Nascimento, Arnaldo Gil Sardella; Schmitz, Eber; Silva, Monica; Lima, Priscila
A MAPPING STUDY OF SCIENTIFIC MERIT OF PAPERS, WHICH SUBJECT ARE WEB
APPLICATIONS TEST TECHNIQUES, CONSIDERING THEIR VALIDITY THREATS
JISTEM - Journal of Information Systems and
Technology Management, vol. 15, e201815006, 2018
TECSI Laboratório de Tecnologia e Sistemas de Informação - FEA/USP

A MAPPING STUDY OF SCIENTIFIC MERIT OF PAPERS, WHICH SUBJECT ARE WEB APPLICATIONS TEST TECHNIQUES, CONSIDERING THEIR VALIDITY THREATS

Arnaldo Gil Sardella Nascimento[1] ⓘhttp://orcid.org/0000-0002-3534-8141
Eber Schmitz[1] ⓘhttps://orcid.org/0000-0002-4839-4606
Monica Silva[1] ⓘhttp://orcid.org/0000-0003-0951-6612
Priscila Lima[1] ⓘhttps://orcid.org/0000-0002-8515-9904
[1]Federal University of Rio de Janeiro, Rio de Janeiro, RJ, Brazil.

ABSTRACT

Progress in software engineering requires (1) more empirical studies of quality, (2) increased focus on synthesizing evidence, (3) more theories to be built and tested, and (4) the validity of the experiment is directly related with the level of confidence in the process of experimental investigation. This paper presents the results of a qualitative and quantitative classification of the threats to the validity of software engineering experiments comprising a total of 92 articles published in the period 2001-2015, dealing with software testing of Web applications. Our results show that 29.4% of the analyzed articles do not mention any threats to validity, 44.2% do it briefly, and 14% do it judiciously; that leaves a question: these studies have scientific value?

Keywords: Applications, Conclusion, Construction, External, Internal, Literature, Mapping, Qualitative, Quantitative, Reviews, Study, Systematic, Tests, Threats, Validity, Web.

INTRODUCTION

According to Moran (1995), the fear of a nuclear catastrophe and the incapacity to survive it, prompted scientists to create a non-hierarchical access structure that was implemented in universities, a network designed for military use, the Internet. Since then the network has been used for all kinds of research and business.

Mendes (2006) reports that the importance of the Internet for businesses is incontestable. Companies wishing to expand their business must be on the Web, and the Web site is the digital interface of the organization.

According to McGraw (2006) vulnerabilities found in Web applications place applications, and consequently, companies that use them, at risk and can lead to leakage of inside information, which may result in impairment of the business model and cause great economic impact.

A 2003 study conducted by the Business Internet Group San Francisco (BIG-SF) (2003) reported that about 70% of the sites and Web applications contain defects. In addition to financial costs, defects in web applications result in loss of credibility.

As a solution to the question of flaws in Web applications, studies indicate that we should increase the amount of testing in order to improve its quality. This is a difficult task, as pointed out by Dogan et al. (2014), since the heterogeneity of Web applications make them prone to errors and difficult to test. Its heterogeneity comes from the different languages used for developing applications that must work together and in harmony, and its client/server architecture with (asynchronous) HTTP request/response calls to synchronize the application state.

Li et al. (2014) shows that software testing has been widely used in order to ensure the technical quality of the various artifacts of a software project. In fact, software testing is a critical part of all high quality software production process and manufacturers devote approximately 40% of their time on testing to ensure the quality of the produced application.

However, contrary to the preconceived notion that software testing is used to demonstrate the absence of errors, the test is the process of finding as many errors as possible in order to improve software quality. This is due to the fact that demonstrate the absence of errors in the software requires testing all possible permutations for a given set of inputs which is obviously, for any non-trivial system, technologically and economically infeasible.

The quality assurance of Web applications, based on software testing, demands a large research effort in order to find efficient methods in terms of cost for different aspects of testing (Engström, 2013). Examples include the selection of test cases based on code and specification changes, the evaluation of selection techniques, the change impact analysis and regression testing for different applications. But, the application of methods and effective techniques in software testing requires the execution of empirical studies and that these studies report valid results.

The focus of this article is not on software testing, but the threats to the validity of the empirical studies used to establish the processes and methods created to implement such tests. The goal of this research is to map the threats (quantitative and qualitative) to the validity of empirical studies based on the relevant types of Web testing environment, over the period 2001-2015, and find out if these papers have the necessary empirical rigor to have scientific value.

Section 1 describes and classifies threats to validity of empirical studies and theoretical basis which exists behind the systematic literature reviews. Section 2, describes our research method.

Section 3 shows the research questions. Section 4 is devoted to threats to validity that our study is subject. Finally, the conclusions show the qualitative and quantitative results of this systematic literature mapping.

CONCEPTUAL FRAMEWORK

Man prefers to believe what

he prefers to be true.

(Francis Bacon)

Empirical Studies in Software Engineering

According to Zelkowitz et al. (1998), software engineering experiments do not have yet the maturity observed in other fields of science, because many papers do not follow a specific methodology and do not use appropriate validation techniques.

Experimental Software Engineering (ESE) aims to leverage the scientific method in this area. According to Boehm et al. (2005), is attributed to Victor R. Basili the first experiments to increase the rigor of research in software engineering and the creation of tools to enhance the quality of work, generating reliable evidence.

According to Miller (2000), the reliability of the evidence depends on the type of study carried out and also the quality with which their findings are replicated, because hardly the results of a single experiment are reliable. For a reliable conclusion to be achieved, the combination of the results of several experiments, with common variables, linked to the same hypothesis is required (replication). Among the reasons that bring into question the reliability of the evidence of the studies are: the flaws in the formulation of hypotheses, the error in study design and the error in the execution of the study.

An experimental study is an action carried out with the purpose of revealing something unknown or to prove a hypothesis. The researcher defines the data collection procedure, the data analysis technique and the meaning of the results (Basili et al., 1999). All experimental study should describe its goals, its method, results and conclusions. The research method establishes the accuracy of techniques and procedures adopted and the validity of evidence obtained from the study. The results, other than themselves, discuss threats to validity, limitations of the method used and the study variables. The conclusion confirms if the hypothesis described in the objective is confirmed or rejected and points future implications.

The main characteristics of an experimental study are: hypothesis, population, treatment, applicator and variables. Hypothesis - is the proposition, theory or assumption which can explain a certain behavior, regardless of the fact that it is confirmed or rejected and is free of human intention. Population - the concept of population in accordance with the findings of Juristo et al. (2001), for software engineering, it is an experimental study which is divided into people, products, problems and processes.

Usually empirical studies are made on a sample of the population (units). A single experiment is constituted of applying a combination of operations / treatments to a unit of study. Treatment is the action to be applied to an object of experimental study. It is specified as a set of controlled variables according to the experimental design (factors). To this end, it is observed levels on the variation. These

levels are the possible values of a factor during the study. Treatment defines the factors and levels to be applied to an element of the sample population. Variables - controlled features and measures in an experimental study.

