What a Long, Strange Trip It's Been[†]: Past, Present, and Future Perspectives on Software Testing Research

Vinícius H. S. Durelli, Rodrigo F. Araujo, Marco A. G. Silva, Rafael A. P. Oliveira, José C. Maldonado, Márcio E. Delamaro
Computer Systems Department – University of São Paulo (ICMC/USP)
São Carlos, São Paulo, Brazil
{durelli, rfaraujo, magsilva, rpaes, jcmaldon, delamaro}@icmc.usp.br

***Abstract*—Over the past 25 years the Brazilian Symposium on Software Engineering (SBES) has evolved to become the most important event on software engineering in Brazil. Throughout these years, SBES has gathered a large body of studies in software testing. Aimed at providing an insightful understanding of what has already been published in such event, we synthesized its rich 25-year history of research on software testing. Using information drawn from this overview we attempted to highlight which types of study have been the most applied for conveying software testing efforts. We also devised a co-authorship network to obtain a bird's-eye view of which research groups and scholars have been the most prolific ones. Moreover, by performing a citation analysis of the selected studies we set out to ascertain the importance of SBES in a wider scenario. Finally, borne out by the information extracted from the studies, we shed some light on the state-of-the-art of software testing in Brazil and provide an outlook on its foreseeable future.**

***Keywords*-Software Testing; systematic mapping.**

I. Introduction

Throughout the past two and a half decades, the Brazilian Symposium on Software Engineering (SBES) has established itself as an authoritative venue for Software Engineering (SE) research. Being the premier Brazilian conference on the topic, it has drawn the attention of researchers, academics, and practitioners alike, accumulating a large body of literature on virtually every topics of SE. Such comprehensive literature has featured technical papers on innovative research.

Given that software testing is an essential part of the SE classical literature, a great deal of SBES's 25-year body of research centers around such subject. However, despite this postulated relevance, to the best of our knowledge there are no in-depth studies focusing on providing an overview of what has been published in SBES on software testing. This special track, *"SBES is 25"*, opens up the opportunity to take stock of the extant literature relating to software testing, thereby filling in such research gap.

Toward this end, we went over all of SBES's literature. In effect, this examination was conducted as a systematic mapping study. Systematic mapping is a synthesis method that involves searching the literature to ascertain the nature, extent, and quantity of published research papers (i.e., primary studies) on a particular area of interest [72]. Mapping studies aggregate and categorize primary studies, yielding a synthesized view of the research area being considered. Moreover, it differs from informal literature reviews due to the fact that the approach used for searching is defined in a protocol and reported as an outcome. Such characteristics contribute to mapping studies being transparent, replicable, and updateable.

This paper outlines the results of the mapping study we have undertaken in order to classify and categorize evidence on software testing in the context of SBES. The primary contributions of our study are the identification of *(i)* which areas of software testing research have been most subjected to investigation as well as *(ii)* which areas that have not been given much attention in these past 25 years. In addition, another contribution consists of using our findings to *(iii)* point out the areas that the software testing research community needs to focus on, so that we can meet the stringent demands imposed by the types of systems that are being built today and will increasingly be built in the future.

The remainder of this paper is organized as follows. The protocol we have designed for the mapping study is described in Section II. Considering the identified studies, we compiled a classification regarding the study type (Section III) and contribution for software testing (Section IV). After analyzing them, we identified, in Section V, research groups and their role in SBES history. In Section VI we assessed the research relevance of the venue by measuring productivity and citations. The gathered information sets a baseline to compare SBES research against a broader scenario (Section VII). Threats to validity are summarized in Section VIII. Finally, we established some perspectives on the future of software testing research in Section IX.

II. Systematic Mapping Process

Mapping studies follow a fivefold process [72]: *(i)* definition of research questions, *(ii)* conducting the search for primary studies, *(iii)* screening of papers, *(iv)* keywording of abstracts,

[†] *What a Long, Strange Trip It's Been*, by Grateful Dead, is arguably one of the most famous lines in rock and roll. This snippet has entitled several books and articles since the song's release. Since it evokes a lifespan of constant changes, we argue that it fits perfectly to describe SBES and the myriad of efforts on software testing that have been published in such event.

and *(v)* data extraction and mapping. In what follows we briefly describe how each of these steps was conducted.

The research questions must embody the mapping study purpose. Hence, given that we set out to determine which software testing topics have been investigated in SBES and the studies relevance, assessed by productivity and impact of researchers that have contributed the most to the area, our three research questions reflect this purpose as follows:

RQ$_1$: which test techniques have been most investigated?
RQ$_2$: who are the most prolific researchers (i.e., who are the main researchers according to the number and relevance of contributions)?
RQ$_3$: what are the sorts of studies (e.g., empirical studies and experience reports) that have been published?

The search for primary studies involved going over all previous SBES and its companion Tools Session proceedings. In either case, proceedings fall under one of two categories: hard-copy and soft-copy. SBES proceedings, for instance, were issued as hard copies up to 1999. From 1999 on, these documents were issued in hard and soft copies (except for 2000 and 2003, available just as hard copies). From 1999 to 2009, most soft copies (1999, 2001–2002, 2004, 2006–2009) are available at BDBComp (Brazilian Digital Library on Computing). Since 2009, main track proceedings are indexed by IEEE Digital Library. From 1987 to 1991, the Tools Session proceedings were included in the main track. From 1992 on, they were available in their own proceedings, with the exception of 1994–1995, 1997, and 2000–2002, when they were once again embodied in SBES main track proceedings.

Due to the heterogeneous nature of proceedings, we were not able to use a unified search method to gather all the required studies. Furthermore, the distinct nature of the sources of studies precluded us from devising a search string during the conduction of this step. Rather, we managed to gather all SBES and Tools Session proceedings by the following means: *(i)* downloading soft copies from digital libraries, *(ii)* requesting soft and hard copies from fellow researchers, and *(iii)* borrowing hard copies from local libraries.

The third step, screening, aimed at determining which primary studies were relevant to answer our research questions. To this end, we examined all proceedings, and applied a set of inclusion and exclusion criteria to each study:

- **Inclusion criterion:**
  - Any paper that described one or more study regarding software testing was subjected to be included.
- **Exclusion criteria:**
  - Papers that do not present studies related to software testing (e.g., papers describing research on any of the other SE topics) were excluded.
  - Papers that report on insightful proposals for prospective software testing research but do not apply it to define a test-related technology (e.g., oracle, process, and technique) were excluded.

The inclusion and exclusion criteria were applied using the following procedure. Each of the authors selected a subset of proceedings to read. At first, based solely on title and abstract, we applied the previously established criteria to each paper. Afterward, primary studies deemed as relevant according to our criteria were read through. As we read each study we also sketched out a preliminary analysis and identified their main contributions. Eventually, we carried out a consensus process in order to decide on pending papers, where the authors discussed whether such papers met our criteria.

According to the information we have gathered, throughout SBES's history (from 1987 to 2010), 2521 papers have been submitted, from which 816 were accepted. We ended up with 111 candidate papers. However, after a close examination of these candidate studies, the initial set was reduced to 98. Fig. 1 shows the frequency of primary studies we have selected per year. The research on software testing has been quite constant throughout the period, with the exception of the first years. Considering the selectivity of the conference and its broad coverage, it is fair to say that software testing research plays a substantial role on SBES.

