprehensive about trusting the contributions of such papers.

4 Conclusions and future works

The present paper investigated 120 test case prioritization papers published in conference proceedings and journals with a systematic mapping process. The objective of this systematic mapping study is to determine which aspects of test case prioritization have been studied in most detail and to provide a basis for the improvement of test case prioritization research. We answered nine research questions, identified at the beginning of this study, and explained them in a detailed fashion.

We suggest the following changes in test case prioritization research:

- *Give preference to public datasets over proprietary datasets.*
  The proportion of papers that use public datasets is only 38 %, while the number of studies that utilize private datasets stands at about 36 %. This trend appears to be not satisfactory, and more papers need to be based on public datasets. The applicability of the studies reporting results based on evaluation using private datasets is questionable and, as such, subject to discussion as these studies cannot be repeated by other researchers.
- *Develop more model-based prioritization methods.*
  The proportion of papers on model-based approaches is increasing, but at a slow rate. Between 2001 and 2009, only four papers proposed or used model-based techniques. However, three papers were published in 2009 on model-based approaches, and three more were published in 2010. The execution cost of model-based TCP methods is less than that of code-based TCP methods because models can be analyzed faster than the source code. Also, early TCP (which produces feedback more quickly) can be achieved by using model-based TCP techniques since such models are produced before the source code is implemented. In addition to these special characteristics, some researchers have demonstrated that model-based TCP techniques can outperform code-based TCP techniques (Korel and Koutsogiannakis 2009).
- *Conduct more studies on the comparison of prioritization methods.*
  Only 16 papers out of 120 compared the performance of several prioritization methods. The most common research topic, representing 59 % of the papers, is the development of prioritization methods. To find the best prioritization approach, more studies are required to be carried out on the comparison of prioritization methods as new methods are continuously being developed.
- *Always evaluate the effectiveness of the proposed technique with well-known evaluation metrics, and compare the performance with the existing methods.*

The majority of the papers (62 %) reported the development of a new prioritization method, but 25 % of that proportion did not use any evaluation metric to check the effectiveness of the proposed approach. With this in mind, new approaches should be empirically validated and compared with other prominent prioritization methods using well-known evaluation metrics. Otherwise, it is not possible to judge the effectiveness of the proposed approach.

- *Publish reviews and systematic review papers on test case prioritization.*
  Only three review papers were found during the literature search—an insignificant number—implying that up-to-date reviews and systematic review papers are needed, which can report the current state of the art and state of the practice in the TCP research area.
- *Use datasets from industrial projects that represent the real industrial problems.*
  Sixty-four percentage of the papers investigated in this research used datasets from industrial projects. Industrial companies would like to see industrial projects applied in these research papers and once they see such actions, they may be more inclined to start using the proposed approaches. Apart from that if a student project is large enough to yield meaningful results, it can be used in several thought-provoking studies as well. However, if the student project is too small, then the results may not be meaningful.

The future work includes performing Systematic Literature Review (SLR) study on the effectiveness of some of the TCP techniques identified in this study. For example, we intend to conduct a SLR study to investigate the effectiveness of genetic algorithms for TCP. Also, new techniques based on the genetic algorithms can be developed after this SLR study.

Acknowledgments The authors would like to thank the Associate Editor and the four reviewers for their constructive comments on the paper.

Appendix 1: References for the 120 Included Papers [[1–120]]

The actual mapping of individual papers in each category is shown at the end of the reference. Classification properties are given with C1 (Research Approach), C2 (Study Context), C3 (Dataset Type), C4 (Prioritization Approach), C5 (Research Topic), and C6 (Evaluation Metrics) codes, and the values indicate their subcategories. "Appendix 3" details the subcategories of each property. For example (C1:1, C2:2, C3:1, C4:2, C5:1, C6:2), code at the end of a reference indicates that the research approach is theory, the dataset is from the industrial project, the dataset type is public, the prioritization approach is change-based, the research topic is prioritization method, and the evaluation metric is ASFD.

References from this Appendix must be cited using the [[ref#]] reference style. Please report any problems regarding this classification to: c.catal@iku.edu.tr.

[1] David Leon and Andy Podgurski. 2003. A Comparison of Coverage-Based and Distribution-Based Techniques for Filtering and Prioritizing Test Cases. In *Proceedings of the 14th International Symposium on Software Reliability Engineering* (ISSRE '03). IEEE Computer Society, Washington, DC, USA, 442–. **(C1:3, C2:2, C3:1, C4:1&2, C5:2, C6:1)**

[2] Hyunsook Do and Gregg Rothermel. 2005. A Controlled Experiment Assessing Test Case Prioritization Techniques via Mutation Faults. In *Proceedings of the 21st IEEE International Conference on Software Maintenance* (ICSM '05). IEEE

Computer Society, Washington, DC, USA, 411–420. **(C1:3, C2:2, C3:1, C4:1, C5:2, C6:1)**

[3] Jung-Min Kim and Adam Porter. 2002. A history-based test prioritization technique for regression testing in resource constrained environments. *In Proceedings of the 24th International Conference on Software Engineering* (ICSE '02). ACM, New York, NY, USA, 119–129. **(C1:4, C2:1, C3:2, C4:5, C5:1, C6:9)**

[4] Xiang Chen, Qing Gu, Xinping Wang, Ang Li, Daoxu Chen. 2009. A Hybrid Approach to Build Prioritized Pairwise Interaction Test Suites. *In Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on* pp. 1–4. **(C1:4, C2:3, C3:4, C4:1, C5:1, C6:8)**

[5] Siavash Mirarab and Ladan Tahvildari. 2007. A Prioritization Approach for Software Test Cases based on Bayesian Networks. *In Proceedings of the 10th International Conference on Fundamental Approaches to Software Engineering* (FASE'07), Springer-Verlag, Berlin, Heidelberg, 276–290. **(C1:4, C2:2, C3:1, C4:4, C5:1, C6:1)**

[6] Sihan Li, Naiwen Bian, **Zhenyu Chen**, Dongjiang You, Yuchen He. 2010. A Simulation Study on Some Search Algorithms for Regression Test Case Prioritization. *In International Conference on Quality Software* (*QSIC*) *2010*, pp. 72–81. **(C1:3, C2:3, C3:4, C4:1, C5:2, C6:9)**

[7] Bo Jiang, Zhenyu Zhang, W. K. Chan, and T. H. Tse. 2009. Adaptive Random Test Case Prioritization. *In Proceedings of the 2009 IEEE/ACM International Conference on Automated Software Engineering* (ASE '09). IEEE Computer Society, Washington, DC, USA, 233–244. **(C1:4, C2:2, C3:1, C4:1, C5:1, C6:1)**

[8] Xiaoqing Bai, C. Peng Lam, Huaizhong Li. 2004. An Approach to Generate the Thin-Threads from the UML Diagrams. *In 28th Annual International Computer Software and Applications Conference* (*COMPSAC'04*), vol. 1, pp. 546–552. **(C1:4, C2:3, C3:4, C4:9, C5:4, C6:8)**

[9] Heiko Stallbaum, Andreas Metzger, and Klaus Pohl. 2008. An Automated Technique for Risk-based Test Case Generation and Prioritization. *In Proceedings of the 3rd International Workshop on Automation of Software Test* (*AST '08*). ACM, New York, NY, USA, 67–70. **(C1:4, C2:1, C3:1, C4:9, C5:1&3, C6:9)**

[10] Sejun Kim and Jongmoon Baik. 2010. An Effective Fault Aware Test Case Prioritization by Incorporating a Fault Localization Technique. In *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement* (*ESEM '10*). ACM, New York, NY, USA, Article 5, 10 pages. **(C1:4, C2:2, C3:1, C4:1, C5:1, C6:1)**