Threats to the validity of empirical studies

Perry et al. (2000) defines threats to the validity of an empirical study as the influences that may limit our ability to interpret and draw conclusions from the study data.

According Wohlin et al. (2012), a fundamental question regarding the results of an experiment is how valid they are. The validity of results should be taken into account in the study design phase, because this issue can greatly influence the validity of the result of the study. The samples to be used in the study are drawn from a population of interest, and that its outcome is appropriate they have to be valid for this population. Subsequently, the study results can be generalized to a broader population.

According to Campbell and Stanley (1963) and Cook and Campbell (1979), there are four types of threats to the validity of an experiment:

1- Conclusion Validity: Ability to draw inaccurate conclusions from observations.

2- Internal Validity: Threats that may have affected the results and have not been properly taken into account.

3- Construction Validity: Threats on the relationship between theory and observation.

4- External validity: Threats that affect the generalizability of the results.

Threats to the validity of the conclusion concern the ability to take the correct conclusion on the relationship between treatment and outcome of an experiment. The validity of conclusion is subdivided into low statistical power, violated assumptions of the statistical tests, fishing and error rate, reliability of measures, reliability of the implementation of the treatment, random irrelevancies in the experimental setup and random heterogeneity issues.

The internal validity is established as the causal influences that may affect the independent variable without the knowledge of the researcher, so threaten the conclusion drawn from the experiment. Internal validity is subdivided into single group of threats, threats of multiple groups and social threats. Single group of threats: history, maturation, testing, instrumentation, statistical regression, selection, mortality, ambiguity about the direction of causal influence. Threats of multiple groups: interaction with the selection. Social threats: diffusion or imitation of treatments, compensatory equalization of treatment, compensatory rivalry, and resentment demoralized.

Construct validity concerns the generalization of the results of the experiment with the concept or theory behind the experiment. Some threats refer to the experimental design, others to social factors. The threats of design to construct validity cover issues that are related to the experience design and its ability to reflect the construction being studied. The behavior of those involved and experimenters can change based on the fact of being part of an experiment, which gives false results in experience.

Finishing forms of threats defined by Campbell and Stanley (1963) and Cook and Campbell (1979), we have threats to external validity. They are conditions that limit our ability to generalize the results of our experience with industrial practice. There are three types of interactions with treatment: people, place and time.

Systematic literature reviews

Kitchenham, et al. (2010) argues that empirical researchers in software engineering should develop its studies as evidence-based, as do researchers in medicine and sociology, and that the most reliable evidence comes from the aggregation of empirical studies on a given topic.

Systematic literature reviews are a means of adding knowledge about software engineering topic or research question and aim to be the most impartial possible to be auditable and repeatable.

The systematic literature reviews are classified into conventional and mapping studies. Conventional studies aggregate results related to a particular research question, for example, is technique testing "A" more effective in defect detection that the technique "B"? If there are enough primary studies comparable with quantitative estimates of the difference between the methods, meta-analysis can be used to perform an aggregation based on formal statistics. However, it was found that meta-analysis is rarely usable in systematic literature reviews of software engineering, because there are often insufficient primary studies.

Mapping studies are designed to find and classify primary studies on a specific subject area. They have coarse-grained research questions such as: "What we know about topic x?" and can be used to identify the available literature before performing conventional systematic literature reviews. They use the same methods for searching and extracting data as conventional systematic literature reviews, but most rely on the tabulation of primary studies in specific categories. In addition, some mapping studies are more concerned about how academics conduct research in software engineering rather than what is known about a given topic software engineering. The study reported in this paper is a mapping study.

RESEARCH METHOD

The revised articles were based on primary evidence of three empirical studies (Engström et al., 2010; Engström et al., 2011 and Doğan, 2014). To increase our research universe an automated search was conducted in the period from 2001 to 2015, to find jobs that contained empirical studies and made reference to types of tests applicable to Web environment and the threats to validity related to these studies.

Search strategy

Search strings

To have access to the articles (references) studies based taken in studies (Engström et al., 2010; Engström et al., 2011 and Doğan, 2014) for this study and to the automated search, in which the string was used: "web application testing" OR "web applications tests" OR "test of web applications" OR "testing web applications" OR "tests for web applications" AND "validity threats" OR "threats to validity".

Search engines

http://ieeexplore.ieee.org

http://dl.acm.org

http://scholar.google.com

http://academic.research.microsoft.com

http://citeseerx.ist.psu.edu

http://www.sciencedirect.com

Selection strategy

In the first step, the papers studies (Engström et al., 2010; Engström et al., 2011 and Doğan, 2014) formed the basis to the development of this study. All the references of these articles were used in the search engines mentioned in item 3.1.2, and we downloaded only files that were publicly available. Then, based on the title and summary, we selected those of various topics (since creation of algorithms to test methodologies), linked to Web applications testing, so we had a larger universe and more varied studies, resulting in 42 articles.

In a second step, the search strings mentioned in item 3.1.1 were applied to the same search engines, and adopting the procedure cited above to select new 50 articles. Adding the quantity of articles selected from the basis papers and the articles selected from the search strings, we reach a total of 92 papers for this study.

