Agile Testing in Brazil: A Systematic Mapping

João Farias(✉), Alexandre Vasconcelos(✉), and Ivaldir Junior(✉)

Universidade Federal de Pernambuco, Recife, PE 50670-901, Brazil
{jgfd,almv,ihfj}@cin.ufpe.br

Abstract. The increasing adoption of agile methodologies is changing the landscape of software testing. Both the manifold of testing techniques and the agile mindset have shown as challenges for professionals. In this context, this work conducted a systematic mapping of studies published in Brazil, with the goal of gathering evidence on the use of agile testing practices in the country. Using the Agile Testing Quadrants as classification model, testing techniques, their benefits and challenges were identified. This systematic mapping revealed that, despite the quality improvements and reduction of costs brought by the use of agile testing techniques, learning curves and lack of management cooperation introduce significant difficulties for the full adoption of agile testing.

Keywords: Software testing · Agile methodologies · Systematic mapping

1 Introduction

Given the exponential growth of human dependency upon software and the high risk associated with unexpected and incorrect behavior in computational systems, it is necessary to give prominence to testing processes in industry. The quality of testing activities is crucial for the success of software projects.

Traditional testing methodologies are focused on validating the software against a detailed and thorough set of requirements, ensuring that the expected product has been built. Conversely, agile methodologies recognize that is almost impossible to architect a whole project up front. Thus, in agile projects, testing is used to generate information about the product behavior, with the goal of driving the next steps to be taken. This paradigm demands from the professionals involved in software testing new skills and flexibility.

However, it is notorious the difficulty in adoption of testing in agile projects, due to issues related to technical capability, appropriated planning and skills to deal with changes. A survey promoted by SauceLabs [1] with 732 software professionals showed that 89% of them follow agile methodologies, but only 32% consider themselves totally immersed in the agile mindset. In the light of these problems and the importance of testing in software projects, this paper aims to obtain evidence of the application of agile testing techniques in Brazil. These evidences will provide an overview of studies of the field in the country and help researchers and practitioners to further improve their work in this context.

V. A. Santos et al. (Eds.): WBMA 2017, CCIS 802, pp. 31–43, 2018.
https://doi.org/10.1007/978-3-319-73673-0_3

After this introductory section, Sect. 2 presents the theoretical references about testing and the agile movement. Section 3 explains the procedure used by this paper to search and select primary studies. Section 4 summarizes the findings of the research. Final considerations and suggested future works are exposed in Sect. 5.

2 Theoretical Background

This section explains the theoretical basis of this systematic mapping. Firstly, the origins and purpose of the agile movement are presented. Secondly, the definition of agile testing used throughout this study is detailed.

2.1 The Agile Movement

The experiences that led to the emergence of the agile development movement, which began in the 1990s, came as a reaction to traditional methodologies. These methodologies are focused on up front planning, static processes, dense documentation, and rigid requirements. In 2001, the Agile Manifesto [2] brought a set of 4 values and 12 practices for software development focused on customer interaction, valuing developers as thinking assets, flexibility for change and constant user feedback. These values and practices are rapidly being adopted by the software development practitioners and companies as a way to produce high quality products in short periods of time.

2.2 Software Testing

Software testing promotes the mechanisms to verify the behavior of a software product against a set of requirements, in order to mitigate the risks in its use. These activities, on average, represent 45% of the development costs of a product; additionally, inaccuracy in early failure detection increases the associated repair cost up to 100 times [3]. Thus, it is evident the importance of performing testing effectively and efficiently.

2.2.1 Agile Testing

Agile methodologies treat testing from a different perspective. Since agile projects are highly dynamic in nature and composed by short iterations, testing in these environments aims to provide information to drive planning for the following iterations and to increase the confidence of the development team in the value added by the product [4]. Crispin and Gregory developed, as an improvement to Marick's work [5], a classification model for the most commonly techniques for software testing in agile development projects, called Agile Testing Quadrants [6] (Fig. 1).