III. Classification According to Study Type

As previously stated, one of our research questions is concerned with categorizing SBES literature on software testing. Since formulating classification schemes to be used in evaluating papers is a controversial topic, rather than defining our own scheme, we decided to base our categorization on a scheme proposed by Wieringa et al. [107]. According to them, research can be classified into five categories: *(i)* validation research, *(ii)* evaluation research, *(iii)* solution proposals, *(iv)*

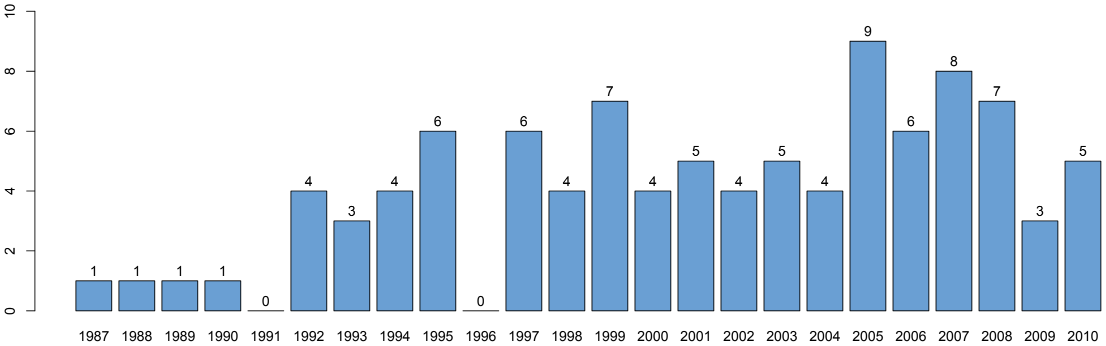


Fig. 1. Year-wise distribution of software testing primary studies.

opinion papers, and *(v)* experience papers. We were able to classify our parcel of SBES literature into four of these categories:

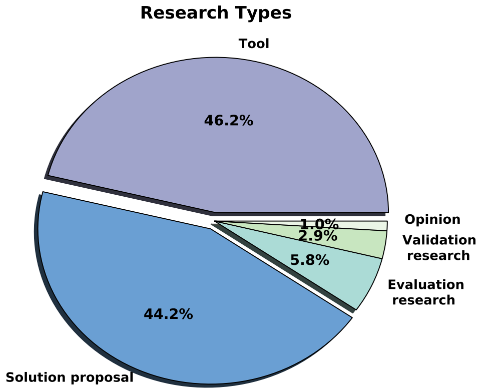
- **Solution proposal:** studies that report on a solution technique and argues for its usefulness, effectiveness, and relevance. The described solution technique is either novel or an extension of an existing technique. Studies in this category do not usually present in-depth validation of the described solution technique, but tend towards describing a proof-of-concept by means of an example, a running prototype, or even a sound argument.
- **Evaluation research:** studies focusing on evaluating a problem or an implemented solution in practice or real settings. To this end, these studies include case studies, controlled experiments, etc.
- **Validation research:** studies that investigate proposed solutions which have not yet been implemented in practice. Such investigations are performed systematically by means of experiments, prototyping, etc.
- **Opinion:** also known as position papers, such studies contain the authors' point of view. In most cases they are not accompanied by evidence in support of their claims.

Given that primary studies selected from the Tools Session proceedings do not fit well in the aforementioned categories, we had to extend the classification scheme. We devised an additional research type category in order to properly classify primary studies from the Tools Session proceedings:

- **Tool:** studies whose main contribution is outlining a tool (often in the form of a research prototype) that automates one or several software testing activities.

Fig. 2 shows the frequency of primary studies by research type. It is worth mentioning that some primary studies were assigned to more than one category, thus affecting the frequency count. The sum of the frequencies shown in Fig. 2 (104) is greater than the total of selected studies (98). It is fairly evident that tool and solution proposal are by far the research type of most of the selected primary studies. If we take into consideration just primary studies from SBES, the answer to $\mathbf{RQ}_3$ is that most software testing studies are solution proposals.

The gathered data also quantitatively characterizes controlled experiments published in SBES. As it can be seen from Fig. 2, there has been a lack of experimental studies. From the final set of 98 primary studies, only 9 of them report on formal experiments investigating the benefits of the presented techniques. These 9 studies were further broken down into two categories, resulting in 6 evaluation research and 3 validation research. We argue that more studies of these categories are necessary. In fact, from observing the evolution of the type of papers that have been published in SBES in the last five years, we believe that there is a tendency towards an increase in the number of papers combining characteristics of a solution proposal and evaluation or validation research. An indicator of such tendency is the ever-increasing adoption of the evidence-based paradigm within software engineering research.



Research Type	Frequency	Percentage
Tool	48	46.2%
Solution proposal	46	44.2%
Evaluation research	6	5.8%
Validation research	3	2.9%
Opinion	1	1.0%

Fig. 2. Distribution of primary studies by research type.

Only one primary study was classified as opinion paper. In effect, the latest call for papers of SBES technical research track does not even mention such types of contribution. However, distinct researchers are often invited as speakers, thereby somehow filling in such a gap with opinion panels.

IV. Classification Regarding Software Testing

Aiming to answer **RQ1**, we classified each primary study contribution according to the technology, source of information, and test phase. In effect, technology spans three subcategories: testing technique, oracle, and process, as it is shown in Fig. 3. Such classification was based upon several other studies described in the literature [10, 97], with minor adaptations, e.g., including random testing as a technique.

The primary studies focusing on oracles also make use of a testing technique. Therefore, they are classified in both categories. Likewise, there are studies that rely on more than on testing technique [59, 90].

One can notice that the structural criteria are the most investigated in SBES. We argue that possible reasons are: *(i)* almost half of our primary studies are based on source code, and *(ii)* it is the most consolidated technique within academy. Nonetheless, mutation testing also has a steady presence since 1993 and, recently, functional testing has captured some attention. These tendencies, as well as evidences extracted from the selected studies regarding their contributions towards oracle, process, source of information and test phase, are described in the following sections.

A. Source of Information

Considering our primary studies, we could ascertain that one research can explore more than one information source. We identified four different ones: *(i)* source code based, which is responsible for 40.8% of our primary studies, *(ii)* models stand for 27.5%, *(iii)* fault-based deals with 17.3% , *(iv)* software specification is accountable for 5.1%, and *(v)* digital images concern for 1% of the selected papers.

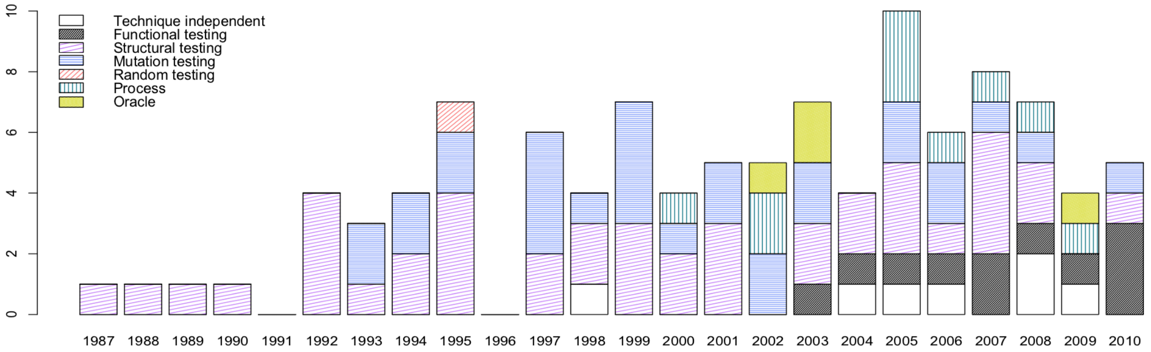


Fig. 3. Software testing technology per year.

B. Test Phase

We classified the selected studies according to four distinct phases. Unit testing, through which individual units (methods, procedures, or functions) are tested. Integration, whose purpose is verifying functional reliability requirements on major design items and combined software modules. System testing, which comprises all these phases, including a wide spectrum of testing activities. Regression testing, which is performed whenever an element of the system under test is changed.

Among the selected studies, 59 (61,2%) report on approaches tailored to unit testing. Studies dealing with integration testing account for 24,5%, from which 10 deal only with integration testing and 16 also employ unit testing. System-level testing was emphasized by 11 (13,3%). Only 1,0% of the selected papers (3) were about regression testing.