Figure 1. Strategy of search and selection of papers.
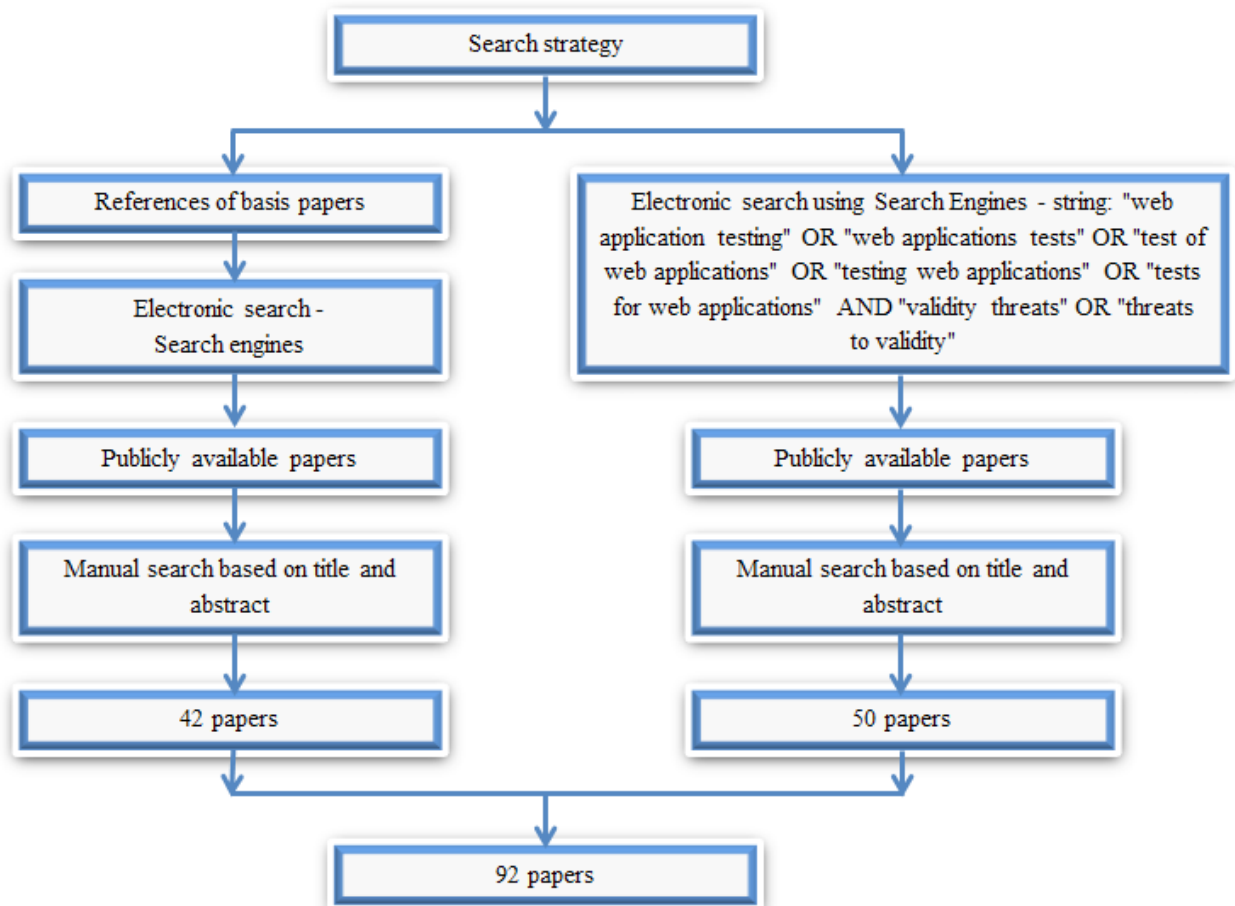
RESEARCH QUESTIONS

How many empirical studies related to Web application testing make reference to threats to validity of the same?

The results showed that of the 92 articles studied, 65 (E1 to E29 and E60 to E95) make reference to threats to validity (70.6%), whereas 16 articles (E3, E60, E62, E66, E69, E70, E71, E73, E74, E77, E78, E79, E85, E87, E89 and E93), 17.4% of the articles cite threats, but not classify them and 27 (E30, E32, E33, E34, E35, E36, E37, E38, E39, E40, E41, E42, E43, E44, E46, E48, E49, E50, E51, E52, E53, E54, E55, E56, E57, E58, E59) do not mention their threats to validity (29.3%).

For studies that mention the threats to validity, which are the most frequently cited threats?

According to the survey, the order (descending) in which threats to validity are quoted is: External Threat, Internal Threat, Threats of Construction, and Conclusion Threat. Table 1 presents threats as named in the articles. The complete tables listing the articles are available in Annex 1.

Table 1. Frequency of threats cited in the articles by type.

Threat	External	Internal	Construction	Conclusion
Frequency	45 (48.9%)	36 (39.1%)	24 (26.1%)	10 (10.9%)
Papers	E1, E2, E4, E5, E6, E7, E8, E9, E10, E11, E12, E13, E14, E15, E16, E18, E19, E20, E21, E23, E24, E25, E26, E27, E29, E61, E63, E65, E67, E68, E72, E75, E76, E80, E81, E82, E83, E84, E86, E88, E90, E91, E92, E94, E95.	E1, E2, E4, E5, E6, E7, E8, E9, E10, E11, E12, E13, E14, E15, E16, E18, E19, E21, E23, E24, E25, E27, E29, E65, E67, E68, E72, E75, E76, E80, E81, E86, E88, E91, E94, E95.	E1, E2, E4, E7, E8, E10, E11, E12, E14, E15, E16, E24, E27, E29, E61, E63, E64, E65, E68, E72, E75, E76, E90, E91.	E6, E7, E11, E12, E14, E24, E27, E65, E75, E90.

N.B.: Percentages do not add up to 100% since one paper may reference more than one thread.

How many studies mention threats to validity, but not classify (external, internal, construction and conclusion)?

A total of 16 papers were found: E3, E60, E62, E66, E69, E70, E71, E73, E74, E77, E78, E79, E85, E87, E89 and E93, all having an item on threats to validity, but not classified into external, internal, construction or conclusion.

How many studies mention the threats to the validity in a resumed form and how many do it thoroughly?

In the survey, the number of studies mentioning threats to validity judiciously is significantly less than that in a summary form. Table 2 shows the amount of threats described briefly and judiciously.

Table 2. Number of threats described briefly and judiciously.

Threat	External	Internal	Construction	Conclusion
Summarized threats	39 (60%)	31 (47.7%)	17 (26.2%)	9 (13.8%)
Detailed threats	7 (10.8%)	7 (10.8%)	7 (10.8%)	1 (1.5%)
Papers with summarized threats	E2, E4, E5, E6, E7, E8, E9, E10, E11, E12, E13, E14, E15, E18, E21, E23, E24, E25, E27, E29, E63, E67, E68, E72, E74, E75, E76, E80, E81, E82, E83, E84, E86, E88, E90, E91, E92, E94, E95.	E1, E2, E4, E5, E6, E7, E8, E9, E10, E11, E12, E13, E14, E15, E18, E19, E21, E23, E24, E29, E67, E68, E73, E74, E76, E80, E81, E88, E91, E94, E95.	E2, E4, E7, E8, E10, E11, E12, E14, E24, E29, E63, E68, E72, E75, E76, E90, E91.	E6, E7, E11, E12, E14, E24, E27, E75, E90.
Papers with detailed threats	E1, E16, E19, E20, E26, E61, E65.	E16, E19, E27, E65, E72, E75, E86.	E1, E15, E16, E27, E61, E64, E65.	E65.

What is the yearly relative frequency of articles making reference to threats to validity?

According to the study, the year 2014 was the year that presented more articles mentioning its threats to validity, a total of 29 articles. Table 3 presents the articles that refer to threats to validity, organized by year of publication.

According to the study, the year 2010 was the year that had more articles that do not cite its threats to validity, for a total of 5 items. Table 4 presents the articles that make no reference to threats to validity, organized by year of publication.

Which types of research are used?

According to our study, there are five types of research found in the group studied, listed below in descending order of quantity: Empirical Study, Systematic Literature Review, Mapping Study, Survey Study and Qualitative Research. Table 4 displays the amount of papers by type of search.

Table 3. Papers that do and do not reference threats to validity, organized by year of publication.