The model consists of four categories of tests with different objectives, divided by two axes. The Q1 and Q2 quadrants aim to support the development team in its activities, ensuring a better understanding of the product in development. Q1 quadrant encompasses the tests that drive the architecture design, validating its internal quality [11]. These tests give developers confidence that significant code changes do not cause functionality regressions [12]. Q2 quadrant consists of tests pertaining to requirements

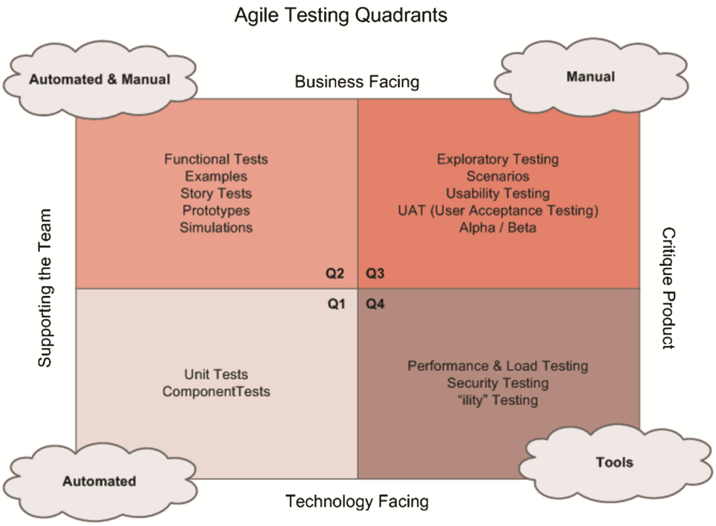

Fig. 1. Agile testing quadrants

design. Functional tests, generally automated, are performed in order to guarantee the system's external quality of the system. Prototyping, mock-ups, and wireframes help to validate whether the understanding of requirements is correct before functionality is implemented.

Tests of the Q3 and Q4 quadrants criticize the implemented product, with the goal of generating information to guide the progress of development. Q3 quadrant consists of tests which investigate if the product meets the user needs, using usability validations and exploratory testing. Q4 quadrant investigates the behavior of the system in production environment. Performance, load and stress tests validate if the system is able to withstand situations of extreme use. Security testing is critical to ensure that the product and the information it handles are protected from malicious attacks.

3 Research Method

The systematic mapping of the literature performed in this study followed the instructions of Kitchenham and Charters [7]. In the following sections, the steps covered during the research are distilled.

3.1 Research Questions

This study aims to answer the following Primary Research Question (PRQ):

- PRQ: How agile testing is conducted in Brazil?

The following Secondary Research Questions (SRQ) guide the process of systematic mapping of the literature:

- SRQ1: Which techniques related to the First Quadrant of Agile Testing (Q1) are used?
- SRQ2: Which techniques related to the Second Quadrant of Agile Testing (Q2) are used?
- SRQ3: Which techniques related to the Third Quadrant of Agile Testing (Q3) are used?
- SRQ4: Which techniques related to the Fourth Quadrant of Agile Testing (Q4) are used?

3.2 Search Strategy

Following the recommendations of Wohlin [8], a search was made based on the process of snowballing. However, given that this study is focused on the research carried out in Brazil, it was sought to identify researchers interested in agile testing working in the country.

From a list of researchers formed by literature indication [9] and manual inspection on conference proceedings, a search was conducted in the Lattes[1] curriculum system on April 16, 2017. For each curriculum analyzed, we selected the papers related to agile testing (published between 2012 and 2016) and the set of authors who collaborated in these works. The new list of researchers was analyzed by the same procedure. The process ended when no new researcher was identified. The process is described in Fig. 2.

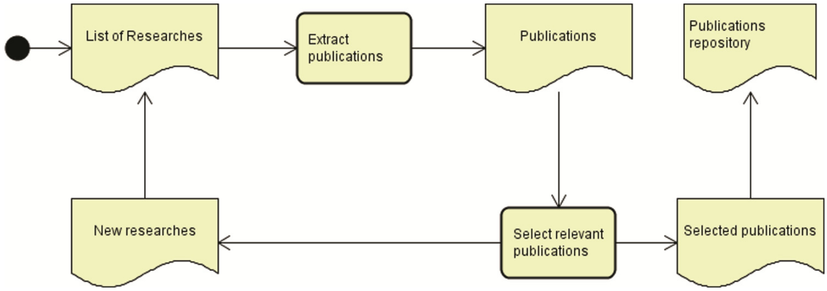

Fig. 2. Snowballing process

The process denominated “Extract publications” was supported by the tool Script-Lattes [10]. This tool aggregates publications of a set of Lattes curricula in a well-structured report, allowing the analysis of curricula in batch.