C. Software Testing Technique

Software testing techniques define the foundations to the systematic detection of faults given a test criterion. Such techniques can be classified into exhaustive, random and partition testing. Exhaustive testing entails executing the product under test against every possible input data and condition [61]. Although it is a strong technique, it is not always feasible. Some of the factors that may hinder its application are *(i)* the input data size, *(ii)* the combination of conditions, and *(iii)* undecidable computing functions. Random testing defines statistic-based criteria to model the input space and to sample data from the input space randomly [61]. Finally, partition testing defines subsets of the input domain given a criterion. Due to the fact that exhaustive testing is, in most situations, unfeasible, research is devoted mostly to random and partition testing techniques.

SBES research on software testing is mostly devoted to partition testing techniques: functional, structural and mutation testing. From 98 primary studies, only 1 was about random testing, 11 reported on functional testing approaches, 42 described research related to structural testing, mutation testing related research appeared in 29 studies, and technique independent research is described in 7 studies.

D. Oracle

An important element throughout the automation of testing activities is a testing oracle, which can be a manual or an automated one. Oracles are used to ascertain whether the application under test yields the expected outcome [9].

Despite their importance and pervasive nature, according to our results, oracles have not been getting much attention in SBES literature. Among the few (4) studies found, test oracles are exploited in support of mutation testing [87], test assertions [39], automated verification of specifications [70] and, finally, the test of systems or web applications with graphical user interfaces [71].

E. Process

Processes are a well-established research area. Nevertheless, in the context of SBES and Tools Session proceedings, papers addressing such subject have begun to emerge only in 2000 and currently account for 10.2% of our selected primary studies. From observing these studies, we conjecture that what might have driven the focus to this subject is the widespread adoption of agile processes.

V. Co-authorship Network

One of the goals of this study is the identification of the most prominent software testing researchers in SBES. Considering the 98 studies selected for analysis, 133 researchers contributed as authors, as shown in Fig. 4. The network of authors and papers clearly identifies the collaboration level among authors. The distinguished members of each group are usually at the center of each clique. This visual organization shows the main research groups in the Brazilian scenario, but it conceals quantitative information on the authors contribution.

Aiming at identifying the main researchers and answering **RQ**$_2$, we have compiled a ranking considering authors that have published at least three papers, as depicted in Table I. We also included the authors h-index according to Google Scholar and Scopus, with the purpose of showing the relevance of SBES researchers in a wider scenario. The h-index attempts to measure both the productivity and impact of a published paper. A scholar with an index of h has published h papers each of which has been cited by others at least h times.

Despite authors from Table I account for 89 primary studies, it is important to highlight that the studies were conducted collaboratively. Most of them ensued from the cooperation between advisors and advisees that takes place in a typical Brazilian academic setting. It is important to acknowledge

TABLE I
NUMBER OF PUBLICATIONS AND H_{index} PER AUTHOR.

Rank	Name	Papers	H_i Scholar	H_i Scopus
1	José Carlos Maldonado	46	22	9
2	Mario Jino	16	8	3
3	Márcio Eduardo Delamaro	14	14	6
4	Paulo César Masiero	13	15	6
5	Adenilso da Silva Simão	12	5	4
6	Auri Marcelo Rizzo Vincenzi	9	10	5
7	Ana Maria de Alencar Price	8	2	1
	Sandra Camargo Pinto Ferraz Fabbri	8	9	3
	Silvia Regina Vergilio	8	8	4
8	Eliane Martins	7	10	7
	Marcos Lordello Chaim	7	3	1
9	Patricia Duarte de Lima Machado	5	8	4
	Simone do Rocio Senger de Souza	5	5	0
10	Juliana Silva Herbert	4	2	0
	Otávio Augusto Lazzarini Lemos	4	6	4
11	Edmundo Sérgio Spoto	3	2	1
	Eric W. Wong	3	22	11
	Guilherme Horta Travassos	3	17	9
	Paulo Henrique Monteiro Borba	3	20	9
	Plínio Roberto Souza Vilela	3	3	1
	Reginaldo Ré	3	2	0
	Taisy Silva Weber	3	3	2

that once advisees establish their own research groups, most of them maintain a long-term interaction with their former research group.

The network of authors and papers gives a bird's-eye view of the main groups and their contribution to software testing in SBES. Considering only the authors from Table I and their academic peers, such co-authorship network (Fig. 5) comprises 8 groups, having most of the contributions related to Group 2.

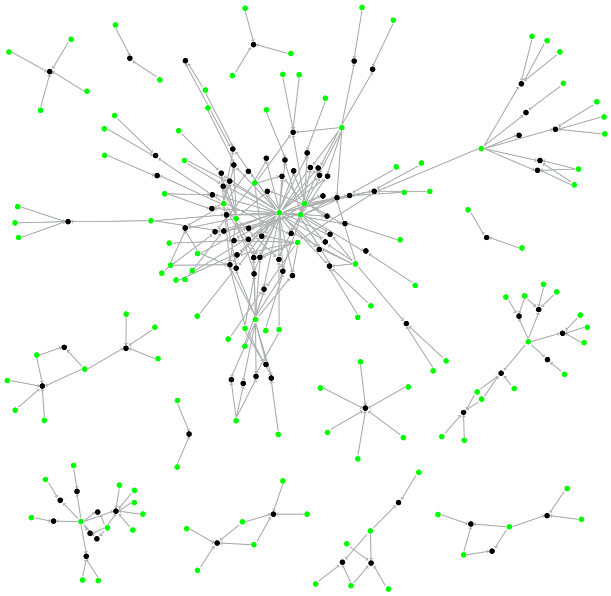


Fig. 4. Network of authors (green dots) and papers (black dots).

The first paper on software testing was published by the research group headed by Ana Price (Group 1). It was a short paper about a tool, PROTESTE [74], that automates the conduction of structural testing using control-flow-based criteria. Improved versions of this tool were later released, providing support for integration testing [76], data-flow criteria [75] and integration testing with respect to control and data-flow criteria [50]. Apart from PROTESTE, her research group addressed test data generation [51], testing of Smalltalk-based software [73] and processes [49].

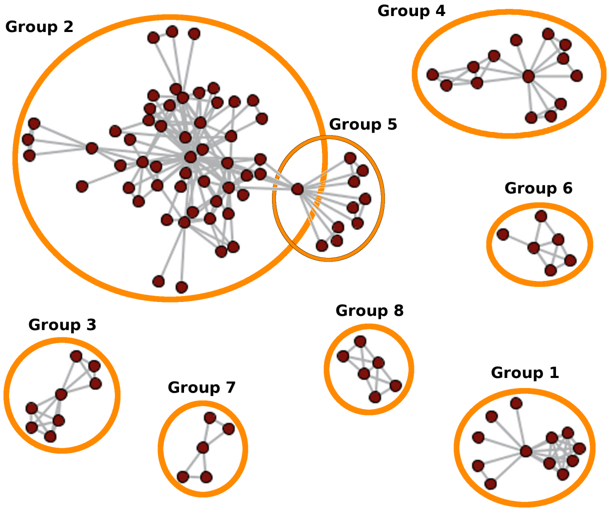


Fig. 5. Software testing research groups.

In 1988, Mario Jino and José Maldonado established a new research group at State University of Campinas (UNICAMP). Their first contribution was a seminal work whose main contribution is the potential uses criteria family [57]. It constituted a milestone for the largest research group on software testing in SBES, as it can be seen in Fig. 5 (Group 2). The group is led by Jino, Maldonado and Masiero. Its main focus is the structural and mutation testing. Regarding structural testing, there are studies on test criteria based on source code for procedural languages [21, 24, 48, 58, 94, 99, 100, 102, 104], aspect-oriented [31, 42, 53, 54, 78–80], services [34], models [89, 91], test data generation [6, 16, 18, 65, 101] and testing tools [13, 22, 32, 52, 55, 66, 90, 103, 106]. For mutation testing, this research group focus on the definition of test criteria based on source code [8, 17, 27–30, 64, 87, 88, 105, 108, 109] and models [20, 36–38, 85, 86, 88, 92, 93, 95].