[11] Adam M. Smith and Gregory M. Kapfhammer. 2009. An Empirical Study of Incorporating Cost into Test Suite Reduction and Prioritization. *In Proceedings of the 2009 ACM symposium on Applied Computing* (SAC '09). ACM, New York, NY, USA, 461–467. **(C1:4, C2:1, C3:2, C4:1, C5:1, C6:7)**

[12] Hyunsook Do, Siavash Mirarab, Ladan Tahvildari, and Gregg Rothermel. 2008. An Empirical Study of the Effect of Time Constraints on the Cost-benefits of Regression Testing. *In Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering* (SIGSOFT '08/FSE-16). ACM, New York, NY, USA, 71–82. **(C1:3, C2:2, C3:1, C4:1&4, C5:2, C6:9)**

[13] Siavash Mirarab and Ladan Tahvildari. 2008. An Empirical Study on Bayesian Network-based Approach for Test Case Prioritization. *In Proceedings of the 2008 International Conference on Software Testing, Verification, and Validation* (ICST

‘08). IEEE Computer Society, Washington, DC, USA, 278–287. **(C1:4, C2:2, C3:1, C4:4, C5:1, C6:1)**

[14] Sheng Huang, Yang Chen, Jun Zhu, Zhong Jie Li, and Hua Fang Tan. 2009. An Optimized Change-driven Regression Testing Selection Strategy for Binary Java Applications. *In Proceedings of the 2009 ACM Symposium on Applied Computing* (SAC ‘09). ACM, New York, NY, USA, 558–565. **(C1:4, C2:2, C3:2, C4:1, C5:1, C6:1)**

[15] Arshad Mansoor. 2010. Analytical Survey on Automated Software Test Data Evaluation. *In Proceedings of the 4th International Conference on New Trends in Information Science and Service Science* (*NISS*), pp. 580–585. **(C1:6, C2:3, C3:4, C4:10, C5:4, C6:8)**

[16] Bogdan Korel, George Koutsogiannakis, Luay H. Tahat. 2008. Application of System Models in Regression Test Suite Prioritization. *In Proceedings of the IEEE International Conference on Software Maintenance* (*ICSM*), 247–256. **(C1:4, C2:2, C3:1, C4:7, C5:1, C6:6)**

[17] Shan–Shan Hou, Lu Zhang, Tao Xie, Hong Mei, Jia-Su Sun. 2007. Applying Interface-Contract Mutation in Regression Testing of Component-Based Software. *In Proceedings of the IEEE International Conference on Software Maintenance, ICSM 2007,* pp. 174–183. **(C1:3, C2:2, C3:3, C4:9, C5:4, C6:9)**

[18] Khin Haymar Saw Hla, YoungSik Choi, and Jong Sou Park. 2008. Applying Particle Swarm Optimization to Prioritizing Test Cases for Embedded Real Time Software Retesting. *In Proceedings of the 2008 IEEE 8th International Conference on Computer and Information Technology Workshops* (CITWORKSHOPS ‘08). IEEE Computer Society, Washington, DC, USA, 527–532. **(C1:4, C2:2, C3:1, C4:1, C5:1, C6:9)**

[19] Gary Chen, Jamie Rogers. 2010. Arranging Software Test Cases through an Optimization Method. *In Proceedings of the Technology Management for Global Economic Growth PICMET 2010*, 1596–1600. **(C1:4, C2:3, C3:4, C4:9&3, C5:1, C6:8)**

[20] Xiang Chen, Qing Gu, Xin Zhang, and Daoxu Chen. 2009. Building Prioritized Pairwise Interaction Test Suites with Ant Colony Optimization. *In Proceedings of the 2009 Ninth International Conference on Quality Software* (QSIC ‘09). IEEE Computer Society, Washington, DC, USA, 347–352. **(C1:4, C2:3, C3:4, C4:9, C5:1, C6:9)**

[21] X. Qu, M. B. Cohen, and K. M. Woolf. 2007. Combinatorial Interaction Regression Testing: A Study of Test Case Generation and Prioritization. *In Proceedings of the* Intl. Conference on Software Maintenance, pages 255–264. **(C1:3, C2:2, C3:1, C4:9, C5:2, C6:1&5)**

[22] Leonardo Mariani, Sofia Papagiannakis, and Mauro Pezze. 2007. Compatibility and Regression Testing of COTS-Component-Based Software. *In Proceedings of the 29th International Conference on Software Engineering* (*ICSE ‘07*). IEEE Computer Society, Washington, DC, USA, 85–95. **(C1:4, C2:2, C3:1, C4:7, C5:1, C6:9)**

[23] Xiao Qu. 2009. Configuration Aware Prioritization Techniques in Regression Testing. *In Proceedings of the* ICSE Companion 2009, 375–378. **(C1:4, C2:2, C3:1, C4:1, C5:1, C6:5)**

[24] Xiao Qu, Myra B. Cohen, and Gregg Rothermel. 2008. Configuration-aware Regression Testing: An Empirical Study of Sampling and Prioritization. *In Proceedings of the 2008 International Symposium on Software Testing and Analysis*

(ISSTA '08). ACM, New York, NY, USA, 75–86. **(C1:4, C2:2, C3:1, C4:1, C5:1, C6:5)**

[25] Renee C. Bryce. Constructing Interaction Test Suites with Greedy Algorithms, *In Proceedings of the 20th IEEE/ACM International Conference on Automated Software Engineering* (*ASE05*), Extended Abstract—Doctoral Symposium. Long Beach, California (November 2005), pp. 440–443. **(C1:5, C2:3, C3:4, C4:1, C5:4, C6:8)**

[26] Jianjun Yuan, Zuowen Jiang. 2010. Constructing Prioritized Interaction Test Suite with Interaction Relationship. *In Proceedings of the Second International Workshop on Education Technology and Computer Science* (*etcs 2010*), vol. 3, pp. 181–184. **(C1:4, C2:3, C3:4, C4:1, C5:4, C6:8)**

[27] Fevzi Belli, Mubariz Eminov, Nida Gökçe. Coverage-Oriented, Prioritized Testing—A Fuzzy Clustering Approach and Case Study. *In Proceedings of 3rd Latin-American Symposium* (*LADC'2007*), pp. 95–110. **(C1:4, C2:2, C3:2, C4:7&1, C5:1, C6:8)**

[28] Yu-Chi Huang, Chin-Yu Huang, Jun-Ru Chang, Tsan-Yuan Chen. 2010. Design and Analysis of Cost-Cognizant Test Case Prioritization Using Genetic Algorithm with Test History. *In Proceedings of the IEEE 34th Annual Computer Software and Applications Conference* (*compsac 2010*), pp. 413–418. **(C1:4, C2:2, C3:1, C4:5&1, C5:1, C6:4)**

[29] Chin-Yu Huang, Jun-Ru Chang, and Yung-Hsin Chang. 2010. Design and Analysis of GUI Test-case Prioritization using Weight-based Methods. *J. Syst. Softw.* 83, 4 (April 2010), 646–659. **(C1:4, C2:2, C3:1, C4:7, C5:1, C6:1)**

[30] Renee C. Bryce, Sreedevi Sampath, and Atif M. Memon. 2011. Developing a Single Model and Test Prioritization Strategies for Event-Driven Software. *IEEE Trans. Softw. Eng.* 37, 1 (January 2011), 48–64. **(C1:4, C2:2, C3:2, C4:1, C5:1, C6:1)**