Using the ScriptLattes’ report as input, the process denominated “Select relevant publications” was performed in three steps, using the inclusion and exclusion criteria from Sect. 3.3. Firstly, the report was analyzed in a semi-automatized way, utilizing a

[1] The “Lattes Curriculum” is considered the brazilian standard for information of scientific and academic production, students, professors, researches and professionals envolved in science and technology.

Python script[2] developed by the authors. This program asks the user if the title of a paper is relevant to the mapping, saving the paper's meta-data in a report when a positive answer is given. If the paper title was not determinant to exclusion of the search process, it was considered for further analysis. In the second step, the abstract of each paper was analyzed, filtering the studies which have shown non-conformity with the inclusion and exclusion criteria. Lastly, the remaining papers were analyzed in full text, excluding the papers that do not conform with the inclusion and exclusion criteria.

3.3 Inclusion and Exclusion Criteria

Table 1 lists the inclusion and exclusion criteria for primary studies. To be selected, a study must meet all the inclusion criteria and can not meet any of the exclusion criteria.

Table 1. Inclusion and exclusion criteria

Inclusion criteria	• Studies which exercise at least one Agile Testing technique • Studies with validation in industry; • Primary studies; • Peer-reviewed studies; • Complete studies published in periodicals or annals of congresses; • Qualitative or quantitative research;
Exclusion criteria	• Repeated studies, with no new information; • Chapters of books, artistic production, patents, abstracts, texts in newspapers or magazines; • Secondary or tertiary studies; • Studies limited to educational application; • Studies not available in UFPE's institutional network; • Studies not performed in Brazil;

3.4 Data Extraction and Synthesis

Information about primary study context, used techniques, and results were extracted in a standard form. The data was analyzed using scripts written in Python. Charts and tables were created using the LibreOffice tool.

3.5 Process Overview

The research was conducted in 3 stages. The initial list of authors was created using the list of researchers of agile methodologies in Brazil indicated by Goldman and Katayama [9], adding to a set of authors who published relevant (considering the criteria from Sect. 3.3) papers in the conferences SBES, SBQS, WAMPS, WASHES and WTDQS

[2] The Python script is available on the following link: https://goo.gl/PguSfd.

between 2012 and 2016. From this list of authors, the snowballing process described in Sect. 3.2 formed the studies repository of this mapping.

4 Results

4.1 Results of the Search Procedure

The snowballing process of authors resulted in the analysis of papers from 239 researchers, from which a total of 4664 unique studies were extracted. The analysis of titles and abstracts reduced the number of studies to 223. After the full text analysis, the final set of 40 studies was defined. Figure 3 shows the study numbers resulting from each step. The list of selected articles is available for download[3].

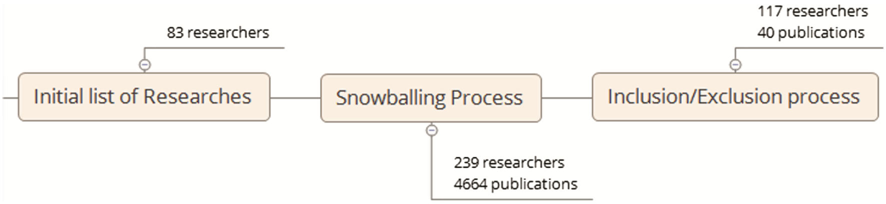

Fig. 3. Literature search results

Figure 4 shows the distribution of papers by year of publication. Of the 40 studies selected, 55% were published in the last 2 years, showing the growth of experiments with agile testing techniques in the Brazilian academic community.

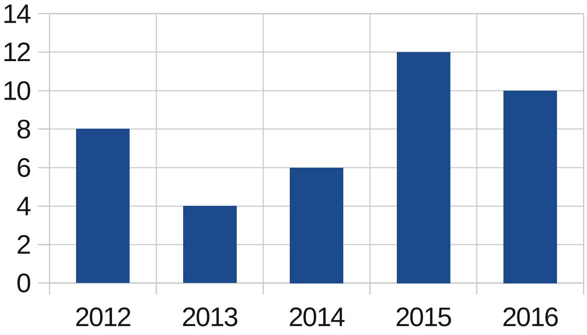

Fig. 4. Distribution of papers per year of publication

The primary publishing medium in these studies were conferences, symposia and workshops. The events with the highest number of publications were the Brazilian Software Quality Symposium (SBQS) and the Conference on Software Engineering and Knowledge Engineering (SEKE).

