International Journal of Combinatorial Optimization Problems and Informatics, 13(4), Dec 2022, 47-57. ISSN: 2007-1558.

www.editada.org

A Systematic Mapping Study in the area of Software Testing

Raúl A. Aguilar[1], Naomi Garcia[1], Julio C. Díaz[1]*

[1] Universidad Autónoma de Yucatán, Facultad de Matemáticas. Mérida, Yucatán, México

*Corresponding author (avera@correo.uady.mx)

Abstract. This article offers an overview of the research developed in the last decade in the area of software testing. Systematic Literature Mapping was used as a research methodology, and based on three databases, 171 primary studies were selected. The study raises a set of research questions about techniques, strategies and types of tests, aspects such as quality factors analyzed, degree of innovation, and even about the empirical method used in the study. Among the findings, it was found that performance tests are the most analyzed types of tests, mixed strategies were the most cited, that efficiency turned out to be the most important quality factor, and that case studies are the most used technique by researchers. The secondary study carried out allowed us to conclude that research in the field of software testing is still valid in the context of Software Engineering, so it is pertinent to continue research on specific topics through both primary and secondary studies.

Keywords: Software Quality, Software Testing, Validation, Verification.

Article Info
Received March 25, 2022
Accepted August 12, 2022

1 Introduction

For at least a couple of decades, the Software Engineering Body of Knowledge (SWEBOK) integrated the main topics related to software development and management processes, which were documented in 2004 [1] and updated in 2014 [2]. However, finding a software development process that satisfies the real needs of the client and thereby generating better quality software, has been and continues to be one of the objectives of SE from the moment it was conceived as a discipline, and in this sense , questions such as: (1) How to know if the built product works correctly? and (2) How to know if the product corresponds to what the client needed? They have tried to be served with empirical research related to software testing, particularly in topics such as software verification and validation [3].

One of the development areas directly linked to the software quality improvement process is Software Testing. In order to explore areas of opportunity to continue research in the field of Software Testing, the authors proposed the development of a secondary study —a systematic mapping of the literature— to identify and classify the main characteristics of the research carried out in the area of Software testing. It was agreed to analyze primary studies published between 2010 and 2021, with the intention of updating, to a certain extent, the state of the art structured in the SWEBOK [1, 2], but above all, to evaluate the relevance of continuing the

investigation with a study primary, or even secondary, on a particular topic—a systematic review of the literature.

The following section describes the main concepts on which the secondary study is based; the third section describes the methodology used to carry out the reported study; in the fourth section the tasks developed for the planning of the study are presented; the fifth section cites the tasks performed to obtain the 171 primary studies finally selected; the sixth section answers the research questions that guide the study; the seventh section discusses aspects related to threats to the validity of the study. Finally, a synthesis of the information analyzed is presented, as well as a reflection on future work in the context of software testing.

2 Background

According to [4], the software testing process aims to reveal defects in the software, and in parallel, measure the degree of quality that it possesses with respect to a set of selected attributes. A first position on software testing establishes that process consists of the dynamic verification of the expected behaviors of the software, based on a set of selected cases [2], and with this, the non-adaptation of said behaviors would generate failures in the software system. Another somewhat more flexible position considers that there are many ways to evaluate software, or in its case, to test a system without having to run it [5]; for example, reviews are a type of testing technique that can be used to verify the quality of a software artifact, such as a requirements specification document or code listing, this technique allows to identify faults in said artifacts. Both positions, the dynamic analysis and the static one, represent mechanisms that allow us to answer the first question mentioned above. For a long time, it was stated that testing software was an activity that should occur at the end of the development process; however, this perspective has changed in recent decades, since it has been shown that testing in the early stages of development avoid finding defects in the software at the time of its delivery, thus allowing the development of better-quality software. Regarding the second question, which necessarily involves the client, has a more subjective character, and is linked to the software validation process, said process is defined as the one that is responsible for evaluating the software during or at the end of the process. of the development cycle, to determine if it satisfies the requirements specified and agreed with the client [6].