[31] Mark Last, Shay Eyal, Abraham Kandel. 2005. Effective Black-Box Testing with Genetic Algorithms. *In Proceedings of Haifa Verification Conference '2005.* pp. 134 ~ 148. **(C1:4, C2:1, C3:2, C4:1, C5:1, C6:9)**

[32] Sara Alspaugh, Kristen R. Walcott, Michael Belanich, Gregory M. Kapfhammer, and Mary Lou Soffa. 2007. Efficient Time-aware Prioritization with Knapsack Solvers. *In Proceedings of the 1st ACM International Workshop on Empirical Assessment of Software Engineering Languages and Technologies: held in conjunction with the 22nd IEEE/ACM International Conference on Automated Software Engineering* (*ASE*) *2007* (*WEASELTech '07*). ACM, New York, NY, USA, 13–18. **(C1:3, C2:2, C3:1, C4:1&9, C5:2, C6:9)**

[33] Hyunsook Do, Gregg Rothermel, and Alex Kinneer. 2004. Empirical Studies of Test Case Prioritization in a JUnit Testing Environment. *In Proceedings of the 15th International Symposium on Software Reliability Engineering* (ISSRE '04). IEEE Computer Society, Washington, DC, USA, 113–124. **(C1:3, C2:2, C3:1, C4:1, C5:2, C6:1)**

[34] Alexander P. Conrad, Robert S. Roos, and Gregory M. Kapfhammer. 2010. Empirically Studying the Role of Selection Operators during Search-Based Test Suite Prioritization. *In the Proceedings of the ACM SIGEVO Genetic and Evolutionary Computation Conference*, Portland, Oregon, July 2010. **(C1:4, C2:2, C3:1, C4:1, C5:1, C6:7)**

[35] T. M. S. Ummu Salima, A. Askarunisha, and N. Ramaraj. 2007. Enhancing the Efficiency of Regression Testing through Intelligent Agents. *In Proceedings of the International Conference on Computational Intelligence and Multimedia*

Applications (*ICCIMA 2007*)—*Volume 01* (ICCIMA ‘07), Vol. 1. IEEE Computer Society, Washington, DC, USA, 103–108. **(C1:1, C2:3, C3:4, C4:1, C5:4, C6:1)**

[36] Bogdan Korel and George Koutsogiannakis. 2009. Experimental Comparison of Code-Based and Model-Based Test Prioritization. *In Proceedings of the IEEE International Conference on Software Testing, Verification, and Validation Workshops* (ICSTW ‘09). IEEE Computer Society, Washington, DC, USA, 77–84. **(C1:3, C2:2, C3:2, C4:1, C5:2, C6:6)**

[37] Dennis Jeffrey and Neelam Gupta. 2008. Experiments with Test Case Prioritization using Relevant Slices. *J. Syst. Softw.* 81, 2 (February 2008), 196–221. **(C1:4, C2:2, C3:1, C4:1&9, C5:1, C6:1)**

[38] R. Krishnamoorthi and S. A. Sahaaya Arul Mary. 2009. Factor Oriented Requirement Coverage based System Test Case Prioritization of New and Regression Test Cases. *Inf. Softw. Technol.* 51, 4 (April 2009), 799–808. **(C1:4, C2:1, C3:2, C4:3&6, C5:1, C6:2)**

[39] Y. T. Yu, S. P. Ng, and Eric Y. K. Chan. 2003. Generating, Selecting and Prioritizing Test Cases from Specifications with Tool Support. *In Proceedings of the Third International Conference on Quality Software* (QSIC ‘03). IEEE Computer Society, Washington, DC, USA, 83–. **(C1:1, C2:3, C3:4, C4:9, C5:4, C6:8)**

[40] Hyuncheol Park, Hoyeon Ryu, and Jongmoon Baik. 2008. Historical Value-Based Approach for Cost-Cognizant Test Case Prioritization to Improve the Effectiveness of Regression Testing. In *Proceedings of the 2008 Second International Conference on Secure System Integration and Reliability Improvement* (SSIRI ‘08). IEEE Computer Society, Washington, DC, USA, 39–46. **(C1:4, C2:2, C3:1, C4:8, C5:1, C6:4)**

[41] Bo Jiang, Zhenyu Zhang, T. H. Tse, and T. Y. Chen. 2009. How Well Do Test Case Prioritization Techniques Support Statistical Fault Localization. *In Proceedings of the 2009 33rd Annual IEEE International Computer Software and Applications Conference—Volume 01* (COMPSAC ‘09), Vol. 1. IEEE Computer Society, Washington, DC, USA, 99–106. **(C1:3, C2:2, C3:1, C4:1&2, C5:4, C6:9)**

[42] Yalda Fazlalizadeh, Alireza Khalilian, Mohammad Abdollahi Azgomi, and Saeed Parsa. 2009. Incorporating Historical Test Case Performance Data and Resource Constraints into Test Case Prioritization. *In Proceedings of the 3rd International Conference on Tests and Proofs* (TAP ‘09), Catherine Dubois (Ed.). Springer-Verlag, Berlin, Heidelberg, 43–57. **(C1:4, C2:2, C3:1, C4:5, C5:1, C6:1)**

[43] Ramasamy Krishnamoorthi, S.A. Sahaaya Arul Mary. 2008. Incorporating Varying Requirement Priorities and Costs in Test Case Prioritization for New and Regression Testing. In *Proceedings of the* Computing, Communication and Networking, 2008. ICCCn 2008. pp. 1–9. **(C1:4, C2:1, C3:2, C4:3&6, C5:1, C6:2)**

[44] Sebastian Elbaum, Alexey Malishevsky, and Gregg Rothermel. 2001. Incorporating Varying Test Costs and Fault Severities into Test Case Prioritization. *In Proceedings of the 23rd International Conference on Software Engineering* (ICSE ‘01). IEEE Computer Society, Washington, DC, USA, 329–338. **(C1:3, C2:2, C3:1, C4:1, C5:3, C6:4)**

[45] Lingming Zhang, Ji Zhou, Dan Hao, Lu Zhang, and Hong Mei. 2009. Jtop: Managing JUnit Test Cases in Absence of Coverage Information. *In Proceedings of the 2009 IEEE/ACM International Conference on Automated Software Engineering* (ASE ‘09). IEEE Computer Society, Washington, DC, USA, 677–679. **(C1:5, C2:3, C3:4, C4:7, C5:4, C6:8)**

[46] Praveen Ranjan Srivastava. 2008. Model for Optimizing Software Testing Period using Non homogenous Poisson Process based on Cumulative Test Case Prioritization. *In Proceedings of the TENCON 2008–2008 IEEE Region 10 Conference*, pp. 1–6, 19–21 Nov. 2008. **(C1:4, C2:3, C3:4, C4:4, C5:1, C6:8)**

[47] Bogdan Korel, George Koutsogiannakis, and Luay H. Tahat. 2007. Model-based Test Prioritization Heuristic Methods and Their Evaluation. *In Proceedings of the 3rd International Workshop on Advances in Model-based Testing* (A-MOST '07). ACM, New York, NY, USA, 34–43. **(C1:4, C2:2, C3:2, C4:7, C5:1, C6:6)**

[48] Fevzi Belli, Mubariz Eminov, and Nida Gokce. 2009. Model-based Test Prioritizing: A Comparative Soft-computing Approach and Case Studies. *In Proceedings of the 32nd Annual German Conference on Advances in Artificial Intelligence* (KI'09), Springer-Verlag, Berlin, Heidelberg, 427–434. **(C1:4, C2:2, C3:2, C4:7&1, C5:1, C6:8)**