Year													
2001	2002	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
Frequency of articles that reference threats to validity													
1 (1.5%)	0 (0.0%)	0 (0.0%)	2 (3.0%)	1 (1.5%)	2 (3.0%)	2 (3.0%)	1 (1.5%)	7 (10.8%)	4 (6.1%)	3 (4.6%)	4 (6.1%)	29 (44.6%)	9 (13.8%)
Papers													
E1.	--	--	E6, E19.	E27.	E14, E17.	E15, E22.	E24.	E2, E4, E7, E8, E20, E21, E23.	E3, E9, E12, E16.	E5, E10, E18.	E11, E13, E28, E29	E25, E26, E60, E61, E64, E65, E66, E67, E68, E69, E70, E72, E74, E76, E78, E79, E81, E82, E84, E85, E86, E87, E89, E90, E91, E92, E93, E94, E95.	E62, E63, E71, E73, E75, E77, E80, E83, E88.
Frequency of articles that do not reference threats to validity													
0 (0.0%)	1 (3.7%)	2 (7.4%)	2 (7.4%)	3 (11.1%)	0 (0.0%)	3 (11.1%)	3 (11.1%)	5 (18.5%)	4 (14.8%)	2 (7.4%)	1 (3.7%)	1 (3.7%)	0 (0.0%)
Papers													
--	E40.	E30, E39.	E35, E43.	E32, E36, E50.	--	E37, E44, E59.	E38, E49, E51.	E33, E34, E42, E53, E58.	E52, E54, E55, E57.	E41, E46.	E48.	E56.	--

Table 4. Number of papers by type of research.

Type of Research	Empirical Study	Sistematic Literature Review	Mapping Study	Survey Study	Qualitative Survey
Frequency	83	4	3	1	1
Papers	E1, E3, E5, E6, E7, E8, E9, E10, E11, E12, E13, E14, E15, E16, E17, E18, E19, E20, E21, E22, E23, E25, E26, E27, E28, E29, E30, E32, E33, E34, E35, E36, E37, E38, E39, E40, E41, E42, E43, E44, E46, E48, E49, E50, E51, E53, E54, E57, E58, E59, E60, E61, E63, E64, E65, E66, E67, E68, E69, E70, E71, E72, E73, E74, E76, E77, E78, E79, E80, E81, E82, E83, E84, E85, E86, E87, E88, E90, E91, E92, E93, E94, E95.	E24, E4, E52, E62.	E55, E75, E89.	E56.	E2.

How many studies of each type take into account the threat validity type?

Table 5 summarizes the relation between threat types and types of research.

Table 5. Threats to validity by type of research.

	Empirical Study			
Threat	**External**	**Internal**	**Construction**	**Conclusion**
Frequency	42	26	21	10
Papers	E1, E5, E6, E7, E8, E9, E10, E11, E12, E13, E14, E15, E16, E18, E19, E20, E21, E23, E25, E26, E27, E29, E61, E63, E65, E67, E68, E72, E75, E76, E80, E81, E82, E83, E84, E86, E88, E90, E91, E92, E94, E95.	E1, E5, E6, E7, E8, E9, E10, E11, E12, E14, E15, E16, E27, E29, E65, E67, E68, E72, E75, E76 E80, E81, E86, E88, E94, E95.	E1, E7, E8, E10, E11, E12, E14, E15, E16, E27, E29, E61, E63, E64, E65, E68, E72, E75, E76, E90, E91.	E6, E7, E11, E12, E14, E27, E65, E75, E90, E91.
	Systematic Literature Review			
Threat	**External**	**Internal**	**Construction**	**Conclusion**
Frequency	2	2	2	1
Papers	E4, E24.	E4, E24.	E4, E24.	E24.

Table 5. Cont.

Mapping Study				
Threat	**External**	**Internal**	**Construction**	**Conclusion**
Frequency	1	1	1	1
Papers	E75.	E75.	E75.	E75.

Survey Study				
Threat	**External**	**Internal**	**Construction**	**Conclusion**
Frequency	0	0	0	0
Papers	--	--	--	--

Qualitative Survey				
Threat	**External**	**Internal**	**Construction**	**Conclusion**
Frequency	1	1	1	0
Papers	E2.	E2.	E2.	--

THREATS TO THE VALIDITY OF THIS STUDY

Internal validity

Research questions

The set of defined questions may not have covered the whole area of threats to validity in Web application testing. As we consider this as a viable threat, we discussed the issue in depth, in order to calibrate the questions. Thus, even if we have not selected the best set of questions, we seek to deeply address the most frequently asked and considered questions in the field.

Conclusion validity

In this article we focus on threats to validity to empirical studies in Web application testing. We found that 32% of the studies make no reference to any threats to validity. Also, in the studies that do it, (another 68%) only 14% of these do it judiciously. The large disparity between the two sets leaves no room to unrealistic inferences about the validity threats to the conclusion of our study.

Selected articles

It is possible that some relevant studies have not been chosen during the research process. We mitigated this threat, as far as possible, following the references of primary studies and also we use the search string to have a larger universe of research.

Validity of Construction

One of the detailing of the validity of construction is the threat that experimenters can influence the results of a study, consciously and unconsciously, based on what they expect from the experience.

In this study we do not see the threat to construct validity as a real threat, because when we started the study, we did not have preconceived ideas of results. We simply choose the items in a comprehensive manner and took the numbers found in the contents thereof. In fact, we were surprised with the result.

External validity

Based on what was said by Wohlin et al. (2012) if an experiment is carried out in a limited scope and its design is done to address issues related to the scope concerned, its results are valid in that context and in that case, adequate validity does not mean general validity, but if general conclusions will be taken from the experiment, its validity will have to cover a broader context, we conclude that under the 92 studies used, our results are valid, but if we consider the whole universe of articles relating to tests of Web applications, the selected amount of articles is reduced compared to the size of the universe. If we have chosen a larger number of studies maybe the conclusions drawn were different. So we think there is a real threat to external validity of this study because we cannot generalize it to the whole universe articles.

FINAL REMARKS

According to Sjøberg et al. (2007), progress in software engineering requires (1) more empirical studies of quality, (2) increased focus on synthesizing evidence and (3) more theories should be built and tested. Therefore, we need more studies in which we can trust, especially regarding their validity. In order to obtain this confidence, the collected information should be more focused on observation and experimentation than in deductive logic or mathematics. For this, we must use the same empirical methods used by the sciences who study real-world phenomena - empirical sciences.

According to Wainer (2007) and Travassos et al. (2002), for an experiment to be considered valid, it must provide a high level of confidence in the process of experimental investigation. This level of trust must permeate all elements involved in the process from the theoretical basis adopted, up to the final results. It is also important that the experiment takes into account all the threats to its validity: external, internal, construction and conclusion.

In order to have a wide range of studies that enable us to analyze more accurately the validity threats of software engineering experiments in Web application testing, we selected a number of articles with subjects ranging from algorithms to methodologies for web testing developed for such purposes.

The analysis of this wide range of papers shows that approximately 32% of the articles do not mention any threats to validity. On the other hand, in the remaining 68%, only 14% do it in a judiciously manner. In the set of the papers that mention validity threats (68%) one quarter do not provide validity classification. In the last five years we have had an increase in quantity of articles mentioning his threats to validity, and the year 2014 was the leader with 29 articles. The validity threats most frequently cited are respectively: external, internal, construction and conclusion.