In 2002, a third group was established and the software testing community has begun to spread. Led by Borba (Group 3), the first paper regards the testing of web applications [4]. Later on, his group worked on process estimation [5] and model-based testing of product lines [40]. Machado and her co-workers started their activity in 2003 (Group 4) on model-based testing [7, 19, 39, 70]. As for testing tools, a computational grid for software testing was implemented [33].

In the following year, Eliane Martins, which had already authored some papers in SBES [59], established her own group (Group 5) focusing on regression testing [41], test data generation [1, 2] and testing of software components [45]. Weber (Group 6) and Travassos (Group 7) created their groups in 2005. Weber addresses mutation testing using fault injection [43, 44, 96]. Travassos aims at integration testing [67], process [68], and test data generation [3]. Group 8 started in 2006 led by Mattiello Francisco. It has been conducting research on aerospace embedded system testing research [83, 84].

It is possible to mention that groups 5, 6 and 7 are headed by researchers who achieved their doctorate in foreign institutions. That can highlight the significance of interaction with international research groups. Our study outcomes also evince collaboration between local researchers, e.g., Maldonado and Masiero [38], and foreign researchers, such as Mathur and Wong [60]. Although in small number, the presence of non-Brazilian authors corroborates the importance and outreach of the conference.

VI. Citation Analysis

Aspiring to investigate the external impact of SBES software testing literature, we conducted a survey aiming at seeking data on the wide-ranging extent of citations of our primary studies. Our approach consisted in searching for citations related to our selected studies in several electronic databases, i.e., ACM, IEEE, Science Direct, and Google Scholar. ACM, IEEE, and Science Direct were selected because they are deemed as relevant scientific sources on Software Engineering. Google Scholar was also used due its widespread coverage, holding even papers that belong to non-indexed events (which is the case of most SBES's literature).

In order to retrieve statistical data with respect to the number of citations of each study, we used the aforementioned electronic databases to search for occurrences of the title of each study. We found out that the total amount of citations of our 98 primary studies is 276: 21 citations in ACM, 5 in IEEE, 9 in Science Direct, and 241 in Google Scholar.

Based on these numbers, we established an Impact Factor (IF) of our software testing studies. An IF is an index based on the frequency of citations of a given paper, and is used to rate its relevance. A three-year period is generally used to calculate an IF. In this scenario, we calculated it by dividing the sum of studies of the previous three years regarding the evaluating date, which were cited in the same concerned year, by the total number of software testing papers published in that period of time. This calculation was performed for each of the electronic databases. Nonetheless, the obtained results were not significant, as the IF was zero in most instances.

Our alternative was to collect the raw number of citations of each of the electronic databases and group it by three-year ranges, as it is presented in Fig. 6. We can point out the papers of Wong et al. [108] and Vincenzi et al. [106], which received most of the citations. Wong et al. [108] hold 5 citations in ACM, 3 in IEEE, 3 in Science Direct and 32 in Google Scholar. Vincenzi et al. [106] feature 5 citations in ACM, 0 in IEEE, 2 in Science Direct and 23 in Google Scholar.

We argue that SBES citations tend to increase because the venue, since 2009, is being indexed by IEEE. Prior to this period, the number of citation may have been bogged down due to the reasonable complications concerning obtaining a copy of a published paper.

VII. An Outlook on Software Testing in Brazil

Since the first edition of SBES, held in 1987, there have been studies on software testing. The first six years were exclusively devoted to researches taking into account a criterion implementation or improvement – the structural one – to a given context. Since 1993, mutation testing has also been continuously investigated and reported in SBES. It was only in 2000 that results were published regarding other topic than testing techniques. Indeed, the last decade was characterized by a healthy diversification of research topics on software testing, exploring testing process and oracles, yet with a stronger focus on testing techniques.

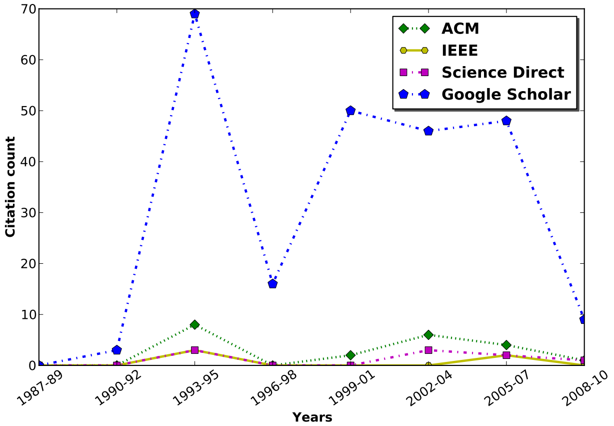


Fig. 6. Number of citations per year.

According to the growth of technologies, researchers attempt to improve software products quality as the software testing evolves. Bertolino [10] argues that some software testing research areas are considered as future trends to the software testing evolution, e.g., test data generation, domain-specific approaches, model based testing, and evaluation and validation research. We analyzed these topics taking into consideration our selected primary studies, aiming to indicate which areas could be further explored in the future.

For the establishment of a testing theory, it is important to evaluate the effectiveness of a criterion. According to Harrold [47], there is a demand for research studies providing analytical, statistical, or empirical evidence of the effectiveness of a criterion in revealing faults and of which fault classes are addressed. In particular, a common sense is to generally use a combination of testing techniques, even if one is deemed as more powerful, considering that they can handle different types of faults [56].

SBES papers which seek empirical evidence to address evaluation and validation research have begun to be published in 1992. Altogether, they are responsible for 9.2% of primary studies. We consider it as a poor percentage amount, which can contribute to the shortage of industrial cooperation. Moreover, such academic empirical studies may not be representative in terms of scalability and complexity. They have a tendency to be stronger in terms of internal validity, i.e., to generate accurate conclusions from a given data set, and to lack support for external validity, i.e., to generalize the results to an industrial context [14]. We argue that this trend will continue in the near future, since there is still a large gap between research published in SBES and what may be practical in industrial settings.

Another demand to improve the testing activity is automating the generation of input data. Such area has always raised interest over the years, resulting in several studies focusing on this subject [62]. Nevertheless, according to Bertolino [10], such efforts have produced no significant impact on the industry until the mid-2000s, when the test data was usually manually generated.

In SBES, this research area has started to be addressed in 1993. Until its latest edition, 11.2% of the publications addresses such subject. Given that there is not a constant presence of papers throughout the years dealing with test data generation, we consider that a greater emphasis could have been given to such area. The combination of advances in technologies such as symbolic execution, model checking, static and dynamic analysis, coupled with progress in the standardization of models and the growth of the available computational resources, provide positive expectations regarding the ever increasing automation of this task [10].

In 1995, the first studies regarding model based testing have arisen. Although the idea comes from the mid-fifties by the generation of data from FSMs, the intention to apply it in real-world applications has been growing in recent years. In this approach, the testing activity takes place in a more abstract level, that can occur even before the software is coded. This can lead to a more efficient process with significant cost reduction and a final product with higher quality [15]. Another benefit is that it can make it easier to automate the generation of test sets.

SBES, which holds 25.5% of studies addressing model based testing, has contributed with its adoption. Currently, the level of industrial acceptance is not high [10]. However, researchers are shedding their efforts in this direction. In particular, the reactive systems industry has been investing in this area by focusing on specific domains.

Domain-specific approaches emerge as an efficient solution allowing field experts to express abstract specifications that meet their demands. The testing activity may be specific to the scope of an application. Thus, the testing task can make use of particular approaches, processes and tools, taking into consideration the domain requirements of the underlying product.