[49] Alexey G. Malishevsky, Gregg Rothermel, Sebastian Elbaum. 2002. Modeling the Cost-Benefits Tradeoffs for Regression Testing Techniques. *In Proceedings of the International Conference on Software Maintenance* (*ICSM'02*) (ICSM '02). IEEE Computer Society, Washington, DC, USA, 204-. **(C1:3, C2:2, C3:2, C4:1, C5:4, C6:9)**

[50] Gregg Rothermel, Sebastian Elbaum, Alexey G. Malishevsky, Praveen Kallakuri, and Xuemei Qiu. 2004. On Test Suite Composition and Cost-effective Regression Testing. *ACM Trans. Softw. Eng. Methodol.* 13, 3 (July 2004), 277–331. **(C1:3, C2:2, C3:2, C4:1, C5:4, C6:1)**

[51] Hema Srikanth and Laurie Williams. 2005. On the Economics of Requirements-based Test Case Prioritization. In *Proceedings of the Seventh International Workshop on Economics-driven Software Engineering Research* (*EDSER '05*). ACM, New York, NY, USA, 1–3. **(C1:4, C2:1, C3:2, C4:6&3, C5:1, C6:2)**

[52] Bo Jiang and W. K. Chan. 2010. On the Integration of Test Adequacy, Test Case Prioritization, and Statistical Fault Localization. *In Proceedings of the 2010 10th International Conference on Quality Software* (QSIC '10). IEEE Computer Society, Washington, DC, USA, 377–384. **(C1:3, C2:2, C3:1, C4:1&9, C5:4, C6:9)**

[53] Hyunsook Do and Gregg Rothermel. 2006. On the Use of Mutation Faults in Empirical Assessments of Test Case Prioritization Techniques. *IEEE Trans. Softw. Eng.* 32, 9 (September 2006), 733–752. **(C1:3, C2:2, C3:1, C4:1, C5:2, C6:1)**

[54] Murali Krishna Ramanathan, Mehmet Koyuturk, Ananth Grama, and Suresh Jagannathan. 2008. PHALANX: A Graph-theoretic Framework for Test Case Prioritization. In *Proceedings of the 2008 ACM Symposium on Applied Computing* (*SAC '08*). ACM, New York, NY, USA, 667–673. **(C1:3, C2:2, C3:1, C4:1&9, C5:2, C6:1)**

[55] Ke Zhai and W. K. Chan. 2010. Point-of-Interest Aware Test Case Prioritization: Methods and Experiments. *In Proceedings of the 2010 10th International Conference on Quality Software* (QSIC '10). IEEE Computer Society, Washington, DC, USA, 449–456. **(C1:3, C2:1, C3:2, C4:9, C5:2, C6:1)**

[56] Mark Sherriff, Mike Lake, and Laurie Williams. 2007. Prioritization of Regression Tests using Singular Value Decomposition with Empirical Change Records. *In Proceedings of the The 18th IEEE International Symposium on Software Reliability* (ISSRE '07). IEEE Computer Society, Washington, DC, USA, 81–90. **(C1:4, C2:2, C3:2, C4:5&9, C5:1, C6:8)**

[57] Sapna P.G., Hrushikesha Mohanty. 2009. Prioritization of Scenarios Based on UML Activity Diagrams. *In Proceedings of the First International Conference on*

Computational Intelligence, Communication Systems and Networks (*cicsyn 2009*), pp. 271–276. **(C1:4, C2:1, C3:4, C4:7, C5:1, C6:8)**
[58] Sangeeta Sabharwal, Ritu Sibal, Chayanika Sharma. 2010. Prioritization of Test Case Scenarios Derived from Activity Diagram using Genetic Algorithm. *In Proceedings of the Computer and Communication Technology* (*ICCCT*), *2010 International Conference on,* pp. 481–485. **(C1:4, C2:1, C3:4, C4:1, C5:1, C6:8)**
[59] Yuen-Tak Yu and Man Fai Lau. 2002. Prioritization of Test Cases in MUMCUT Test Sets: An Empirical Study. *In Proceedings of the 7th Ada-Europe International Conference on Reliable Software Technologies* (*Ada-Europe '02*), Springer-Verlag, London, UK, UK, 245–256. **(C1:3, C2:2, C3:1, C4:9, C5:2, C6:1)**
[60] Renee C. Bryce, Charles J. Colbourn. 2006. Prioritized Interaction Testing for Pair-wise Coverage with Seeding and Constraints, *Information and Software Technology*, Volume 48, Issue 10, Advances in Model-based Testing, October 2006, Pages 960–970. **(C1:4, C2:3, C3:4, C4:1, C5:1, C6:8)**
[61] Jianhua Gao and Jie Zhu. 2009. Prioritized Test Generation Strategy for Pair-Wise Testing. *In Proceedings of the 2009 15th IEEE Pacific Rim International Symposium on Dependable Computing* (PRDC '09). IEEE Computer Society, Washington, DC, USA, 99–102. **(C1:4, C2:3, C3:4, C4:1, C5:1, C6:8)**
[62] Fevzi Belli, Mubariz Eminov, and Nida Gokce. 2007. Prioritizing Coverage-Oriented Testing Process—An Adaptive-Learning-Based Approach and Case Study. *In Proceedings of the 31st Annual International Computer Software and Applications Conference–Volume 02* (COMPSAC '07), Vol. 2. IEEE Computer Society, Washington, DC, USA, 197–203. **(C1:4, C2:2, C3:2, C4:1, C5:1, C6:8)**
[63] Lingming Zhang, Ji Zhou, Dan Hao, Lu Zhang and Hong Mei. 2009. Prioritizing JUnit Test Cases in Absence of Coverage Information. *In Proceedings of 25th International Conference on Software Maintenance* (*ICSM 2009*), 19–28. **(C1:4, C2:2, C3:1, C4:7, C5:1, C6:1)**
[64] Hyunsook Do, Gregg Rothermel, Alex Kinneer. 2006. Prioritizing JUnit Test Cases: An Empirical Assessment and Cost-Benefits Analysis. *Empirical Software Engineering.* 11, 1, 33–70. **(C1:3, C2:2, C3:1, C4:1, C5:1, C6:1)**
[65] Dianxiang Xu and Junhua Ding. 2010. Prioritizing State-Based Aspect Tests. *In Proceedings of the 2010 Third International Conference on Software Testing, Verification and Validation* (ICST '10). IEEE Computer Society, Washington, DC, USA, 265–274. **(C1:4, C2:2, C3:2, C4:7, C5:1, C6:9)**
[66] Gregg Rothermel, Roland J. Untch, and Chengyun Chu. 2001. Prioritizing Test Cases For Regression Testing. *IEEE Trans. Softw. Eng.* 27, 10 (October 2001), 929–948. **(C1:3, C2:2, C3:1, C4:1, C5:1, C6:1)**
[67] Y. Fazlalizadeh, A. Khalilian, M. Azgomi, and S. Parsa. 2009. Prioritizing Test Cases for Resource Constraint Environments using Historical Test Case Performance Data. *2nd IEEE International Conference on Computer Science and Information Technology*, 190–195. **(C1:4, C2:2, C3:1, C4:5, C5:1, C6:1)**
[68] Alberto Gonzalez-Sanchez, Eric Piel, Hans-Gerhard Gross, and Arjan J. C. van Gemund. 2010. Prioritizing Tests for Software Fault Localization. *In Proceedings of the 2010 10th International Conference on Quality Software* (QSIC '10). IEEE Computer Society, Washington, DC, USA, 42–51. **(C1:4, C2:2, C3:1, C4:9, C5:1, C6:1)**
[69] Emad Shihab, Zhen Ming Jiang, Bram Adams, Ahmed E. Hassan, and Robert Bowerman. 2010. Prioritizing Unit Test Creation for Test-Driven Maintenance of Legacy Systems. *In Proceedings of the 2010 10th International Conference on*