[3] List of articles available on the following link: https://goo.gl/19IAXO.

Figure 5 displays the paper distribution by related Agile Testing Quadrant. The most cited quadrants in the articles are related to development support (Q1 and Q2). Based on this, we can observe less interest in the study of techniques related to product critique (Q3 and Q4). Q3 is important to detect opportunities to improve user experience and validate whether the product meets the user needs. While Q4 is critical to guarantee the sustainability of software in the production environment, especially in times of intense use or under malicious attacks.

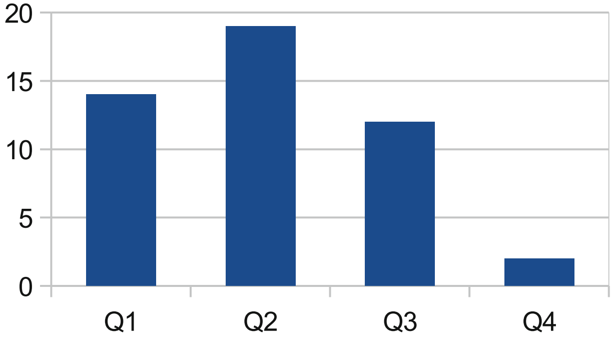

Fig. 5. Studies distribution per related Agile Testing Quadrant

The majority of the selected studies were the result of studies in the Northeast and North regions of Brazil, especially the states of Pernambuco and Amazonas, followed by the South and Southeast regions. The most active universities were UFAM, UFPE, UFCE and PUCRS. An important observation is that only 22 researchers (18% of the total) are connected to non-academic institutions, showing a possible lack of incentive

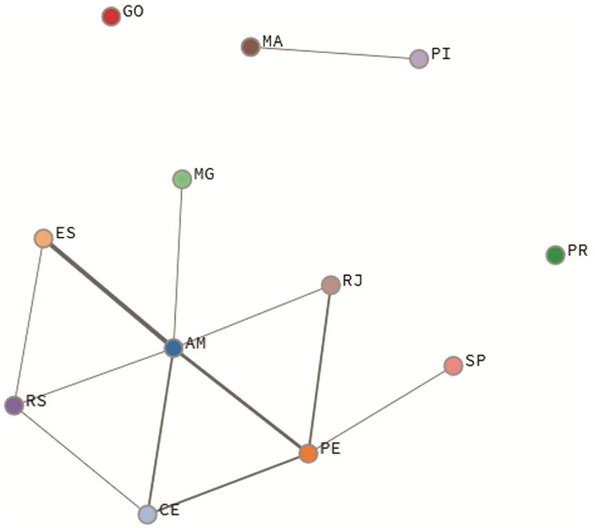

Fig. 6. Collaboration graph between researches of different states

to research and dissemination of experiences directly from the industry. Figure 6 displays the relationship of collaboration between researches of different states.

4.2 Evidences of Agile Testing Practices in Brazil

This section presents the results concerning each Secondary Research Question.

SRQ1: Which techniques related to the First Quadrant of Agile Testing (Q1) are used?

A total of 14 studies reported test techniques which support the development team with focus on code quality, as shown in Fig. 7. These studies demonstrated the use of 11 different techniques. The most used technique was Unit Testing, followed by Test-Driven Development (TDD) and Static Code Analysis.