On the topic of testing strategies, two alternatives are recognized: (1) based on the knowledge of the internal functioning of the software, tests can be designed and executed to ensure that the internal operations are carried out according to the specifications and that all the internal components have been properly exercised; and (2) According to the specification for which the software has been designed, test cases can be designed and executed to demonstrate that each function is fully operational, while it is possible to identify flaws in it. Under the first approach, the tests require an internal view of the software and are known as white box tests; while with the second, the tests have an external view and are known as black box tests [7]. On the other hand, regarding the test level, it is possible to distinguish four levels depending on the test object: (1) unit tests, (2) integration tests, (3) system tests and some type of (4) acceptance tests [8]. It is important to mention that both black box and white box tests can be classified based on what they evaluate, thus dividing them into functional and non-functional tests. On the one hand, we have the functional tests that have been mentioned above (unit, integration, system and acceptance) and, on the other hand, we have the non-functional tests that evaluate certain types of behavior in the software such as compatibility, security, stress, usability performance, among others. As mentioned above, the software testing process is carried out with the aim of ensuring the quality of the final product, and that is why this activity considers a set of quality factors [9] that must be evaluated, such as usability, maintainability, security, portability, efficiency, testability, flexibility, reliability, robustness, to name a few. The evaluation of these factors aims to verify that the software to be delivered not only complies with the agreed functional part, but also with those non-functional requirements agreed with the clients.

With the purpose of analyzing previous secondary studies, in particular, mapping-type studies, the authors found that these studies have been carried out with an orientation to a particular topic or type of software, for example in [10] the authors analyze knowledge management for the testing process, in [11] they are oriented to mobile applications, in [12] the authors analyze their applicability for software in the cloud, in [13] for the agent-oriented paradigm, and even in [14] The authors synthesize primary studies in the context of teaching programming. This study has a more general purpose, in order to identify new opportunity niches.

3 Research Method

In order to collect the results of empirical studies published over the last decade in the area of software testing, and thereby characterize the research carried out under a quantitative approach, it was decided to carry out a Systematic Mapping of Literature, also known as Mapping Study (MS); This type of secondary study aims to explore and provide a global vision of a research area, which allows identifying opportunity niches for primary studies, or where appropriate, more specific secondary studies, such as a systematic literature review; according to [15] in MS it is common to classify the findings obtained according to some predefined classification scheme. For the development of this study, the guide proposed in [16] was used as a reference, in which the following tasks are established:

1. *Formulation of research questions:* The main objective of the EM is to provide an overview of a research area and to identify the amount, type of research and results available within it.
2. *Search for primary studies:* Primary studies are identified by using search strings in scientific Databases or by manually browsing through conference proceedings or relevant journal publications.
3. *Selection of relevant articles:* A set of inclusion and exclusion criteria are applied to determine the eligibility of the primary articles to be analyzed.
4. *Definition of a classification scheme:* Researchers review the abstracts and look for keywords and concepts that allow a classification scheme to be identified.
5. *Extraction of data and preparation of the report:* When the classification scheme is in place, the relevant articles are classified in the scheme, that is, the extraction of real data is carried out and the results are analyzed, presenting the frequencies of the publications for each category, this makes it possible to see which categories have been emphasized in previous research and, therefore, identify gaps and possibilities for future research.

4 Study Planning

The review of the state of the art embodied in the SWEBOK, the analysis of some materials considered as obligatory references for the area of software testing [4], [5], [9], as well as the reading of previous secondary studies [10], [11], [12]. [13], [14], allowed the accumulation of sufficient information for the formulation of the questions of research.