Quality Software (QSIC '10). IEEE Computer Society, Washington, DC, USA, 132–141. **(C1:4, C2:2, C3:2, C4:5, C5:1&3, C6:9)**

[70] Sapna P.G. and Hrusikesha Mohanty. 2009. Prioritizing Use Cases to Aid Ordering of Scenarios. *In Proceedings of the 2009 Third UKSim European Symposium on Computer Modeling and Simulation* (EMS '09). IEEE Computer Society, Washington, DC, USA, 136–141. **(C1:4, C2:1, C3:2, C4:3&7, C5:1, C6:1)**

[71] Sreedevi Sampath, Renee C. Bryce, Gokulanand Viswanath, Vani Kandimalla, and A. Gunes Koru. 2008. Prioritizing User-Session-Based Test Cases for Web Applications Testing. *In Proceedings of the 2008 International Conference on Software Testing, Verification, and Validation*. IEEE Computer Society, Washington, DC, USA, 141–150. **(C1:4, C2:2, C3:2, C4:1, C5:1, C6:1)**

[72] Gregg Rothermel and Sebastian Elbaum. 2003. Putting Your Best Tests Forward. *IEEE Softw.* 20, 5 (September 2003), 74–77. **(C1:5, C2:2, C3:2, C4:1, C5:1, C6:1)**

[73] Shan–Shan Hou, Lu Zhang, Tao Xie, Jiasu Sun. 2008. Quota-Constrained Test-Case Prioritization for Regression Testing of Service-Centric Systems. *In Proceedings of International Conference on Software Maintenance 2008*, 257–266. **(C1:4, C2:1, C3:2, C4:9, C5:1, C6:1)**

[74] Hema Srikanth, Myra B. Cohen, and Xiao Qu. 2009. Reducing Field Failures in System Configurable Software: Cost-based Prioritization. *In Proceedings of the 20th IEEE International Conference on Software Reliability Engineering* (ISSRE'09), IEEE Press, Piscataway, NJ, USA, 61–70. **(C1:4, C2:2, C3:2, C4:8, C5:1&3, C6:5)**

[75] Cristian Simons, Emerson Cabrera Paraiso. Regression Test Cases Prioritization using Failure Pursuit Sampling. *In Proceedings of the ISDA 2010*, 923–928. **(C1:4, C2:2, C3:1, C4:2, C5:1, C6:1)**

[76] Zhihao Zhang, Jun Huang, Bo Zhang, Jinlong Lin, and Xiaolan Chen. 2010. Regression Test Generation Approach Based on Tree-Structured Analysis. *In Proceedings of the 2010 International Conference on Computational Science and Its Applications* (ICCSA '10). IEEE Computer Society, Washington, DC, USA, 244–249. **(C1:1, C2:3, C3:4, C4:5, C5:1, C6:8)**

[77] Nashat Mansour and Wael Statieh. 2009. Regression test selection for c\# programs. *Adv. Soft. Eng.* 2009, Article 1 (January 2009), 16 pages. **(C1:1, C2:2, C3:2, C4:5, C5:1, C6:8)**

[78] Pavan Kumar Chittimalli and Mary Jean Harrold. 2008. Regression Test Selection on System Requirements. *In Proceedings of the 1st India Software Engineering Conference* (*ISEC '08*). ACM, New York, NY, USA, 87–96. **(C1:4, C2:2, C3:2, C4:6, C5:1, C6:8)**

[79] S. Yoo, M. Harman. 2012. Regression Testing Minimization, Selection and Prioritization: a Survey. *Software Testing, Verification and Reliability* 22, 2, 67–120. **(C1:6, C2:3, C3:4, C4:10, C5:1, C6:8)**

[80] Alireza Mahdian, Anneliese Amschler Andrews, and Orest Jacob Pilskalns. 2009. Regression Testing with UML Software Designs: A Survey. *J. Softw. Maint. Evol.* 21, 4 (July 2009), 253–286. **(C1:6, C2:3, C3:4, C4:10, C5:1, C6:8)**

[81] R. Kavitha, V.R. Kavitha, N. Suresh Kumar. 2010. Requirement Based Test Case Prioritization. *In Proceedings of International Conference on Communication Control and Computing Technologies*, 826–829. **(C1:4, C2:2, C3:2, C4:6, C5:1&3, C6:3)**

[82] Zheng Li, Mark Harman, and Robert M. Hierons. 2007. Search Algorithms for Regression Test Case Prioritization. *IEEE Trans. Softw. Eng.* 33, 4 (April 2007), 225–237. **(C1:3, C2:2, C3:1, C4:10, C5:1, C6:9)**

[83] Sebastian Elbaum, Gregg Rothermel, Satya Kanduri, and Alexey G. Malishevsky. 2004. Selecting a Cost-Effective Test Case Prioritization Technique. *Software Quality Journal* 12, 3 (September 2004), 185–210. **(C1:3, C2:2, C3:3, C4:1, C5:4, C6:1)**

[84] Roberto S. Silva Filho, Christof J. Budnik, William M. Hasling, Monica McKenna, Rajesh Subramanyan. 2010. Supporting Concern-Based Regression Testing and Prioritization in a Model-Driven Environment. *In Proceedings of IEEE 34th Annual Computer Software and Applications Conference Workshops*, 323–328. **(C1:1, C2:3, C3:4, C4:7, C5:1, C6:8)**

[85] Hema Srikanth, Laurie Williams, Jason Osborne. 2005. System Test Case Prioritization of New and Regression Test Cases. *In Proceedings of ISESE'2005*, 64–73. **(C1:4, C2:1, C3:2, C4:14&6, C5:6&3, C6:9)**

[86] Debasish Kundu, Monalisa Sarma, Debasis Samanta, and Rajib Mall. 2009. System Testing for Object-Oriented Systems with Test Case Prioritization. *Softw. Test. Verif. Reliab.* 19, 4 (December 2009), 297–333. **(C1:4, C2:2, C3:2, C4:7, C5:1, C6:9)**

[87] Lijun Mei, W. K. Chan, T. H. Tse, and Robert G. Merkel. 2009. Tag-Based Techniques for Black-Box Test Case Prioritization for Service Testing. *In Proceedings of the 2009 Ninth International Conference on Quality Software* (QSIC '09). IEEE Computer Society, Washington, DC, USA, 21–30. **(C1:4, C2:2, C3:2, C4:9, C5:1, C6:1)**

[88] Ke Zhai, Bo Jiang, W. K. Chan, T. H. Tse. 2010. Taking Advantage of Service Selection: A Study on the Testing of Location-Based Web Services through Test Case Prioritization. *In Proceedings of the ICWS 2010,* 211–218. **(C1:4, C2:2, C3:2, C4:9, C5:1&3, C6:9)**

[89] P. R. Srivastava, M. Ray, and J. Dermoudy, and B. H. Kang, and T. H. Kim. 2009. *Test Case Minimization and Prioritization Using CMIMX Technique*. *In Proceedings of the International Conference on Advanced Software Engineering and Its Applications*. Jeju Island, Korea, 25–33. **(C1:4, C2:3, C3:4, C4:1, C5:1, C6:8)**

[90] Zengkai Ma and Jianjun Zhao. 2008. Test Case Prioritization Based on Analysis of Program Structure. *In Proceedings of the 2008 15th Asia–Pacific Software Engineering Conference* (APSEC '08). IEEE Computer Society, Washington, DC, USA, 471–478. **(C1:4, C2:2, C3:1, C4:5, C5:1&3, C6:4)**