Another information that can be retrieved from our primary studies is that there has been few interest in developing software testing approaches geared towards specific domains. In SBES, the first paper was published in 2002. We classified them as web applications, GUI, and embedded systems. Altogether, 9.1% of the research encompass such subject. A field that presented a prominent growth in the last few years and could be addressed more rigorously is the embedded systems one, taking into account that these kind of systems have become an integral part of everyone's daily life.

The research areas considered in our analysis have started to overlap and could be used to head upcoming research studies in SBES. A Venn diagram was drawn in order to gain a better understanding of the distribution of such studies, as it can be seen in Fig. 7. We can mention a SBES Tools paper that overlaps among 3 of the selected research areas, focusing on test data generation for embedded systems models [6].

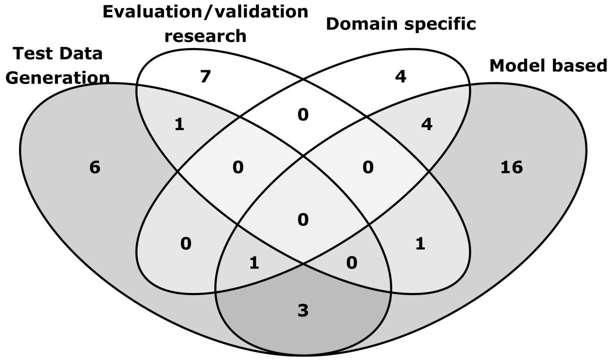


Fig. 7. Overview of published papers in SBES regarding future trend topics.

Although the event has already published research on most trend topics, there are some areas that have not received too much attention yet. For instance, software testing process and test oracles are not sufficiently discussed in the venue. Agile methods and processes, which highlight the importance of software testing, are other examples of subjects that have not been covered in-depth in the venue. Despite the existence of conferences devoted to agile development, studies about its software testing facet could have been reported in SBES, thereby integrating evidences about practices whose side-effects are closely related to software testing, e.g., test-driven development.

VIII. Threats to Validity

We have considered papers published in the Tools Session track in the same regard as papers published in the SBES's main track. That may be considered unfair and biased, as papers centered around tools are shorter. Nonetheless, the relevance and scientific rigor of several primary studies from the Tools Session is corroborated by the citation analysis shown in Section VI. As can be seen in that section, several primary studies have achieved a considerable number of citations [29, 106].

Regarding the conference quality assessment perspective, after undergoing a rigorous peer review process, every paper accepted in a Tools Session track needs to be presented at the conference. In addition, currently, one of the authors must dedicate a two-day-time slot for clarifying any questions about the tool being presented, providing more opportunities for stimulating interactions. Therefore, we argue that the inclusion of Tools Session studies is required to ensure that our overview encompasses all technical hurdles that have been mitigated by software testing researchers.

We may also mention that the current unavailability of an index for the entire SBES body of knowledge poses a threat due to the fact that it may hamper the proper replication of our study.

IX. Concluding Remarks

In this paper we have described an overview of the software testing literature that has been published in SBES. Thus, the major contribution of this paper should be seen as a picture of this research discipline in Brazil. We believe that such up-to-date overview of the event's history regarding software testing can benefit practice and future research. Therefore, during the conduction of our systematic mapping, we have privileged broadness rather than depth. We did not delve deep into each software testing study, instead we only mapped out and described fundamental characteristics of each study, emphasizing the ones that contributed the most to outlining the big picture.

The result of our mapping study reveals that from the inception of SBES and its Tool Session track (1987) to 2010 there has been at least a study per year reporting on software testing research – apart from 1991 and 1996. Structural testing is the criteria emphasized by most of these contributions. Another subject investigated by our mapping study is what source of information has been used in most studies. We found out that source code along with high-level models are by far the information sources more widely employed in academic settings. This survey also provides some evidence that there is room for research tailored towards automating several software testing activities. For instance, out of 98 studies, only 4 describe research on implementing testing oracles.

Information drawn from our mapping study was also used to devise a co-authorship network, which illustrates how authors have been collaborating in the software testing research area as well as which are the most prolific ones. As each research group addresses a different topic, it would be important to improve the interaction between these groups, covering the gap on software process and establishing the foundations for a strong cooperation with the industry.

ACKNOWLEDGEMENT

This research was funded by FAPESP (grant 2009/00632-1) and CNPq (grants 141976/2008-0, 142381/2009-8 and 559915/2010-1). We would like to thank all colleagues who contributed to this study, especially Fabiano Ferrari, Adenilso Simão and Sandra Fabbri. We are also grateful to the anonymous reviewers for their truly insightful comments.

REFERENCES

[1] B. T. Abreu, E. Martins, and F. L. Sousa. Automatic test data generation for path testing using a new stochastic algorithm. In *19th SBES*, pp. 247–262, Uberlândia, Brazil, Oct. 2005.

[2] B. T. Abreu, E. Martins, and F. L. de Sousa. Generalized extremal optimization: A competitive algorithm for test data generation. In *XXI SBES*, pp. 342–358, João Pessoa, Brazil, Oct. 2007.

[3] P. P. B. Albuquerque, J. L. Massollar, and G. H. Travassos. ModelT2: Apoio ferramental à geração de casos de testes funcionais a partir de casos de uso. In *24th SBES – 17th Tools Session*, pp. 55–60, Salvador, Brazil, Sept. 2010.

[4] E. Aranha and P. Borba. Testes e geração de código de sistemas web. In *16th SBES*, pp. 114–129, Gramado,Brazil, Oct. 2002.

[5] E. Aranha, F. Almeida, T. Diniz, V. Fontes, and P. Borba. Automated test execution effort estimation based on functional test specifications. In *22nd SBES – 15th Tools Session*, pp. 1–6, Campinas, Brazil, Oct. 2008.

[6] R. F. Araujo and M. E. Delamaro. TeTooDS – testing tool for dynamic systems. In *22nd SBES – 15th Tools Session*, pp. 1–6, Campinas, Brazil, Oct. 2008.

[7] D. L. Barbosa, W. L. Andrade, P. D. L. Machado, and J. C. A. Figueiredo. SPACES – uma ferramenta para teste funcional de componentes. In *18th SBES – 11th Tools Session*, pp. 55–60, Brasília, Brazil, Oct. 2004.

[8] E. F. Barbosa, A. M. R. Vincenzi, and J. C. Maldonado. Uma contribuição para a determinação de um conjunto essencial de operadores de mutação no teste de programa C. In *12th SBES*, pp. 103–120, Maringá, Brazil, Oct. 1998.

[9] L. Baresi and M. Young. Test oracles. Technical Report CIS-TR-01-02, University of Oregon, Dept. of Computer and Information Science, Eugene, OR, USA, Aug. 2001.

[10] A. Bertolino. Software testing research: Achievements, challenges, dreams. *Future of Software Engineering*, 0:85–103, 2007.

[11] F. M. Besson, D. M. Beder, and M. L. Chaim. Um conjunto de ferramentas para modelagem de casos de teste de aceitação de aplicações web. In *23rd SBES – 16th Tools Session*, pp. 13–18, Fortaleza, Brazil, Oct. 2009.

[12] L. Biasi and K. Becker. Geração automatizada de drivers e stubs de teste para JUnit a partir de especificações U2TP. In *20th SBES*, pp. 33–48, Florianópolis, Brazil, Oct. 2006.

[13] K. N. Borges, F. S. Ramos, J. C. Maldonado, M. L. Chaim, and M. Jino. POKE-TOOL versão CLIPPER – uma ferramenta para suporte ao teste estrutural de programas baseado em análise de fluxo de dados. In *9th SBES – 2nd Tools Session*, pp. 483–486, Recife, Brazil, Oct. 1995.

[14] L. Briand and Y. Labiche. Empirical studies of software testing techniques: challenges, practical strategies, and future research. *SIGSOFT Softw. Eng. Notes*, 29:1–3, Sept. 2004.