4.1 Research questions

The research questions that guided the study are:

- ***PI01.*** What is the distribution of the primary studies published on software testing from 2010 to 2021?
- ***PI02.*** What are the levels of software testing addressed by primary studies from 2010 to 2021?
- ***PI03.*** What type of software tests have been the most reported from 2010 to 2021?
- ***PI04***. What are the quality factors that have been evaluated in the software from 2010 to 2021?
- ***PI05.*** What are the characteristics of the strategies used in software tests reported from 2010 to 2021?
- ***PI06.*** What have been the validation methods used from 2010 to 2021 by primary studies related to the area of software testing?
- ***PI07.*** In what context have the primary studies related to the software testing area been developed from 2010 to 2021?

The review of the state of the art and the formulation of the research questions served as the basis for carrying out a PICOC analysis [17] of the area under study, this analysis allowed the identification of a set of key aspects to assist in the subsequent analysis of the studies selected.

- ***P***opulation: Software
- ***I***ntervention: Software testing techniques or strategies.
- ***C***omparison: Software Quality.
- ***O***utput: Improved performance and software performance.
- ***C***ontext: Industry or Academy.

With the PICOC analysis and the formulated research questions, possible categories were identified that would allow the researcher to classify the information resulting from the analysis of the selected primary studies (see Table 1).

Table 1. Categories identifies for the analysis of the selected primary studies.

Dataset	*N*
PI02. Test level	Unit, Integration, System, Acceptance
PI03. Test strategy	Black Box, White Box, Mixed
PI05. Innovation	New, Improvement, Existing.
PI06. Validation mechanism.	Case Study, Experiment, Comparative Analysis
PI07. Context	Industry, Academy

4.2 Search strategy

According to the objective of the proposed research, two databases and a repository to access specialized bibliography were selected for the Systematic Mapping:

- ***ACM Digital Library:*** is a collection of full-text articles and bibliographic records on computing, computer science, and related areas. The full-text database includes the entire collection of Association for Computing Machinery's (ACM) publications.

- ***IEEE Xplore:*** an academic research DB in the areas of Computer Science, Electrical and Electronic Engineering. Although the articles in this DB are restricted access, the summaries are freely accessible.

- ***Google Scholar:*** A database of scholarly articles (from a wide range of sources, but primarily journal articles, conference proceedings, white papers, and dissertations) with no language, journal, or geographic restrictions, allowing access to literature not available in other Databases.

Although these databases do not ensure the availability of the texts of the articles, in the case of our study, access to their abstracts is sufficient.

Based on the results obtained with the PICOC methodology, the following generic search string was developed to perform the search:

("software testing" AND "software development") AND "testing software"

The chain was configured based on the repositories considered; the resulting search strings allowed the ACM DL, IEEE Xplore and Google Scholar to be crawled to more accurately obtain information from the last decade regarding software testing. These strings were:

- ***ACM DL:*** [Title: "software testing"] AND [[Abstract: types] OR [Abstract: techniques] OR [Abstract: strategies]] AND [Publication Date: (01/01/2010 TO 12/31/2021)]

- ***IEEE Xplore:*** *("Document Title":Software Testing) AND (("Abstract":Techniques) OR ("Abstract":Types) OR ("Abstract":Strategies)).*

- ***Google Scholar:*** *allintitle: Software AND Testing AND (Types OR Techniques OR Strategies).*

4.3 Inclusion/exclusion criteria for study selection

In order to filter the search and reduce the results to only those documents relevant to the proposed research object, which would allow answering the research questions, some inclusion and exclusion criteria were defined and used; These criteria allowed us to obtain a better organization of the results obtained in the searches carried out in this study.
Inclusion criteria:

- Time window between 2010 and 2022
- Primary studies published in specialized journals.
- Articles with titles and abstracts in English.

Exclusion criteria:

- Dissemination articles.
- Duplicate articles.

5 Execution of the Study

The reported study was carried out during the month of February 2022, for the study the results retrieved from the searches in the ACM-DL, IEEE Xplore and Google Scholar repositories were filtered with the purpose of excluding those results that were not aligned with the objectives of the systematic mapping, according to the three phases defined for the study:

- Phase 1: Application of the search string in the selected databases.
- Phase 2: Application of the inclusion criteria to the set of studies obtained in phase 1.
- Phase 3: Application of the exclusion criteria to the set of studies obtained in phase 2.