Our research also suffers validity threats. The first one, conclusion validity is not critical since, as a mapping study, we restricted ourselves to find the relative frequency of the papers that mention threats to validity (32%) against the ones that do not mention any threats (68%). We consider that there is a small risk that our study suffers Internal validity threats because we addressed the most

frequently asked questions in this field. The selection of the articles, started with the papers referenced in the primary studies used followed by a search string to top up our universe of research. We do not think that the construct validity of our systematic mapping study is a real threat, because we had no preconceived ideas about the findings. The study consisted simply in collecting and counting information. Finally, we believe that external validity is the greatest threat to our study. Although we trust the results found in the 92 papers analyzed, we think that there is non-negligible chance of missing some major studies conducted where the analysis of all validity threats had been properly discussed.

Reiterating our previous observations, although this study is not totally immune to validity threats, the calculated figures lead us to the conclusion that the studies published in the time frame analyzed, lack scientific rigor both from the authors and the readers, and 73.6% of the studies have no scientific value.

BIBLIOGRAPHIC REFERENCES

BASILI, V. R., SHULL, F., and LANUBILE, F (1999). Building knowledge through families of experiments. *IEEE Transactions of Software Engineering*, **25.4**: 456-474.

BOEHM, BARRY; ROMBACH, HANS DIETER; ZELKOWITZ, MARVIN V. (2005). *Foundations of Empirical Software Engineering: The Legacy of Victor R. Basili*. 83. Germany, Springer-Verlag.

BUSINESS INTERNET GROUP SAN FRANCISCO (BIG-SF) (2003): Black Friday Report on Web Application Integrity, The Business Internet Group of San Francisco, http://www.tealeaf.com/news/press_releases/2003/0203.asp. Accessed 17-Oct-2014.

CAMPBELL, D.T., STANLEY, J.C. (1963). *Experimental and Quasi-experimental Designs for Research*. Boston, Houghton Mifflin Company.

COOK, T.D., CAMPBELL, D.T. (1979). *Quasi-experimentation – Design and Analysis Issues for Field Settings*. Boston, Houghton Mifflin Company.

DOĞAN, SERDAR, AYSU BETIN-CAN, and VAHID GAROUSI (2014). Web application testing: A systematic literature review. *Journal of Systems and Software* **91**: 174-201.

ENGSTRÖM, EMELIE, PER RUNESON, and MATS SKOGLUND (2010). A systematic review on regression test selection techniques. *Information and Software Technology* **52.1**: 14-30.

ENGSTRÖM, EMELIE, and PER RUNESON (2011). Software product line testing–a systematic mapping study. *Information and Software Technology* **53.1**: 2-13.

ENGSTRÖM, EMELIE (2013). Supporting Decisions on Regression Test Scoping in a Software Product Line Context–from Evidence to Practice. Dissertation. Lund University.

JURISTO, NATALIA; MORENO, ANA M (2001). *Basics of Software Engineering Experimentation*. Kluwer Academic Publishers.

KITCHENHAM, BARBARA, et al. (2010). Systematic literature reviews in software engineering–a tertiary study. *Information and Software Technology* **52.8**: 792-805.

KITCHENHAM, BARBARA A.; DYBA, TORE; JORGENSEN, MAGNE (2004). Evidence-based software engineering. *Internacional Conference on Software Engineering (ICSE'04). Edinburgh, UK: [s.n.], 2004. p. 273–281.*

LI, YUAN-FANG, PARAMJIT K. DAS, and DAVID L. DOWE (2014). Two decades of Web application testing - A survey of recent advances. *Information Systems* **43**: 20-54.

McGRAW, G. (2006). *Software Security – Building Security In*. São Paulo, Addison-Wesley.

MENDES, RENATO (2006): Quando a empresa não sabe o que fazer do seu site. http://www.milajuns.com.br/quando-a-empresa-nao-sabe-o-que-fazer-de-seu-site/. Accessed 25-Oct-2014.

MILLER, JAMES (2000). Applying meta-analytical procedures to software engineering experiments. *Journal of Systems and Software*, **54.1**: 29-39.

MORAN, JOSÉ MANUEL (1995). Novas tecnologias e o reencantamento do mundo. *Tecnologia educacional* **23.126**: 24-26.

PERRY, DEWAYNE E., ADAM A. PORTER, and LAWRENCE G. VOTTA (2000). Empirical studies of software engineering: a roadmap. *Proceedings of the conference on The future of Software engineering. ACM.*

PRESSMAN, R. (1992). *Software Engineering - A Practioner's Approach*. New York, Third Edition, McGraw Hill International Edition.

SCALET, D. (1995). Avaliação da Qualidade do Produto de Sotfware. *Workshop da Qualidade e Produtividade em Software e IX SBES/SBC.*

SHULL, F., SINGER, J., and SJOBERG, D. I. K. (2008). *Guide to Advanced Empirical Software Engineering*. London, Springer-Verlag.

SJØBERG, D. I., DYBÅ, T., and JØRGENSEN, M. (2007). The Future of Empirical Methods in Software Engineering Research. *In 2007 Future of Software Engineering (May 23 - 25, 2007). International Conference on Software Engineering. IEEE Computer Society, Washington, DC, 358-378.*

TRAVASSOS, G. H., GUROV, D. E AMARAL, E. A. G. (2002): Introdução à Engenharia de Software Experimental. COPPE/UFRJ, Rio de Janeiro, Relatório técnico: RT-ES-590/02, http://cultura.ufpa.br/cdesouza/teaching/methods/6-ES-Experimental.pdf. Accessed 12-Dec-2014.

WAINER, J. (2007): Métodos de pesquisa quantitativa e qualitativa para a Ciência da Computação. http://www.pucrs.br/famat/viali/mestrado/mqp/material/textos/Pesquisa.pdf. Accessed 01-Nov-2014.

WOHLIN, C., RUNESON, P., HÖST, M., OHLSSON, M. C., REGNELL, B., and WESSLÉN, A. (2012). *Experimentation in software engineering*. Berlin, Springer-Verlag Berlin Heidelberg.

ZELKOWITZ, MARVIN V., WALLACE, DOLORES R (1998). Experimental models for validating computer technology. *IEEE Computer* **31.5**: 23-31.