[15] M. Broy, B. Jonsson, J. Katoen, M. Leucker, and A. Pretschner, editors. *Model-Based Testing of Reactive Systems, Advanced Lectures*, volume 3472 of *Lecture Notes in Computer Science*. Springer-Verlag, 2005.

[16] P. M. S. Bueno and M. Jino. Geração automática de dados e tratamento de não executabilidade no teste estrutural de software. In *13th SBES*, pp. 307 – 322, Florianópolis, Brazil, Oct. 1999.

[17] D. N. Campanha, S. R. S. Souza, and J. C. Maldonado. Teste de mutação nos paradigmas procedimental e OO: Uma avaliação no contexto de estrutura de dados. In *24th SBES*, pp. 91–100, Salvador, Brazil, Sept. 2010.

[18] M. A. P. Candolo, A. S. S. Simão, and J. C. Maldonado. MGASet – uma ferramenta para apoiar o teste e validação de especificações baseadas em máquinas de estado finito. In *15th SBES – 8th Tools Session*, pp. 386–391, Rio de Janeiro, Brazil, Oct. 2001.

[19] E. G. Cartaxo, P. D. L. Machado, F. G. O. Neto, and J. F. S. Ouriques. Usando funções de similaridade para redução de conjuntos de casos de teste em estratégias de teste baseado em modelos. In *22nd SBES*, pp. 1–16, Campinas, Brazil, Oct. 2008.

[20] R. A. Carvalho, S. C. P. F. Fabbri, and J. C. Maldonado. Um estudo sobre a avaliação do custo de aplicação da análise de mutantes na validação de máquinas de estados finitos. In *13th SBES*, pp. 323 – 338, Florianópolis, Brazil, Oct. 1999.

[21] M. Chaim, J. Maldonado, and M. Jino. Modelando a determinação de potenciais DU-caminhos através de análise de fluxo de dados. In *3rd SBES*, pp. 239–251, Recife, Brazil, Oct. 1989.

[22] M. L. Chaim, M. Jino, and J. C. Maldonado. POKE-TOOL – estado atual de uma ferramenta para teste estrutural de software baseado em análise de fluxo de dados. In *12th SBES – 5th Tools Session*, pp. 37–45, Maringá, Brazil, Oct. 1998.

[23] R. Coelho, Élder Cirilo, U. Kulesza, A. von Staa, C. Lucena, and A. Rashid. JAT framework: Creating JUnit-style test for multi agent systems. In *21st SBES – 14th Tools Session*, pp. 40–46, João Pessoa, Brazil, Oct. 2007.

[24] A. N. Crespo, A. Pasquini, M. Jino, and J. C. Maldonado. Cobertura dos critérios potenciais-usos e a confiabilidade do software. In *9th SBES*, pp. 379–394, Fortaleza, Brazil, Oct. 1997.

[25] P. N. Cruz Filho and S. R. Vergilio. Uma ferramenta para o teste de web services baseado em perturbação de dados dirigida por padrões. In *21st SBES – 14th Tools Session*, pp. 63–69, João Pessoa, Brazil, Oct. 2007.

[26] A. Dantas, M. Gaudencio, F. Brasileiro, and W. Cirne. Ob-

taining trustworthy test results in multi-threaded systems. In *22nd SBES*, pp. 33–48, Campinas, Brazil, Oct. 2008.

[27] M. Delamaro, M. Pezzè, A. M. R. Vincenzi, and J. C. Maldonado. Mutant operators for testing concurrent java programs. In *15th SBES*, pp. 272–301, Rio de Janeiro, Brazil, Oct. 2001.

[28] M. E. Delamaro and J. C. Maldonado. Teste de integração: Projeto de operadores para o critério mutação de interface. In *9th SBES*, pp. 413–428, Fortaleza, Brazil, Oct. 1997.

[29] M. E. Delamaro, J. C. Maldonado, M. Jino, and M. L. Chaim. Proteum: Uma ferramenta de teste baseada na análise de mutantes. In *7th SBES – Tools Session*, pp. 31–33, Rio de Janeiro, Brazil, Oct. 1993.

[30] M. E. Delamaro, J. C. Maldonado, and E. Y. Nakagawa. PROTEUM/IM: Uma ferramenta de apoio ao teste de integração. In *9th SBES – 4th Tools Session*, pp. 487–490, Fortaleza, Brazil, Oct. 1997.

[31] M. E. Delamaro, P. A. Nardi, O. A. L. Lemos, P. C. Masiero, E. S. Spoto, J. C. Maldonado, and A. M. R. Vincenzi. Static analysis of java bytecode for domain-specific software testing. In *21st SBES*, pp. 325–341, João Pessoa, Brazil, Oct. 2007.

[32] A. L. S. Domingues, A. S. S. Simão, A. M. R. Vincenzi, and J. C. Maldonado. EvalTool: Um ambiente de apoio à avaliação e seleção de ferramentas de teste para programas orientados a objetos. In *16th SBES*, pp. 384–389, Gramado,Brazil, Oct. 2002.

[33] A. N. Duarte, W. Cirne, F. Brasileiro, and P. D. L. Machado. Gridunit: Using the computational grid to speed up software testing. In *19th SBES – 12th Tools session*, pp. 33 – 38, Uberlândia, Brazil, Oct. 2005.

[34] M. M. Eler, M. E. Delamaro, J. C. Maldonado, and P. C. Masiero. Built-in structural testing of web services. In *24th SBES*, pp. 71–80, Salvador, Brazil, Sept. 2010.

[35] M. Emer and S. Vergilio. Selection and evaluation of test data sets based on genetic programming. In *16th SBES*, pp. 82–97, Gramado,Brazil, Oct. 2002.

[36] S. C. P. F. Fabbri, J. C. Maldonado, P. C. Masiero, and M. E. Delamaro. Aplicação da análise de mutantes na validação de especificações baseadas em Redes de Petri. In *8th SBES*, pp. 423–437, Curitiba, Brazil, Oct. 1994.

[37] S. C. P. F. Fabbri, J. C. Maldonado, M. E. Delamaro, and P. C. Masiero. PROTEUM/FSM – uma ferramenta para apoiar a validação de máquinas de estado finito pelo critério análise de mutantes. In *9th SBES – 2nd Tools Session*, pp. 475–478, Recife, Brazil, Oct. 1995.

[38] S. C. P. F. Fabbri, J. C. Maldonado, and P. C. Masiero. Aplicação do critério análise de mutante na validação de especificações baseadas em Statecharts. In *9th SBES*, pp. 429–444, Fortaleza, Brazil, Oct. 1997.

[39] C. M. Farias and P. D. L. Machado. Um método de teste funcional para verificação de componentes. In *17th SBES*, pp. 193–208, Manaus, Brazil, Oct. 2003.

[40] F. Ferreira, L. Neves, M. Silva, and P. Borba. TaRGeT: a model based product line testing tool. In *24th SBES – 17th Tools Session*, pp. 67–72, Salvador, Brazil, Sept. 2010.

[41] W. I. O. Fidelis and E. Martins. FireWeb: Uma ferramenta de suporte aos testes de regressão de aplicações web. In *18th SBES – 11th Tools Session*, pp. 1–6, Brasília, Brazil, Oct. 2004.

[42] I. G. Franchin, O. A. L. Lemos, and P. C. Masiero. Pairwise structural testing of object and aspect-oriented java programs. In *21st SBES*, pp. 377–393, João Pessoa, Brazil, Oct. 2007.

[43] J. Gerchman, G. Jacques-Silva, R. J. Drebes, and T. S. Weber. Ambiente distribuído de injeção de falhas de comunicação para teste de aplicações java de rede. In *19th SBES*, pp. 232–246, Uberlândia, Brazil, Oct. 2005.