Table 2 presents the number of studies obtained at the end of each phase of the selection process.

Table 2. Selection process of primary studies.

Dataset	Phase 1	Phase 2	Phase 3
ACM Digital Library	151	108	42
IEEE Xplore	101	48	41
Google Scholar	306	165	88
		Total	***171***

During the aforementioned phases, 69.35% of the set of studies from the three databases were discarded because they did not meet the established inclusion and exclusion criteria. It is worth mentioning that a considerable number of discarded articles were related to software testing education, particularly on strategies and techniques to teach software testing in a way that allows students to understand its importance and to know what tools currently exist. Thus, a total of 171 reported primary studies were obtained that are directly related to software testing research from 2010 to 2021. Information was extracted from each selected study based on the predefined classification scheme, carefully analyzing the abstract of each of the selected articles, the above, by virtue of the purpose and time available for the study.

6 Results of the Study

This section presents the results obtained from the quantitative analysis and synthesis of the 171 primary studies selected from the three databases: 42 from ACM Digital Library, 41 from IEEE Xplore and 88 from Google Scholar. It is important to mention that it was observed that a good part of the abstracts of the selected studies did not provide information in such a detailed way as to classify them in one of the identified categories, therefore, in several predefined classifications, the category of "Not specify" was included. The template used with the data retrieved from the primary studies selected, can be obtained from the following link: [23].

PI01. What is the distribution of the primary studies published on software testing from 2010 to 2021?

Figure 1 illustrates the number of publications per year, on average 14 articles are reported from 2010 to 2021.

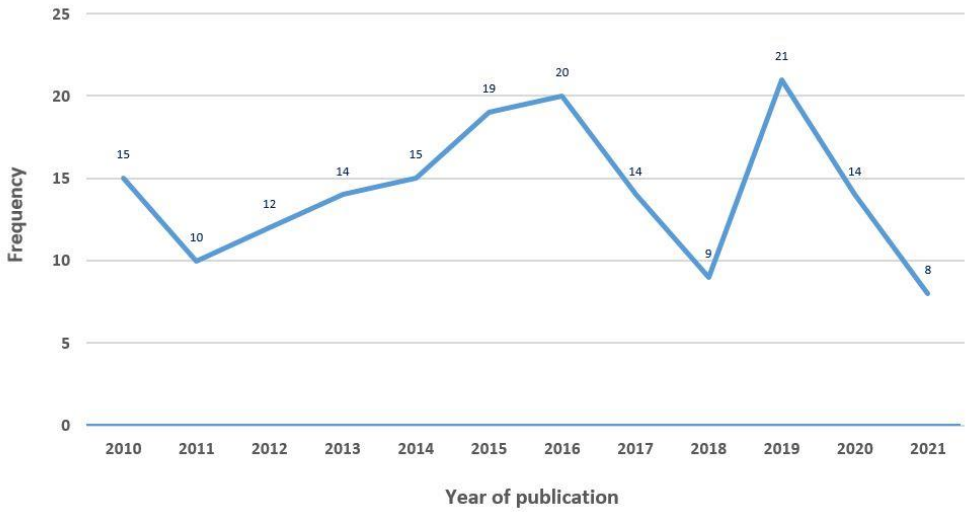

Figure 1. Frequency of selected studies by year of publication

As can be seen, from 2011 to 2016 there was a growing trend of publications; however, in the second half of the analyzed period there is no homogeneous behavior, and peaks of 20 and 21 articles are observed in 2016 and 2019 respectively, and with the lowest number of articles in 2018 and 2021, of 9 and 8 publications respectively.

PI02. What are the levels of software testing addressed by primary studies from 2010 to 2021?