ANNEX

References of Primary Studies

[E1]	BIBLE, JOHN, GREGG ROTHERMEL, and DAVID S. ROSENBLUM (2001). A Comparative Study of Coarse-and Fine-Grained Safe Regression Test-Selection Techniques. *ACM Transactions on Software Engineering and Methodology* **10.2**: 149-183.
[E2]	ENGSTRÖM, EMELIE, and PER RUNESON (2010). A qualitative survey of regression testing practices. *Product-Focused Software Process Improvement*. 3-16. Berlin, Springer Berlin Heidelberg.
[E3]	DA MOTA SILVEIRA NETO, PAULO ANSELMO, et al. (2011). A systematic mapping study of software product lines testing. *Information and Software Technology* **53.5**: 407-423.
[E4]	ENGSTRÖM, EMELIE, PER RUNESON, and MATS SKOGLUND (2010). A systematic review on regression test selection techniques. *Information and Software Technology* **52.1**: 14-30.
[E5]	MARBACK, AARON, HYUNSOOK DO, and NATHAN EHRESMANN (2012). An Effective Regression Testing Approach for PHP Web Applications. Software Testing, Verification and Validation (ICST), *IEEE Fifth International Conference*. 221-230.
[E6]	SPRENKLE, SARA, et al. (2005). An empirical comparison of test suite reduction techniques for user-session-based testing of web applications. *Software Maintenance, ICSM'05. Proceedings of the 21st IEEE International Conference*.
[E7]	ENGSTRÖM, E., PER RUNESON, and GREGER WIKSTRAND (2010). An empirical evaluation of regression testing based on fix-cache recommendations. *Software Testing, Verification and Validation (ICST), Third International Conference*.
[E8]	PRAPHAMONTRIPONG, UPSORN, and JEFF OFFUTT (2010). Applying mutation testing to web applications. *Software Testing, Verification, and Validation Workshops (ICSTW), Third International Conference*.
[E9]	MESBAH, ALI, and MUKUL R. PRASAD (2011). Automated cross-browser compatibility testing. *Proceedings of the 33rd International Conference on Software Engineering. ACM*.
[E10]	DALLMEIER, VALENTIN, et al. (2012). Automatically generating test cases for specification mining. *Software Engineering, IEEE Transactions* **38.2**: 243-257.
[E11]	LEOTTA, MAURIZIO, et al. (2013). Capture-replay vs. programmable web testing: An empirical assessment during test case evolution. *Reverse Engineering (WCRE), 20th Working Conference. IEEE*.
[E12]	MARCHETTO, ALESSANDRO, et al. (2011). Crawlability metrics for automated web testing. *International journal on software tools for technology transfer* **13.2**: 131-149.
[E13]	MIRSHOKRAIE, SHABNAM, ALI MESBAH, and KARTHIK PATTABIRAMAN (2013). Efficient JavaScript mutation testing. *Software Testing, Verification and Validation (ICST), IEEE Sixth International Conference. IEEE*.
[E14]	MARCHETTO, ALESSANDRO, FILIPPO RICCA, and PAOLO TONELLA (2007). Empirical validation of a web fault taxonomy and its usage for fault seeding. *Web Site Evolution. WSE 2007. 9th IEEE International Workshop*.
[E15]	ARTZI, SHAY, et al. (2008). Finding bugs in dynamic web applications. *Proceedings of the 2008 international symposium on Software testing and analysis. ACM*.
[E16]	ENGSTROM, EMELIE, PER RUNESON, and ANDREAS LJUNG (2011). Improving Regression Testing Transparency and Efficiency with History-Based Prioritization--An Industrial Case Study. *Software Testing, Verification and Validation (ICST), 2011 IEEE Fourth International Conference. IEEE*.

[E17]	HALFOND, WILLIAM GJ, and ALESSANDRO ORSO (2007). Improving test case generation for web applications using automated interface discovery. *Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering. ACM.*
[E18]	MESBAH, ALI, ARIE VAN DEURSEN, and DANNY ROEST (2012). Invariant-based automatic testing of modern web applications. *Software Engineering, IEEE Transactions* **38.1**: 35-53.
[E19]	ELBAUM, SEBASTIAN, et al. (2005). Leveraging user-session data to support web application testing. *Software Engineering, IEEE Transactions* **31.3**: 187-202.
[E20]	OFFUTT, JEFF, and YE WU (2010). Modeling presentation layers of web applications for testing. *Software & Systems Modeling* **9.2**: 257-280.
[E21]	LI, NUO, et al. (2010). Perturbation-based user-input-validation testing of web applications. *Journal of Systems and Software* **83.11**: 2263-2274.
[E22]	SAMPATH, SREEDEVI, et al. (2008). Prioritizing user-session-based test cases for web applications testing. *Software Testing, Verification, and Validation, 2008 1st International Conference. IEEE.*
[E23]	ROEST, DANNY, ALI MESBAH, and ARIE VAN DEURSEN (2010). Regression testing ajax applications: Coping with dynamism. *Software Testing, Verification and Validation (ICST), 2010 Third International Conference. IEEE.*
[E24]	AFZAL, WASIF, RICHARD TORKAR, and ROBERT FELDT (2009). A systematic review of search-based testing for non-functional system properties. *Information and Software Technology* **51.6**: 957-976.
[E25]	MAEZAWA, YUTA, et al. (2014). Validating ajax applications using a delay-based mutation technique. *Proceedings of the 29th ACM/IEEE international conference on Automated software engineering. ACM.*
[E26]	COLOMBO, CHRISTIAN, MARK MICALLEF, and MARK SCERRI (2014). Verifying Web Applications: From Business Level Specifications to Automated Model-Based Testing. *arXiv preprint arXiv*1403.7258.
[E27]	SAMPATH, SREEDEVI, et al. (2006). Web application testing with customized test requirements-An experimental comparison study. *Software Reliability Engineering, 2006. ISSRE'06. 17th International Symposium. IEEE.*
[E28]	DOĞAN, SERDAR, AYSU BETIN-CAN, and VAHID GAROUSI (2014). Web application testing: A systematic literature review. *Journal of Systems and Software* **91**: 174-201.
[E29]	DALLMEIER, VALENTIN, et al. (2013). WebMate: Generating test cases for web 2.0. *SWQD* **133**: 55-69.
[E30]	TONELLA, PAOLO, and FILIPPO RICCA (2004). A 2-layer model for the white-box testing of web applications. *Telecommunications Energy Conference, 2004. INTELEC 2004. 26th Annual International. IEEE.*
[E31]	BARADHI, GHINWA, and NASHAT MANSOUR (1997). A comparative study of five regression testing algorithms. *Software Engineering Conference, 1997. Proceedings. Australian. IEEE.*
[E32]	ZHENG, JIANG, et al. (2006). A lightweight process for change identification and regression test selection in using cots components. *Commercial-off-the-Shelf (COTS)-Based Software Systems, 2006. Fifth International Conference. IEEE.*
[E33]	SAXENA, PRATEEK, et al. (2010). A symbolic execution framework for javascript. *Security and Privacy (SP), 2010 IEEE Symposium. IEEE.*
[E34]	DOBOLYI, KINGA (2010). An Exploration of User-Visible Errors in Web-based Applications to Improve Web-based Applications. Dissertation. University of Virginia.