[91] Lucas Lima, Juliano Iyoda, Augusto Sampaio, and Eduardo Aranha. 2009. Test Case Prioritization based on Data Reuse an Experimental Study. *In Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement* (ESEM '09). IEEE Computer Society, Washington, DC, USA, 279–290. **(C1:3, C2:2, C3:2, C4:9, C5:1, C6:9)**

[92] Xiaofang Zhang, Changhai Nie, Baowen Xu, and Bo Qu. 2007. Test Case Prioritization Based on Varying Testing Requirement Priorities and Test Case Costs. *In Proceedings of the Seventh International Conference on Quality Software* (QSIC '07). IEEE Computer Society, Washington, DC, USA, 15–24. **(C1:4, C2:3, C3:4, C4:6&8, C5:1&3, C6:9)**

[93] Bo Qu, Changhai Nie, Baowen Xu, and Xiaofang Zhang. 2007. Test Case Prioritization for Black Box Testing. *In Proceedings of the 31st Annual International Computer Software and Applications Conference—Volume 01* (COMPSAC '07), Vol. 1. IEEE Computer Society, Washington, DC, USA, 465–474. **(C1:4, C2:1, C3:2, C4:5, C5:1&3, C6:9)**

[94] Bo Qu, Changhai Nie, Baowen Xu. 2008. Test Case Prioritization for Multiple Processing Queues. *In Proceedings of International Symposium on Information Science and Engieering*, 646–649. **(C1:4, C2:1, C3:2, C4:9, C5:1&3, C6:1)**

[95] Lijun Mei, Zhenyu Zhang, W. K. Chan, and T. H. Tse. 2009. Test Case Prioritization for Regression Testing of Service-Oriented Business Applications. *In Proceedings of the 18th International Conference on World Wide Web* (WWW '09). ACM, New York, NY, USA, 901–910. **(C1:4, C2:2, C3:2, C4:1, C5:1, C6:1)**

[96] Lin Chen, Ziyuan Wang, Lei Xu, Hongmin Lu, and Baowen Xu. 2010. Test Case Prioritization for Web Service Regression Testing. *In Proceedings of the 2010 Fifth IEEE International Symposium on Service Oriented System Engineering* (SOSE '10). IEEE Computer Society, Washington, DC, USA, 173–178. **(C1:4, C2:3, C3:4, C4:7, C5:1, C6:8)**

[97] Dennis Jeffrey and Neelam Gupta. 2006. Test Case Prioritization Using Relevant Slices. In *Proceedings of the 30th Annual International Computer Software and Applications Conference—Volume 01* (COMPSAC '06), Vol. 1. IEEE Computer Society, Washington, DC, USA, 411–420. **(C1:4, C2:2, C3:1, C4:1, C5:1, C6:1)**

[98] Sebastian Elbaum, Alexey G. Malishevsky, Gregg Rothermel. 2002. Test Case Prioritization: A Family of Empirical Studies. *IEEE Transactions on Software Engineering*. 28, 2, 159–182. **(C1:3, C2:2, C3:1, C4:1, C5:1, C6:1)**

[99] Jussi Kasurinen, Ossi Taipale, and Kari Smolander. 2010. Test Case Selection and Prioritization: Risk-based or Design-based? *In Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement* (ESEM '10). ACM, New York, NY, USA, Article 10, 10 pages. **(C1:2, C2:3, C3:4, C4:10, C5:1&3, C6:8)**

[100] Farn Wang and Geng-Dian Huang. 2008. Test Plan Generation for Concurrent Real-Time Systems Based on Zone Coverage Analysis. *In Proceedings of the 20th IFIP TC 6/WG 6.1 International Conference on Testing of Software and Communicating Systems: 8th International Workshop* (TestCom '08/FATES '08). Springer-Verlag, Berlin, Heidelberg, 234–249. **(C1:3, C2:2, C3:2, C4:1, C5:4, C6:9)**

[101] Bogdan Korel, Luay H. Tahat, and Mark Harman. 2005. Test Prioritization Using System Models. *In Proceedings of the 21st IEEE International Conference on Software Maintenance* (*ICSM '05*). IEEE Computer Society, Washington, DC, USA, 559–568. **(C1:4, C2:2, C3:2, C4:7, C5:1&3, C6:6)**

[102] Renee C. Bryce and Atif M. Memon. 2007. Test Suite Prioritization by Interaction Coverage. *In Workshop on Domain Specific Approaches to Software Test Automation: in conjunction with the 6th ESEC/FSE Joint Meeting* (*DOSTA '07*). ACM, New York, NY, USA, 1–7. **(C1:4, C2:2, C3:2, C4:9, C5:1, C6:1)**

[103] Adam M. Smith, Joshua Geiger, Gregory M. Kapfhammer, and Mary Lou Soffa. 2007. Test Suite Reduction and Prioritization with Call Trees. *In Proceedings of the Twenty-Second IEEE/ACM International Conference on Automated Software Engineering* (ASE '07). ACM, New York, NY, USA, 539–540. **(C1:3, C2:1, C3:2, C4:1, C5:1, C6:7)**

[104] Marek G. Stochel and Radek Sztando. 2008. Testing Optimization for Mission-Critical, Complex, Distributed Systems. *In Proceedings of the 2008 32nd Annual IEEE International Computer Software and Applications Conference* (COMPSAC '08). IEEE Computer Society, Washington, DC, USA, 847–852. **(C1:4, C2:2, C3:2, C4:5, C5:1, C6:9)**

[105] James A. Jones and Mary Jean Harrold. 2001. Test-Suite Reduction and Prioritization for Modified Condition/Decision Coverage. *In Proceedings of the IEEE International Conference on Software Maintenance* (*ICSM'01*) (ICSM '01). IEEE Computer Society, Washington, DC, USA, 92–101. **(C1:4, C2:2, C3:1, C4:1, C5:1, C6:8)**

[106] James A. Jones and Mary Jean Harrold. 2003. Test-Suite Reduction and Prioritization for Modified Condition/Decision Coverage. *IEEE Trans. Softw. Eng.* 29, 3 (March 2003), 195–209. **(C1:4, C2:2, C3:1, C4:1, C5:1, C6:8)**

[107] Panduka Nagahawatte and Hyunsook Do. 2010. The Effectiveness of Regression Testing Techniques in Reducing the Occurrence of Residual Defects. *In Proceedings of the 2010 Third International Conference on Software Testing, Verification and Validation* (ICST '10). IEEE Computer Society, Washington, DC, USA, 79–88. **(C1:3, C2:2, C3:1, C4:1, C5:4, C6:9)**

[108] Hyunsook Do, Siavash Mirarab, Ladan Tahvildari, and Gregg Rothermel. 2010. The Effects of Time Constraints on Test Case Prioritization: A Series of Controlled Experiments. *IEEE Trans. Softw. Eng.* 36, 5 (September 2010), 593–617. **(C1:3, C2:2, C3:1, C4:1&4, C5:4, C6:9)**

[109] Saif ur Rehman Khan, Inayat ur Rehman, Saif Ur Rehman Malik. 2009. The Impact of Test Case Reduction and Prioritization on Software Testing Effectiveness. *In Proceedings of 5th International Conference on Emerging Technologies.* Islamabad, Pakistan, 416–421. **(C1:3, C2:1, C3:2, C4:1, C5:4, C6:8)**

[110] K. Walcott, M. Soffa, G. Kapfhammer, and R. Roos. 2006. Time-Aware Test Suite Prioritization. *In Proceedings of ISSTA* '06. Portland, Maine, USA, 1–11. **(C1:4, C2:2, C3:3, C4:1, C5:1, C6:1)**