According to the categories considered, it is observed (see figure 2) that the type of tests with the highest mention in the studies was System tests with 71 mentions (41.52%), remaining with 13 (7.60%), 8 (4.68%) and 4 (2.34%), the integration, unit and acceptance tests, respectively. It is important to highlight a high number of analyzed studies (75) were placed in the "Not Specified" category, since the level of evidence was not indicated in the abstract of the article.

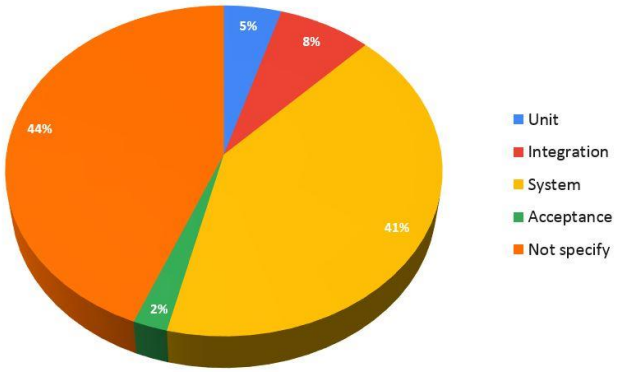

Figure 2. Percentage of selected studies according to the level of test

PI03. What type of software tests have been the most reported from 2010 to 2021?

As can be seen in table 3, most of the articles reported were classified as software tests that are oriented to its

performance (52.63%), in second place we find automated tests (15.79%), which are mostly improvements or new proposals, and there are various types of tests that are mentioned in smaller numbers. It should be noted that, as with the level of evidence, there is a subset of selected studies (16,37%), in which the abstract does not specify the type of evidence addressed.

Table 3. Frequency of studies according to the type of software test reported

Type of test	*f*
Performance	90
Automation	27
Security	6
Usability	5
Regression	5
Stress	3
Maintainability	3
Scalability	2
Concurrence	1
Adaptative	1
Not specify	28

PI04. What are the quality factors that have been evaluated in the software from 2010 to 2021?

As shown in figure 3, efficiency and test capacity are the most cited quality factors in the research analyzed, reliability and maintainability turn out to be the second group of factors cited, and in a third group, although in less quantity, we find security and usability. It should be noted that research on aspects such as quality factors in software testing, denotes the concern that still exists regarding the aspects of software that are most affected in the software industry. As we have been reporting in the previous two questions, there is a subset of selected studies (14.62%), in which the abstract does not specify the quality factor addressed.

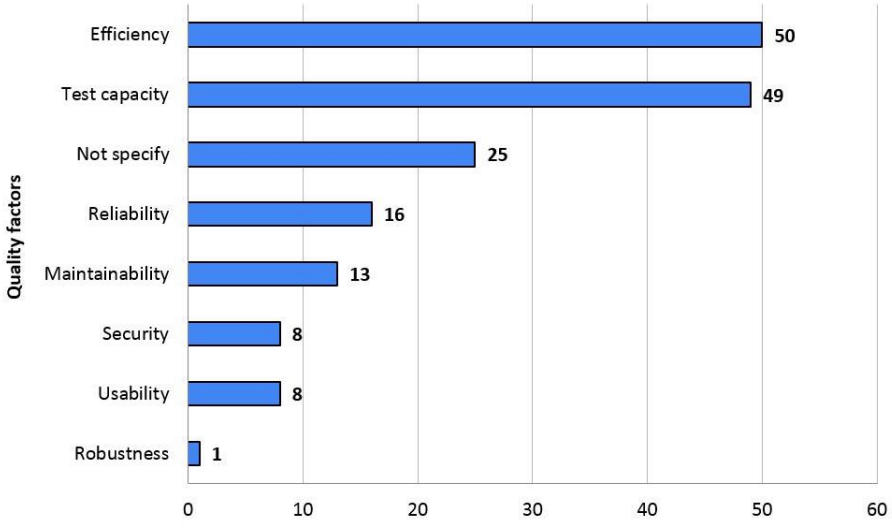

Figure 3. Frequency of selected studies according to the quality factor addressed.