[44] J. Gerchman, C. C. Menegotto, and T. S. Weber. Geração de cargas de falha para campanhas de injeção de falhas a partir de modelos uml de teste. In *22nd SBES*, pp. 17–32, Campinas, Brazil, Oct. 2008.

[45] P. A. C. Guerra, C. S. Araújo, C. R. Rocha, and E. Martins. CBDUnit – uma ferramenta para testes unitários de componentes. In *19th SBES – 12th Tools session*, pp. 1 – 6, Uberlândia, Brazil, Oct. 2005.

[46] J. A. Góes and D. P. Renaux. PERF – um ambiente de desenvolvimento voltado para testes temporais de software. In *15th SBES – 8th Tools Session*, pp. 404–409, Rio de Janeiro, Brazil, Oct. 2001.

[47] M. J. Harrold. Testing: a roadmap. In *Proceedings of the Conference on The Future of Software Engineering*, pp. 61–72, New York, NY, USA, 2000. ACM.

[48] A. C. Hausen, S. Vergilio, S. R. S. Souza, P. S. L. Souza, and A. S. Simão. ValiMPI: Uma ferramenta para teste de programas paralelos. In *20th SBES – 13th Tools Session*, Florianópolis, Brazil, Oct. 2006.

[49] J. Herbert and A. Price. Ambiente de apoio ao teste cooperativo de software – Testflow Manager. In *14th SBES – 7th Tools Session*, pp. 380–382, João Pessoa, Brazil, Oct. 2000.

[50] J. S. Herbert and A. M. de Alencar Price. PROTESTE+: Ambiente de validação automática de qualidade de software através de técnicas de teste e de métricas de complexidade. In *9th SBES – 2nd Tools Session*, pp. 487–490, Recife, Brazil, Oct. 1995.

[51] J. S. Herbert and A. M. A. Price. Estratégia de geração de dados de teste baseada na análise simbólica e dinâmica do programa. In *9th SBES*, pp. 397–411, Fortaleza, Brazil, Oct. 1997.

[52] P. S. L. Júnior, J. C. Maldonado, and M. Jino. POKE-TOOL versão COBOL. In *6th SBES – Tools Session*, pp. 24–27, Gramado, Brazil, Nov. 1992.

[53] O. A. L. Lemos and P. C. Masiero. Integration testing of aspect-oriented programs: a structural pointcut-based approach. In *22nd SBES*, pp. 49–64, Campinas, Brazil, Oct. 2008.

[54] O. A. L. Lemos, A. M. R. Vincenzi, J. C. Maldonado, and P. C. Masiero. Teste de unidade de programas orientados a aspectos. In *18th SBES*, pp. 55–70, Brasília, Brazil, Oct. 2004.

[55] J. J. Li, S. London, P. Vilela, and J. R. Horgan. xSUDS-SDL: A tool for diagnosis and understanding software specifications. In *13th SBES – 6th Tools Session*, pp. 5 – 8, Florianópolis, Brazil, Oct. 1999.

[56] M. R. Lyu, editor. *Handbook of software reliability engineering*. McGraw-Hill, Inc., Hightstown, NJ, USA, 1996.

[57] J. Maldonado, M. Chaim, and Jino. Seleção de casos de testes baseada em fluxo de dados através dos critérios potenciais usos. In *2nd SBES*, pp. 24–35, Canela, Brazil, Oct. 1988.

[58] J. C. Maldonado, S. R. Vergílio, M. L. Chaim, and M. Jino. Critérios potenciais usos: Análise da aplicação de um benchmark. In *6th SBES*, pp. 357–370, Gramado, Brazil, Nov. 1992.

[59] E. Martins. Integrando injeção de falhas e testes formais na validação da tolerância a falhas. In *9th SBES*, pp. 223–239, Recife, Brazil, Oct. 1995.

[60] A. Mathur and W. Wong. Evaluation of the cost of alternate mutation strategies. In *7th SBES*, pp. 320 –335, Rio de Janeiro, Brazil, Oct. 1993.

[61] A. P. Mathur. *Foundations of Software Testing*. Pearson Education, 2008.

[62] P. McMinn. Search-based software test data generation: a survey: Research articles. *Softw. Test. Verif. Reliab.*, 14:105–156, June 2004.

[63] N. Mendes, R. Moraes, E. Martins, and H. Madeira. Jaca tool improvements for speeding up fault injection campaigns. In *20th SBES – 13th Tools Session*, Florianópolis, Brazil, Oct. 2006.

[64] E. Y. Nakagawa and J. C. Maldonado. ITOOL – uma ferramenta para injeção de defeitos de software. In *13th SBES – 6th Tools Session*, pp. 49 – 52, Florianópolis, Brazil, Oct. 1999.

[65] K. K. Nakazato, M. Alexandrino, J. C. Maldonado, S. C. P. F. Fabbri, and P. C. Masiero. MGASET – módulo de geração de seqüências de teste. In *9th SBES – 2nd Tools Session*, pp. 479–482, Recife, Brazil, Oct. 1995.

[66] P. A. Nardi, M. E. Delamaro, E. S. Spoto, and A. M. R. Vincenzi. Jabuti/bd: Utilização de critérios estruturais em aplicações de bancos de dados java. In *19th SBES – 12th Tools session*, pp. 45 – 50, Uberlândia, Brazil, Oct. 2005.

[67] A. C. D. Neto, G. M. P. S. Lima, and G. H. Travassos. FAROL: Uma ferramenta de apoio à aplicação de heurísticas de ordenação de classes para teste de integração. In *19th SBES – 12th Tools session*, pp. 13–18, Uberlândia, Brazil, Oct. 2005.

[68] A. D. Neto and G. H. Travassos. Maraká: Apoio ao planejamento e controle de testes de software. In *20th SBES – 13th Tools Session*, pp. 85–90, Florianópolis, Brazil, Oct. 2006.

[69] P. R. A. F. Nunes and A. C. V. Melo. Ocongra - uma ferramenta

para geração de grafos de fluxo de controle de objetos. In *18th SBES – 11th Tools Session*, pp. 7–12, Brasília, Brazil, Oct. 2004.

[70] K. A. Oliveira, P. D. L. Machado, and W. L. Andrade. CASLTEST – test case, test oracle and test data generation from CASL specifications. In *17th SBES*, pp. 73–78, Manaus, Brazil, Oct. 2003.

[71] R. A. P. Oliveira, M. E. Delamaro, and F. L. S. Nunes. O-FIm – oracle for images. In *23rd SBES – 16th Tools Session*, pp. 1–6, Fortaleza, Brazil, Oct. 2009.

[72] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson. Systematic mapping studies in software engineering. *12th International Conference on Evaluation and Assessment in Software Engineering*, pp. 71–80, 2008.

[73] I. M. Pinto and A. M. A. Price. Um sistema de apoio ao teste de programas orientados a objetos com uma abordagem reflexiva. In *12th SBES*, pp. 87–102, Maringá, Brazil, Oct. 1998.

[74] A. Price, C. Purper, and F. Garcia. PROTESTE: Projeto de uma ferramenta para teste de programas. In *1st SBES*, pp. 20–21, Petrópolis, Brazil, Oct. 1987.

[75] A. Price, F. Rosa, I. Bins, J. Silva, L. Santi, P. Dias, and V. Rombaldi. PROTESTE – ambiente de apoio ao teste de programas. In *6th SBES – Tools Session*, pp. 29–30, Gramado, Brazil, Nov. 1992.

[76] A. M. A. Price and A. F. Zorzo. Ambiente de apoio ao teste estrutural de programas. In *4th SBES*, pp. 169–182, Águas de São Pedro, Brazil, Oct. 1990.

[77] A. D. Rocha, A. S. Simão, J. C. Maldonado, and P. C. Masiero. Uma ferramenta baseada em aspectos para o teste funcional de programas java. In *19th SBES*, pp. 263–278, Uberlândia, Brazil, Oct. 2005.