[111] Lu Zhang, Shan–Shan Hou, Chao Guo, Tao Xie, and Hong Mei. 2009. Time-aware Test-case Prioritization using Integer Linear Programming. *In Proceedings of the eighteenth international symposium on Software testing and analysis* (ISSTA '09). ACM, New York, NY, USA, 213–224. **(C1:4, C2:2, C3:1, C4:9, C5:1, C6:9)**

[112] Matthew J. Rummel, Gregory M. Kapfhammer, and Andrew Thall. 2005. Towards the Prioritization of Regression Test Suites with Data Flow Information. *In Proceedings of the 2005 ACM Symposium on Applied Computing* (SAC '05), ACM, New York, NY, USA, 1499–1504. **(C1:4, C2:1, C3:2, C4:7, C5:1, C6:1)**

[113] Sebastian Elbaum, David Gable, and Gregg Rothermel. 2001. Understanding and Measuring the Sources of Variation in the Prioritization of Regression Test Suites. *In Proceedings of the 7th International Symposium on Software Metrics* (METRICS '01). IEEE Computer Society, Washington, DC, USA, 169-. **(C1:3, C2:2, C3:1, C4:1, C5:4, C6:1)**

[114] Sebastian Elbaum, Praveen Kallakuri, Alexey Malishevsky, Gregg Rothermel, Satya Kanduri. 2003. Understanding the Effects of Changes on the Cost-effectiveness of Regression Testing Techniques. *J. Softw. Test. Verif. Reliab.* 13, 2, 65–81. **(C1:3, C2:2, C3:3, C4:1, C5:4, C6:1)**

[115] Gregory M. Kapfhammer and Mary Lou Soffa. 2007. Using Coverage Effectiveness to Evaluate Test Suite Prioritizations. In *Proceedings of the 1st ACM International Workshop on Empirical Assessment of Software Engineering Languages and Technologies: held in conjunction with the 22nd IEEE/ACM International Conference on Automated Software Engineering* (*ASE*) *2007* (WEASELTech '07). ACM, New York, NY, USA, 19–20. **(C1:1, C2:3, C3:4, C4:10, C5:3, C6:7)**

[116] Zhi Quan Zhou. 2010. Using Coverage Information to Guide Test Case Selection in Adaptive Random Testing. *In Proceedings of 34th Annual Computer Software and Applications Conference Workshops*, 208–213. **(C1:4, C2:2, C3:1, C4:1, C5:1&3, C6:9)**

[117] Zachary D. Williams and Gregory M. Kapfhammer. 2010. Using Synthetic Test Suites to Empirically Compare Search-based and Greedy Prioritizers. *In Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation* (GECCO '10). ACM, New York, NY, USA, 2119–2120. **(C1:3, C2:3, C3:4, C4:10, C5:4, C6:7)**

[118] Paolo Tonella, Paolo Avesani, and Angelo Susi. 2006. Using the Case-Based Ranking Methodology for Test Case Prioritization. *In Proceedings of the 22nd IEEE International Conference on Software Maintenance* (ICSM '06). IEEE Computer Society, Washington, DC, USA, 123–133. **(C1:4, C2:2, C3:1, C4:9, C5:1, C6:1)**

[119] Athira B and Philip Samuel. 2010. Web Services Regression Test Case Prioritization. *In Proceedings of CISIM 2010.* **(C1:4, C2:2, C3:4, C4:7, C5:1, C6:8)**

[120] Lijun Mei, W. K. Chan, T. H. Tse, and Robert G. Merkel. 2011. XML-manipulating Test Case Prioritization for XML-manipulating Services. *J. Syst. Softw*. 84, 4 (April 2011), 603–619. **(C1:4, C2:2, C3:2, C4:9, C5:1, C6:1)**

Appendix 2

The Numbers of Papers in Each Journal

(The numbers of papers identified in each journal are shown in ().)

ACM Transactions on Software Engineering and Methodology (1)
Advances in Software Engineering (1)
Empirical Software Engineering (1)
IEEE Software (1)
IEEE Transactions on Software Engineering (7)—**Rank 1**
Information and Software Technology (2)—**Rank 3**
Journal of Maintenance and Evolution: Research and Practice (1)
Journal of Systems and Software (3)—**Rank 2**
Software Quality Journal (1)
Software Testing, Verification and Reliability (3)—**Rank 2**

Appendix 3

Category 1 and Category 2 were obtained from the review paper of Jorgensen and Shepperd (2007). Category 3 was obtained from our review article Catal and Diri (2009). We created the major categories based on the number of studies that used a particular category. For example, if there were just one or two papers about a subcategory, we did not create a new major subcategory for that item. We made multiple passes over the papers to iteratively extract potential properties and categories and, ultimately, we came up with these categories.

Category 1: Research Approach

1. Theory: Non-empirical research approaches.
2. Survey: Questionnaire and interview-based surveys of industry practice.
3. Experiment: Experiment-based studies.
4. Development of prioritization method: Studies where new prioritization models are developed.
5. Personal experience/lessons learned: Studies where the reference is one's own personal experience.
6. Review: Studies that review other papers.

Category 2: Study Context

1. Datasets from student projects: Studies where the subjects are students or student projects. For instance, academic toy examples belong to this category.
2. Datasets from industrial projects: Studies where the subjects are software professionals and/or industrial software projects. They are developed by professional programmers.
3. Not relevant: Studies where the study context is not available or not relevant.

Category 3: Dataset Type

1. Public: Public datasets are stored in public repositories such as a SIR (Software-artifacts Infrastructure Repository). These datasets are publicly available.
2. Private: Private datasets mostly belong to software companies and are not freely distributed as public datasets.
3. Partial: Partial datasets are datasets that have been created using data from open source projects and have not been distributed to the community.
4. None: If there is no information about the dataset in the paper, it is called "None."

Category 4: prioritization approach Yoo and Harman's (2012) classification

1. Coverage-based: These techniques order test cases based on the coverage of code components. There are different coverage-based techniques. For example, one of them prioritizes test cases in order of coverage of branches, or it prioritizes test cases in order of coverage of statements not yet covered.
2. Distribution-based: These techniques use a multidimensional distribution of test cases based on their multivariable descriptions.
3. Human-based: Prioritization is based on the comparisons made by the human tester.
4. Probabilistic Approach: Probabilistic theory is applied.
5. History-based: Historical information such as faults or execution history is used in this type of technique.
6. Requirements-based: Requirements are taken into account in this type of technique.
7. Model-based: Instead of using code blocks, model-based prioritization techniques utilize different models such as sequence diagram or state charts.
8. Cost-aware approach: The cost of each test case is not equal, and therefore, cost-based techniques provide effective solutions when the cost factor is significant.

9. Other Approaches: The other types of prioritization methods are located in this category.
10. None: Prioritization approach is not used.

Category 5: Research Topic

1. Prioritization method: A new test case prioritization approach is proposed and evaluated in this type of paper.
2. Comparison of prioritization methods: Several prioritization methods are compared in this type of paper.
3. Measures of prioritization performance: A new evaluation metric for TCP is proposed and evaluated in this type of paper.
4. Others: Change effect identification, combination of prioritization and reduction methods, tool development, use of prioritization method for test case selection, use of prioritization method for software fault localization, time constraint effects, selecting the most effective prioritization method, test suite generation, test plan generation, survey on prioritization methods, review of prioritization methods, residual defects identification, test adequacy criterion, measuring the sources of variation in the prioritization, and effects of test suite granularity.