PI05. What are the characteristics of the strategies used in software tests reported from 2010 to 2021?

The analysis of the secondary studies allowed us to identify (see figure 4) that a mixed approach is the most

used with 40 studies (23.39%), in second place is the black box strategy with 22 studies (12.86%) and in third place black box tests with 17 studies (9.94%). It should be noted that ninety-two articles failed to be classified, because either the type of test strategy selected was not explicitly mentioned, or, in its case, the strategies were not an aspect of interest to be investigated

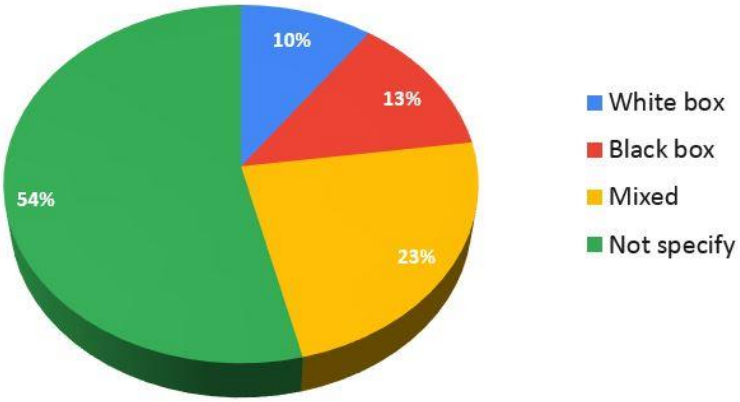

Figure 4. Percentage of studies classified according to testing strategy

In relation to the level of innovation in the study proposal, it was possible to account those 104 (60.82%) studies focused on already existing techniques or methodologies, 41 (23.98%) were oriented towards improvements and only 25 (14.62%) presented innovative proposals in the testing area. It should be noted that one of the articles failed to be classified; it was also possible to identify that the articles with novel proposals began to become more constant as of 2015.

PI06. What have been the validation methods used from 2010 to 2021 by primary studies related to the area of software testing?

For researchers, it is of particular importance to identify the empirical methodology used in the selected studies, as a mechanism for validating proposals in the field of Software Engineering. From the analysis of the one hundred and seventy-one selected studies, the two validation methods used could be counted: 121 case studies (70.76%) and 43 experiments (25.15%), two of the most cited empirical research methods in SE [18]. Six articles were also found that presented a comparative analysis, which is a research method that allows, as its name implies, to contrast one technique or methodology with another; additionally, one study could not be classified. Figure 5 illustrates what is reported here.

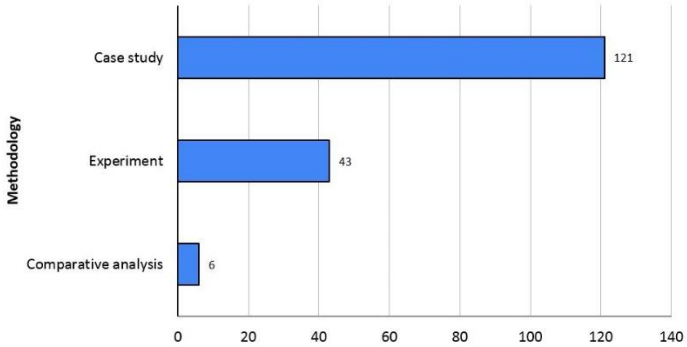

Figure 5. Frequency of selected studies according to the methodology used

PI07. In what context have the primary studies related to the software testing area been developed from 2010 to 2021?

A second aspect of particular interest to researchers is related to the context in which Software Engineering research is carried out. The present study allowed us to identify that 126 (73.63%) of the selected studies —in the field of software testing— are developed in the industrial field; however, 27 (15.79%) could not be identified and the remaining 18 (10.53%) reported the academic environment as context.