[78] R. Ré and P. C. Masiero. Avaliação da abordagem incremental no teste de integração de programas orientados a aspectos. In *19th SBES*, pp. 168–183, Uberlândia, Brazil, Oct. 2005.

[79] R. Ré and P. C. Masiero. Integration testing of aspect-oriented programs: A characterization study to evaluate how minimize the number of stubs. In *21st SBES*, pp. 411–426, João Pessoa, Brazil, Oct. 2007.

[80] R. Ré, A. L. S. Domingues, and P. C. Masiero. Um catálogo de stubs para apoiar o teste de integração de programas orientados a aspectos. In *22nd SBES*, pp. 65–80, Campinas, Brazil, Oct. 2008.

[81] I. S. Santos, A. R. Santos, and P. A. Santos Neto. FERRARE GT: Automação de testes de desempenho e estresse via testes funcionais. In *24th SBES – 17th Tools Session*, pp. 49–54, Salvador, Brazil, Sept. 2010.

[82] J. B. Silva and A. M. A. Price. Métrica de complexidade de software baseado em critério de seleção de caminhos de teste. In *8th SBES*, pp. 471–485, Curitiba, Brazil, Oct. 1994.

[83] W. Silva, V. S. Junior, M. Mattiello-Francisco, and D. Passos. QSEE-TAS: Uma ferramenta para execução e relato automatizados de testes de software para aplicações espaciais. In *20th SBES – 13th Tools Session*, Florianópolis, Brazil, Oct. 2006.

[84] W. P. Silva, V. Santiago, N. L. Vijaykumar, and F. Mattiello-Francisco. SPAC: Ferramenta para processamento e análise de dados científicos no processo de validação de software em aplicações espaciais. In *21st SBES*, pp. 70–76, João Pessoa, Brazil, Oct. 2007.

[85] A. Simão and J. Maldonado. Proteum-RS/PN: A tool to support edition, simulation and validation of Petri Nets based on mutation testing. In *14th SBES*, pp. 227–242, João Pessoa, Brazil, Oct. 2000.

[86] A. Simão and J. Maldonado. Proteum-RS/PN: Uma ferramenta para apoiar a edição, simulação e validação de Redes de Petri baseada no teste de mutação. In *14th SBES – 7th Tools Session*, pp. 376–379, João Pessoa, Brazil, Oct. 2000.

[87] A. Simão, A. Vincenzi, and J. Maldonado. mudelgen: A tool for processing mutant operator descriptions. In *16th SBES – 9th Tools Session*, pp. 426–431, Gramado, Brazil, Oct. 2002.

[88] A. S. Simão and J. C. Maldonado. MuDeL: A language and a system for describing and generating mutants. In *15th SBES*, pp. 240–255, Rio de Janeiro, Brazil, Oct. 2001.

[89] A. S. Simão, S. R. S. de Souza, and J. C. Maldonado. A family of coverage testing criteria for Coloured Petri Nets. In *17th SBES*, pp. 209–224, Manaus, Brazil, Oct. 2003.

[90] A. S. Simão, A. M. Ambrósio, S. C. P. F. Fabbri, A. S. M. S. Amaral, E. Martins, and J. C. Maldonado. Plavis/FSM: an environment to integrate FSM-based testing tools. In *19th SBES – 12th Tools session*, pp. 57 – 62, Uberlândia, Brazil, Oct. 2005.

[91] A. S. Simão, A. Petrenko, and J. C. Maldonado. Experimental evaluation of coverage criteria for FSM-based testing. In *21st SBES*, pp. 359–374, João Pessoa, Brazil, Oct. 2007.

[92] S. R. S. Souza and J. C. Maldonado. Avaliação do impacto da minimização de conjuntos de casos de teste no custo e eficácia do critério análise de mutantes. In *9th SBES – 4th Tools Session*, pp. 445–460, Fortaleza, Brazil, Oct. 1997.

[93] S. R. S. Souza, J. C. Maldonado, and S. C. P. F. Fabbri. FCCE: Uma família de critérios de teste para validação de sistemas especificados em Estelle. In *15th SBES*, pp. 256–271, Rio de Janeiro, Brazil, Oct. 2001.

[94] E. Spoto, M. Jino, and J. Maldonado. Teste estrutural de software: Uma abordagem para aplicações de banco de dados relacional. In *14th SBES*, pp. 243–258, João Pessoa, Brazil, Oct. 2000.

[95] T. Sugeta, J. C. Maldonado, and S. C. P. F. Fabbri. PROTEUM-RS/ST – uma ferramenta para apoiar a validação de especificações statecharts baseada no critério análise de mutantes. In *13th SBES – 6th Tools Session*, pp. 41 – 44, Florianópolis, Brazil, Oct. 1999.

[96] J. C. Vacaro and T. S. Weber. Injeção de falhas na fase de teste de aplicações distribuídas. In *20th SBES*, pp. 161–176, Florianópolis, Brazil, Oct. 2006.

[97] S. Vegas, N. Juristo, and V. R. Basili. Maturing software engineering knowledge through classifications: A case study on unit testing techniques. *IEEE Transactions on Software Engineering*, 35(4):551–565, July 2009.

[98] J. Venâncio, C. Gusmão, E. Mendes, and E. Souza. RBTTool – uma ferramenta de apoio à abordagem de teste de software baseado em riscos. In *23rd SBES – 16th Tools Session*, pp. 7–12, Fortaleza, Brazil, Oct. 2009.

[99] S. Vergilio, J. C. Maldonado, and M. Jino. Caminhos não executáveis na automação da atividade de teste. In *6th SBES*, pp. 343–356, Gramado, Brazil, Nov. 1992.

[100] S. Vergilio, J. Maldonado, and M. Jino. Uma estratégia para geração de dados de teste. In *7th SBES*, pp. 306–319, Rio de Janeiro, Brazil, Oct. 1993.

[101] S. R. Vergilio, J. C. Maldonado, and M. Jino. Geração de dados de teste: Uma estratégia que preserva a hierarquia de critérios. In *9th SBES*, pp. 211–222, Recife, Brazil, Oct. 1995.

[102] S. R. Vergílio, J. C. Maldonado, and M. Jino. Caminhos não executáveis no teste de integração: Caracterização, previsão e determinação. In *8th SBES*, pp. 453–467, Curitiba, Brazil, Oct. 1994.

[103] P. R. S. Vilela, J. C. Maldonado, D. S. Cruzes, and M. Jino. ViewGraph: Visualizing control flow graphs and call graphs. In *12th SBES – 5th Tools Session*, pp. 29–35, Maringá, Brazil, Oct. 1998.

[104] P. R. S. Vilela, J. C. Maldonado, and M. Jino. Data flow based integration testing. In *13th SBES*, pp. 393 – 409, Florianópolis, Brazil, Oct. 1999.

[105] A. M. R. Vincenzi, J. C. Maldonado, E. F. Barbosa, and M. E. Delamaro. Operadores essenciais de interface: Um estudo de caso. In *13th SBES*, pp. 373 – 391, Florianópolis, Brazil, Oct. 1999.

[106] A. M. R. Vincenzi, W. E. Wong, M. E. Delamaro, and J. C. Maldonado. JaBUTi: A coverage analysis tool for java programs. In *17th SBES*, pp. 79–84, Manaus, Brazil, Oct. 2003.

[107] R. Wieringa, N. Maiden, N. Mead, and C. Rolland. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Requirement Engineering*, 11: 102–107, Dec. 2005.

[108] W. E. Wong, J. C. Maldonado, M. E. Delamaro, and A. P. Mathur. Constrained mutation in C programs. In *8th SBES*, pp. 439–452, Curitiba, Brazil, Oct. 1994.

[109] T. Yano, A. S. Simão, and J. C. Maldonado. Proteum/SML: Uma ferramenta de apoio ao teste de mutação para a linguagem Standard ML. In *17th SBES*, pp. 67–72, Manaus, Brazil, Oct. 2003.