[E35]	ZHENG, JIANG, et al. (2005). An initial study of a lightweight process for change identification and regression test selection when source code is not available. *Software Reliability Engineering, 2005. ISSRE 2005. 16th IEEE International Symposium. IEEE.*
[E36]	ZHENG, JIANG, et al. (2006). Applying regression test selection for COTS-based applications. *Proceedings of the 28th international conference on Software engineering. ACM.*
[E37]	HALFOND, WILLIAM GJ, and ALESSANDRO ORSO (2008). Automated identification of parameter mismatches in web applications. *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering. ACM.*
[E38]	RAN, LIHUA, et al. (2009). Building test cases and oracles to automate the testing of web database applications. *Information and Software Technology* **51.2**: 460-477.
[E39]	OFFUTT, JEFF, et al. (2004). Bypass testing of web applications. *Software Reliability Engineering, 2004. ISSRE 2004. 15th International Symposium. IEEE.*
[E40]	YANG, JI-TZAY, et al. (2002). Constructing and Object-Oriented Architecture for Web Application Testing. *Journal of Computing and Information Science in Engineering* **18.1**: 59-84.
[E41]	CHOUDHARY, SHAUVIK ROY, MUKUL R. PRASAD, and ALESSANDRO ORSO (*2012*). Crosscheck: Combining crawling and differencing to better detect cross-browser incompatibilities in web applications. *Software Testing, Verification and Validation (ICST), 2012 IEEE Fifth International Conference. IEEE.*
[E42]	HALLÉ, SYLVAIN, et al. (2010). Eliminating navigation errors in web applications via model checking and runtime enforcement of navigation state machines. *Proceedings of the IEEE/ACM international conference on Automated software engineering. ACM.*
[E43]	BAJWA, HARPREET, WENLIANG XIONG, and FRANK MAURER (2005). Evaluating current testing processes of web-portal applications. *Web Engineering.* 603-605. Springer Berlin Heidelberg.
[E44]	RAFFELT, HARALD, et al. (2008). Hybrid test of web applications with webtest. *Proceedings of the 2008 workshop on Testing, analysis, and verification of web services and applications. ACM.*
[E45]	AGRAWAL, H., HORGAN, J. R., KRAUSER, E. W. and LONDON, S. (1993). Incremental Regression Testing. *ICSM* **93**: 348-357.
[E46]	MIRSHOKRAIE, SHABNAM, and ALI MESBAH (2012). JSART: JavaScript assertion-based regression testing. *Web Engineering.* 238-252. Springer Berlin Heidelberg.
[E47]	KUNG, DAVID C., et a.l (1996). On regression testing of object-oriented programs. *Journal of Systems and Software* **32.1**: 21-40.
[E48]	SAKAMOTO, KAZUNORI, et al. (2013). POGen: a test code generator based on template variable coverage in gray-box integration testing for web applications. *Fundamental Approaches to Software Engineering.* 343-358. Berlin, Springer Berlin Heidelberg.
[E49]	HALFOND, WILLIAM GJ, SASWAT ANAND, and ALESSANDRO ORSO (2009). Precise interface identification to improve testing and analysis of web applications. *Proceedings of the eighteenth international symposium on Software testing and analysis. ACM.*
[E50]	BRYCE, RENÉE C., and CHARLES J. COLBOURN (2006). Prioritized interaction testing for pair-wise coverage with seeding and constraints. *Information and Software Technology* **48.10**: 960-970.
[E51]	CHITTIMALLI, PAVAN KUMAR, and MARY JEAN HARROLD (2009). Recomputing coverage information to assist regression testing. *Software Engineering, IEEE Transactions* **35.4**: 452-469.
[E52]	SOUZA, SIMONE RS, et al. (2011). Research in concurrent software testing: A systematic review. *Proceedings of the Workshop on Parallel and Distributed Systems: Testing, Analysis, and Debugging. ACM.*

[E53]	ANDREWS, ANNELIESE A., et al. (2010). Scalability issues with using FSMWeb to test web applications. *Information and Software Technology* **52.1**: 52-66.
[E54]	ZHENG, YUNHUI, TAO BAO, and XIANGYU ZHANG (2011). Statically locating web application bugs caused by asynchronous calls. *Proceedings of the 20th international conference on World wide web. ACM.*
[E55]	PALACIOS, MARCOS, JOSÉ GARCÍA-FANJUL, and JAVIER TUYA (2011). Testing in Service Oriented Architectures with dynamic binding: A mapping study. *Information and Software Technology* **53.3**: 171-189.
[E56]	LI, YUAN-FANG, PARAMJIT K. DAS, and DAVID L. DOWE (2014). Two decades of Web application testing - A survey of recent advances. *Information Systems* **43**: 20-54.
[E57]	PENG, XUAN, AND LU LU (2011). User-session-based automatic test case generation using GA. *International Journal of Physical Sciences* **6.13**: 3232-3245.
[E58]	CHOUDHARY, SHAUVIK ROY, HUSAYN VERSEE, and ALESSANDRO ORSO (2010). WEBDIFF: Automated identification of cross-browser issues in web applications. *Software Maintenance (ICSM), 2010 IEEE International Conference. IEEE.*
[E59]	SPRENKLE, SARA, et al. (2008). WebVizOr: a visualization tool for applying automated oracles and analyzing test results of web applications. *Practice and Research Techniques, 2008. TAIC PART'08. Testing: Academic & Industrial Conference. IEEE.*
[E60]	JAVED, ABU (2014). A Model-Driven Framework for Context-Dependent Component Testing. PhD Thesis. School of Information Technology and Electrical Engineering. The University of Queensland.
[E61]	AMANNEJAD, YASAMAN, et al. (2014). A search-based approach for cost-effective software test automation decision support and an industrial case study. *Software Testing, Verification and Validation Workshops (ICSTW), 2014 IEEE Seventh International Conference. IEEE.*
[E62]	JAIN, RITU, and UGRASEN SUMAN (2015). A Systematic Literature Review on Global Software Development Life Cycle. *ACM SIGSOFT Software Engineering Notes* **40.2**: 1-14.
[E63]	ORTAC, ALPER, MARTIN MONPERRUS, and MIRA MEZINI (2015). Abmash: mashing up legacy Web applications by automated imitation of human actions. *Software: Practice and Experience* **45.5**: 581-612.
[E64]	DEVROEY, XAVIER, GILLES PERROUIN, and PIERRE-YVES SCHOBBENS (2014). Abstract test case generation for behavioural testing of software product lines. *Proceedings of the 18th International Software Product Line Conference: Companion Volume for Workshops, Demonstrations and Tools-Volume 2. ACM.*
[E65]	RODRIGUES, E. M., DE OLIVEIRA, F. M., COSTA, L. T., BERNARDINO, M., ZORZO, A. F., SOUZA, S. D. R. S., and SAAD, R. (2014). An empirical comparison of model-based and capture and replay approaches for performance testing. *Empirical Software Engineering* **20.6**: 1831-1860.
[E66]	HOQUE, MOHAMMAD IFTEKHARUL, et al. (2014). An Empirical Study on Refactoring Activity. *arXiv preprint arXiv*:1412.6359.
[E67]	ABURAS, ALI, and ALEX GROCE (2014). An improved memetic algorithm with method dependence relations (MAMDR). *Quality Software (QSIC), 2014 14th International Conference. IEEE.*
[E68]	NGUYEN, BAO N., and ATIF M. MEMON (2014). An observe-model-exercise* paradigm to test event-driven systems with undetermined input spaces. *Software Engineering, IEEE Transactions* **40.3**: 216-234.