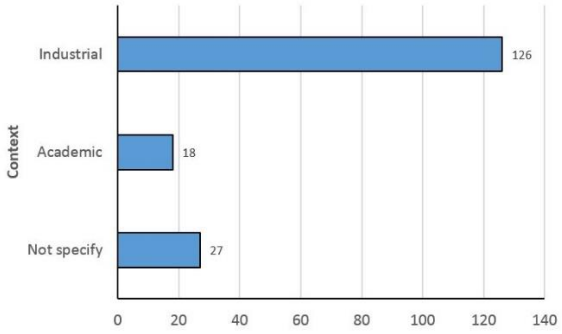

Figure 6. Frequency of selected studies according to the context

7 Threats to Internal and External Validity

According to [19] the validity of a study —primary— refers to the trustworthiness of its results and to what extent the results are true and not biased by the point of view of the researchers. For the present study, we will use the interpretation developed in [20] around threats to internal and external validity, in the context of secondary studies.

7.1 Internal validity

Considering that the Internal Validity must be analyzed around the conduct of the study, particularly with the extraction and synthesis of data, in the sense of the possibility that there are factors that could bias the general process. The main threat to internal validity is due to the fact that the data extraction process is focused on the review, mainly of the abstracts of the selected primary studies; As reflected in [16], abstracts of papers in the context of software engineering are often misleading and lack important information, and therefore the use of structured abstracts is recommended, since they considerably improve the comprehensibility of the papers. In our study, just under 6% of the studies contained structured abstracts, which is reflected in the fact that an important set of studies could not be classified in the defined categories for most of the research questions. However, in general we believe that the internal validity of this study is high given the use of a systematic procedure and discussion between three researchers, as well as the review of additional sections of the abstract, in cases where the content was available.

7.2 External validity

External Validity corresponds to the evaluation of the range covered by the selected primary studies, in terms of their settings, materials and participants. The main threat to external validity is due to the limited number of bibliographic sources consulted (ACM-DL, IEEE Xplore and Google Scholar) as well as the quality of the search engines, which could have influenced the completeness of the selected primary studies. On the other hand, being a mapping study, the protocol does not consider an evaluation of the quality of the selected studies. However, we consider that the study maintains an adequate external validity, because the two databases consulted are the ones that have the greatest recognition in the context of Software Engineering, and the combination with a repository allowed to access to literature not available in other Databases. On the other hand, the research protocol —reviewed by three researchers— was put to the test in a previous study [21], so we consider that it is clear enough to reproduce it.

8 Conclusions and Directions for Further Research

Mapping Studies are studies with a broad scope, since their main objective is to present a global vision on a topic of interest and identify the amount and type of research and results available on it. The secondary study carried out in the field of Software Engineering, in particular in the area of Software Testing, allowed us to know that research in this area has remained constant, although with ups and downs in the last half decade.
In relation to the level of the test, from the analysis of the selected studies it was possible to identify that there is a high number of studies (41%) that are oriented towards system tests, particularly performance and automation tests. Regarding the quality factors analyzed, efficiency and capacity are the most used aspects in 29% of the reported research. It is also interesting to report that mixed strategies (23%) have gained relevance in recent years, compared to traditional black box or white box strategies. In relation to the rigor of the research, two aspects of singular importance for the researchers were knowing the methodologies commonly used in the context of Software Engineering research, as well as the context of the studies reported in the last decade; it was possible to confirm that the case study (71%) and experimentation (25%) are still the most used empirical methods, and that a third of the studies have been developed in industrial contexts.

Finally, with the studies carried out, the authors consider that in the field of software testing, there are problems that can be addressed for the first time in the near future, which is why it is an area of opportunity to continue with a primary study, such as study proposed in [22] around evaluating the maturity of test practices; or in its case, with a secondary study of greater specificity, such as a systematic review of the literature on the characteristics of software tests that incorporate Artificial Intelligence techniques, or in its case, the pedagogical aspect in the subject of software tests in the training of human talent in software engineering.