Category 6: Evaluation Metric

1. APFD (Average Percentage of Faults Detected): "APFD measures the weighted average of the percentage of faults detected over the life of a test suite" Kapfhammer and Soffa (2007). APFD values are between 0 and 100, with higher values indicating better fault detection.
2. ASFD (Average Severity of Faults Detected): A severity value is assigned to each fault. Total Severity of Faults Detected (TSFD) is the sum of severity values of the faults identified. "The ASFDi for requirement i is the ratio of the sum of severity faults identified for the requirement I, to the TSFD" Srikanth and Williams (2005).
3. TPFD (Total Percentage of Faults Detected): "TPFD metric is the area under the curve when plotting a graph with the fraction of requirement on X-axis, and percentage of TSFD on Y-axis" Krishnamoorthi et al. (2009).
4. APFD(c) (Average Percentage of Faults Detected per Cost): APFD metric assumes that faults have equal severity and test cases have equal costs Ma and Zhao (2008). However, this is not possible in practice and therefore, an APFD(c) metric that takes into account the fault severity and test cost was proposed.
5. NAPFD (Normalized APFD): This metric measures test case prioritization and configuration prioritization. It includes information on fault finding and the time of detection Qu et al. (2007).
6. RP (Most Likely Relative Position): "It represents an average relative position of the first failed test that detects d for a test case prioritization method" Korel and Koutsogiannakis (2009).
7. CE (Coverage Effectiveness): This metric incorporates the cost and the coverage of each test case. CE value is between 0 and 1 with a higher value indicating a better test case. It is the ratio between the reordered suite's coverage area and the coverage area of the ideal test suite that covers all requirements Kapfhammer and Soffa (2007).

8. None: Evaluation metric was not used.
9. Others: The other types of evaluation metrics were used.

References

Catal, C., & Diri, B. (2009). A systematic review of software fault prediction studies. *Expert System with Applications, 36*(4), 7346–7354.

Do, H., Elbaum, S., & Rothermel, G. (2005). Supporting controlled experimentation with testing techniques: An infrastructure and its potential impact. *Empirical Software Engineering, 10*(4), 405–435.

Do, H., Mirarab, S., Tahvildari, L., & Rothermel, G. (2010). The effects of time constraints on test case prioritization: A series of controlled experiments. *IEEE Transactions on Software Engineering SE-2, 36*(5), 593–617.

Do, H., Rothermel, G., & Kinneer, A. (2006). Prioritizing JUnit test cases: An empirical assessment and cost-benefits analysis. *Empirical Software Engineering, 11*(1), 33–70.

Elbaum, S., Kallakuri, P., Malishevsky, A. G., Rothermel, G., & Kanduri, S. (2003). Understanding the effects of changes on the cost-effectiveness of regression testing techniques. *Journal of Software Testing, Verification, and Reliability, 13*(2), 65–83.

Elbaum, S. G., Malishevsky, A. G., & Rothermel, G. (2000). Prioritizing test cases for regression testing. *In Proceedings of the 2000 ACM SIGSOFT international symposium on software testing and analysis* (*ISSTA 2000*), ACM, New York, NY, pp. 102–112, 2000.

Engström, E., Runeson, P., & Skoglund, M. (2010). A systematic review on regression test selection techniques. *Information and Software Technology, 52*(1), 14–30.

Engström, E., Skoglund, M., & Runeson, P. (2008). Empirical evaluations of regression test selection techniques: A systematic review. In *Proceedings of the second ACM-IEEE international symposium on empirical software engineering and measurement* (*ESEM '08*). ACM, New York, NY, USA, pp. 22–31.

Jorgensen, M., & Shepperd, M. (2007). A systematic review of software development cost estimation studies. *IEEE Transactions on Software Engineering, 33*(1), 33–53.

Kapfhammer, G. M. (2011). Empirically evaluating regression testing techniques: Challenges, solutions, and a potential way forward. In *Proceedings of the 2011 IEEE fourth international conference on software testing, verification and validation workshops* (ICSTW '11). IEEE Computer Society, Washington, DC, USA, pp. 99–102.

Kapfhammer, G. M., & Soffa, M. L. (2007). Using coverage effectiveness to evaluate test suite prioritizations. *Proceedings of 1st ACM international workshop on empirical assessment of software engineering languages and technologies: held in conjunction with the 22nd IEEE/ACM international conference on automated software engineering* (*ASE*) *2007* (WEASELTech '07), pp. 19–20.

King, G. (2007). An Introduction to the dataverse network as an infrastructure for data sharing. *Sociological Methods and Research, 36*(2), 173–199.

Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering: A systematic literature review. *Information and Software Technology, 51*(1), 7–15.

Kitchenham, B. A., Budgen, D., & Brereton, P. (2010). The value of mapping studies—A participant-observer case study. *Proceedings of evaluation and assessment in software engineering* (EASE 10), pp. 1–9.

Kitchenham, B. A., Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Technical report EBSE-2007-01.

Kitchenham, B. A., Dyba, T., & Jorgensen, M. (2004). Evidence-based software engineering. *Proceedings of the 26th international conference on software engineering* (ICSE '04), pp. 273–281.

Korel, B., & Koutsogiannakis, G. (2009). Experimental comparison of code-based and model-based test prioritization. *Proceedings of the IEEE international conference on software testing, verification, and validation workshops* (ICSTW '09), pp. 77–84.

Korel, B., & Koutsogiannakis, G. (2009). Experimental comparison of code-based and model-based test prioritization. In *Proceedings of the IEEE international conference on software testing, verification, and validation workshops* (ICSTW '09). IEEE Computer Society, Washington, DC, USA, pp. 77–84.

Krishnamoorthi, R., & Sahaaya Arul Mary, S. A. (2009). Factor oriented requirement coverage based system test case prioritization of new and regression test cases. *Information and Software Technology, 51*(4), 799–808.

Ma, Z., & Zhao, J. (2008). Test case prioritization based on analysis of program structure. *Proceedings of the 2008 15th Asia-Pacific software engineering conference*, pp. 471–478.
Petersen, K., Feldt, R., Shahid, M., & Mattsson, M. (2008). systematic mapping studies in software engineering. *Proceedings of 12th international conference on evaluation and assessment in software engineering* (EASE), Department of Informatics, University of Bari, Italy.
Qu, X., Cohen, M. B., & Woolf, K. M. (2007). Combinatorial interaction regression testing: A study of test case generation and prioritization. *Proceedings of ICSM* 2007, pp. 255–264.
Srikanth, H., & Williams, L. (2005). On the economics of requirements-based test case prioritization. *Proceedings of the seventh international workshop on economics-driven software engineering research* (EDSER '05), pp. 1–3.
Yoo, S., & Harman, M. (2012). Regression testing minimization, selection and prioritization: A survey. *Software Testing, Verification and Reliability, 22*(2), 67–120.

Author Biographies

Cagatay Catal is an Assistant Professor in the Department of Computer Engineering at Istanbul Kultur University, Turkey. He received the PhD degree in Computer Engineering from Yildiz Technical University in 2008 and the MSc degree in Computer Engineering from Istanbul Technical University in 2004. He has 10 years industry experience as a software engineer, project manager, and senior researcher. His research interests include software testing, software architecture, software quality, machine learning, and software product lines. He has published more than 40 refereed papers in the area of software engineering, and machine learning.

Deepti Mishra is an Assistant Professor in the Department of Computer Engineering at Atilim University, Turkey. She received the PhD in Computer Science from Rani Durgawati University, India in 2004 and the MSc degree in Computer Science and Applications from Jiwaji University, India in 1994. Her research interests include software testing, software quality, software process improvement, requirements engineering, and databases. She has published many research papers and book chapters at international and national levels. She has been granted the Department of Information Technology Scholarship of Government of India.