[E69]	TAPPENDEN, ANDREW F., and JAMES MILLER (2014). Automated cookie collection testing. *ACM Transactions on Software Engineering and Methodology (TOSEM)* **23.1**: 3.
[E70]	GARG, DEEPAK (2014). Automated Techniques to Detect Faults Early in Large Software Applications. PhD thesis. The University of Western Australia.
[E71]	SHAHBAZ, MUZAMMIL, PHIL MCMINN, and MARK STEVENSON (2015). Automatic generation of valid and invalid test data for string validation routines using web searches and regular expressions. *Science of Computer Programming* **97**: 405-425.
[E72]	MARIANI, LEONARDO, et al. (2014). Automatic testing of GUI based applications. *Software Testing, Verification and Reliability* **24.5**: 341-366.
[E73]	SHEN, DU, et al. (2015). Automating performance bottleneck detection using search-based application profiling. *Proceedings of the 2015 International Symposium on Software Testing and Analysis. ACM.*
[E74]	MARAS, JOSIP (2014). Automating Reuse in Web Application Development. PhD thesis. School of Innovation, Design and Engineering. Mälardalen University. Sweden.
[E75]	ZHI, JUNJI, et al. (2015). Cost, benefits and quality of software development documentation: A systematic mapping. *Journal of Systems and Software* **99**: 175-198.
[E76]	ALSHAHWAN, NADIA, and MARK HARMAN (2014). Coverage and fault detection of the output-uniqueness test selection criteria. *Proceedings of the 2014 International Symposium on Software Testing and Analysis. ACM.*
[E77]	MAHAJAN, SONAL, and WILLIAM GJ HALFOND (2015). Detection and localization of html presentation failures using computer vision-based techniques. *Software Testing, Verification and Validation (ICST), 2015 IEEE 8th International Conference. IEEE.*
[E78]	MIRZAAGHAEI, MEHDI, AND ALI MESBAH (2014). DOM-based test adequacy criteria for web applications. *Proceedings of the 2014 International Symposium on Software Testing and Analysis. ACM.*
[E79]	CHAUDHARY, NEHA, O. P. SANGWAN, and RICHA ARORA (2014). Event-Coverage and Weight based Method for Test Suite Prioritization. *International Journal of Information Technology and Computer Science (IJITCS)* **6.12**: 61.
[E80]	MIRSHOKRAIE, SHABNAM, ALI MESBAH, and KARTHIK PATTABIRAMAN (2015). Guided Mutation Testing for JavaScript Web Applications. *Software Engineering, IEEE Transactions* **41.5** 429-444.
[E81]	JU, XIAOLIN, et al. (2014). HSFal: Effective fault localization using hybrid spectrum of full slices and execution slices. *Journal of Systems and Software* **90**: 3-17.
[E82]	TONELLA, PAOLO, ROBERTO TIELLA, and CU DUY NGUYEN (2014). Interpolated n-grams for model based testing. *Proceedings of the 36th International Conference on Software Engineering. ACM.*
[E83]	MIRSHOKRAIE, SHABNAM, ALI MESBAH, and KARTHIK PATTABIRAMAN (2015). JSeft: automated JavaScript unit test generation. *Software Testing, Verification and Validation (ICST), 2015 IEEE 8th International Conference. IEEE.*
[E84]	MILANI FARD, AMIN, MEHDI MIRZAAGHAEI, and ALI MESBAH (2014). Leveraging existing tests in automated test generation for web applications. *Proceedings of the 29th ACM/IEEE international conference on Automated software engineering. ACM.*
[E85]	FLOSS, BRIANNA (2014). Overcoming Hard Problems in Software Testing with Testing as a Service. Dissertation. Florida Institute of Technology.

[E86]	ZAREI, ALIREZA (2014). Performance improvements in crawling modern Web applications. Dissertation. The Faculty of Graduate and Postdoctoral Studies. The University of British Columbia.
[E87]	CHRISTOPHE, LAURENT, et al. (2014). Prevalence and maintenance of automated functional tests for web applications. *Software Maintenance and Evolution (ICSME), 2014 IEEE International Conference. IEEE.*
[E88]	SHAMSHIRI, SINA, et al. (2015). Random or Genetic Algorithm Search for Object-Oriented Test Suite Generation? *Proceedings of the 2015 on Genetic and Evolutionary Computation Conference. ACM.*
[E89]	DENG, DONGCHENG (2014). Reliable Embedded Systems Development. Dissertation. University of Calgary.
[E90]	MADHAVAN, ARVIND (2014). Semi Automated User Acceptance Testing using Natural Language Techniques. Dissertation. Iowa State University.
[E91]	KRUSE, PETER M., JIRKA NASAREK, and N. CONDORI FERNANDEZ (2014). Systematic Testing of Web Applications with the Classification Tree Method. *XVII Iberoamerican Conference on Software Engineering (CIbSE 2014).*
[E92]	MAYO, QUENTIN, RYAN MICHAELS, and RENEE BRYCE (2014). Test Suite Reduction by Combinatorial-Based Coverage of Event Sequences. *Software Testing, Verification and Validation Workshops (ICSTW), 2014 IEEE Seventh International Conference. IEEE.*
[E93]	SEQUEIRA, SHELDON (2014). Understanding web application test assertion failures. Dissertation. The Faculty of Graduate and Postdoctoral Studies. The University of British Columbia.
[E94]	GHIDUK, AHMED S (2014). Using Evolutionary Algorithms for Higher-Order Mutation Testing. *International Journal of Computer Science Issues (IJCSI)* **11.2**: 93.
[E95]	OCARIZA JR, FROLIN S., KARTHIK PATTABIRAMAN, and ALI MESBAH (2014). Vejovis: Suggesting fixes for JavaScript faults. *Proceedings of the 36th International Conference on Software Engineering. ACM.*

ERRATUM

In the page 1: Authors and affiliation:

where it reads:

Arnaldo Gil Sardella Nascimento[1] http://orcid.org/0000-0002-3534-8141
[1]Federal University of Rio de Janeiro, Rio de Janeiro, RJ, Brazil.

should read:

Arnaldo Gil Sardella Nascimento[1] http://orcid.org/0000-0002-3534-8141
Eber Schmitz[1] https://orcid.org/0000-0002-4839-4606
Monica Silva[1] http://orcid.org/0000-0003-0951-6612
Priscila Lima[1] https://orcid.org/0000-0002-8515-9904
[1]Federal University of Rio de Janeiro, Rio de Janeiro, RJ, Brazil.