References

1. Abran, A. & Moore J.: SWEBOK—Guide to the Software Engineering Body of Knowledge. IEEE CS Professional Practices Committee (2004)
2. Bourque, P & Fairley, R.: Guide to the Software Engineering Body of Knowledge. SWEBOK V3.0. IEEE Computer Society Press (2014)
3. Boehm, B.: Software Engineering Economics. Prentice Hall (1981)
4. Burnstein, I.: Practical Software Testing: A Process-Oriented Approach. Springer (2003)
5. Hetzel, B: A Complete Guide to Software Testing, QED Information Sciences, Inc. (1988)
6. IEEE Std 610.12-1990.: IEEE Standard Glossary of Software Engineering Terminology. Standard Coordinating Committee of the Computer Society. IEEE (1990)
7. Pressman, R. & Maxim, B.: Software engineering: A practitioner's approach. 8th Ed. McGraw-Hill Publishing, (2015)
8. Sánchez, S. Sicilia, M. y Rodriguez, D.: Ingeniería del Software: Un enfoque dese la perspectiva del SWEBOK, Alfaomega (2012)
9. Laporte, C. & April, A.: Software Quality Assurance, IEEE CS - WILEY (2018)
10. De Souza, E., De Almeida, R. & Vijaykumar, N.: Knowledge management initiatives in software testing: A mapping study, Information and Software Technology 57, 378-391 (2015).
11. Samer, Z., Salleh, N. & Grundy J.: A systematic mapping study of mobile application testing techniques." Journal of Systems and Software 117, 334-356 (2016).
12. Ahmad, A., Brereton, P. & Andras, P.:A systematic mapping study of empirical studies on software cloud testing methods. In 2017 IEEE International Conference on Software Quality, Reliability and Security Companion 555-562 (2017)
13. Kumaresen, P. , Frasheri, M. & Enoiu, E.: Agent-based software testing: A definition and systematic mapping study. In 2020 IEEE 20th International Conference on Software Quality, Reliability and Security Companion, 24-31(2020)
14. Scatalon, L. , Carver, J., Garcia, R. & Barbosa, E.: Software testing in introductory programming courses: A systematic mapping study. In Proceedings of the 50th ACM Technical Symposium on Computer Science Education, 421-427 (2019)
15. Genero, M. Cruz-Lemus, J. y Piattini, M.: Métodos de Investigación en Ingeniería de Software, Ed. Ra-Ma (2014)
16. Petersen, K. et al.: Systematic mapping studies in software engineering. In Proceedings 12th International Conference on Evaluation and Assessment in Software Engineering, 2008.
17. Kitchenham, B & Charters, S. Guidelines for performing Systematic Literature Reviews in Software Engineering. Version 2.3, EBSE (2007)
18. Malhotra, R.: Empirical Research in Software Engineering: Concepts, Analysis, and Applications. CRC Press (2015)

19. Wohlin, C. Host, M., Runeson, P., Ohlsson, M., Regnell, B. and Wesslen, A.: Experimentation in Software Engineering, Springer Science & Business Media (2012)
20. Kitchenham, B., Budgen, D. and Brereton, P.: Evidence-Based Software engineering and systematic reviews. CRC Press (2015)
21. García, N., Díaz, J. y Aguilar, R.: Investigación sobre Técnicas y Estrategias de Prueba del Software: Un Estudio de Mapeo Sistemático sobre la Última Década. Research in Computing Science, 150 (11), (2021)
22. Garousi V. & Zhi, J.: A Survey of Software Testing Practices in Canada. Journal of Systems and Software. Vol. 86, 1354-1376 (2013).
23. https://docs.google.com/spreadsheets/d/1j5X399yH-5h_Tq96hTcf1vlWC_2bLN6w/edit?usp=sharing&ouid=106183641574598895421&rtpof=true&sd=true