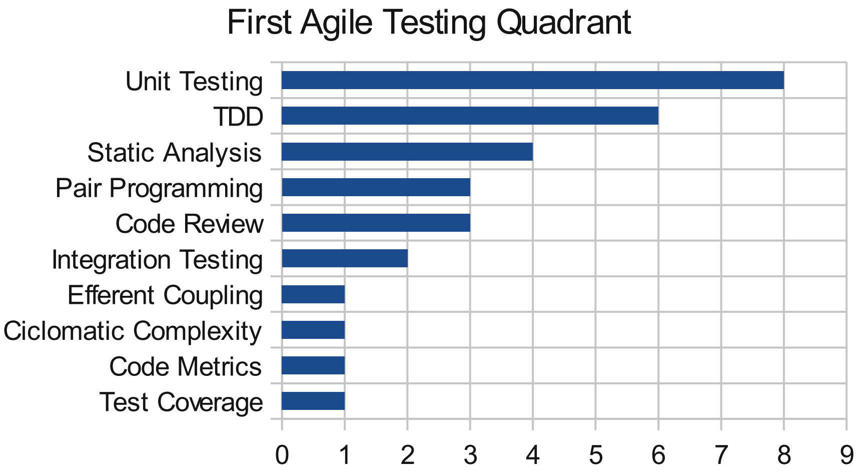

Fig. 7. Techniques related to the First Agile Testing Quadrant

The work of Filho et al. [13] studied the impact of the TDD methodology in relation to the severity of errors detected after the building process. In projects that used TDD, only 27% of the errors found after building were considered severe. In projects that did not use TDD, 51% of these errors were considered severe.

Static analysis proved to be a good tool to ensure internal software quality [12], allowing teams to manage their technical debt and giving quick feedback on possible code degradation, such as code smells and high coupled components [15]. This software internal quality control enables teams to make changes without causing unexpected damages to the system, constantly adding value to the product [12].

SRQ2: Which techniques related to the Second Quadrant of Agile Testing (Q2) are used?

A total of 19 studies reported testing techniques aimed at supporting development with a focus on business, as shown in Fig. 8. These studies demonstrated the use of 5 different techniques. The most used techniques were Automated Functional Tests (AFT) and Usability Inspection.

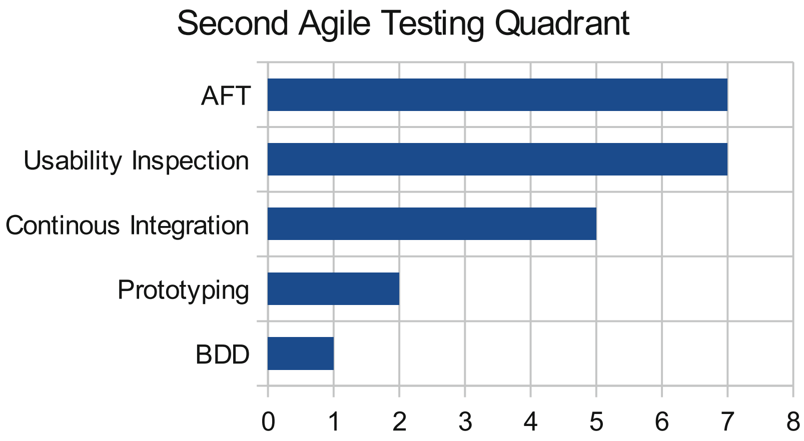

Fig. 8. Techniques related to the Second Agile Testing Quadrant

Due to the fast feedback and capacity to guarantee the external quality of the software, automated functional tests were shown as fundamental in agile projects. De Castro et al. [16] developed an abstraction of the Selenium framework for database verification which reduced the time of functional tests execution by 88% when compared to manual tests. AFT was also effective in contexts of mobile applications [17] and critical software [18].

The works of De Vaz et al. [19] and Rivero et al. [20] defined usability inspection procedures, demonstrating how this technique is able to detect failures in the requirements elicitation phase. The use of tools that aid the inspection process was effective, however, significant differences were observed in the quantity and severity of the defects when comparing inspectors of different levels of experience, reinforcing the need to value human skill in testing.

SRQ3: Which techniques related to the Third Quadrant of Agile Testing (Q3) are used?

A total of 12 studies reported testing techniques with the aim of criticizing the product with a focus on business, as shown in Fig. 9. These studies demonstrated the use of 4 different techniques. The most cited technique was Usability Testing.

The main objective of usability testing is to verify if the end user can use the product effectively and easily. The tester can rely on a list of usability principles [20], use tools for automatic fault detection [22, 23] and collect feedback from users in the form of surveys.

Falcao and Soares [24] used usability testing to validate competitive advantage a product. They analyzed whether a gesture-based device could bring effectiveness improvements to design professionals as compared to traditional mouse and keyboard-based input tools. After raising a list of usability issues, in addition to data on input error rate and effectiveness of use, the researchers showed that the device would not serve its users effectively.

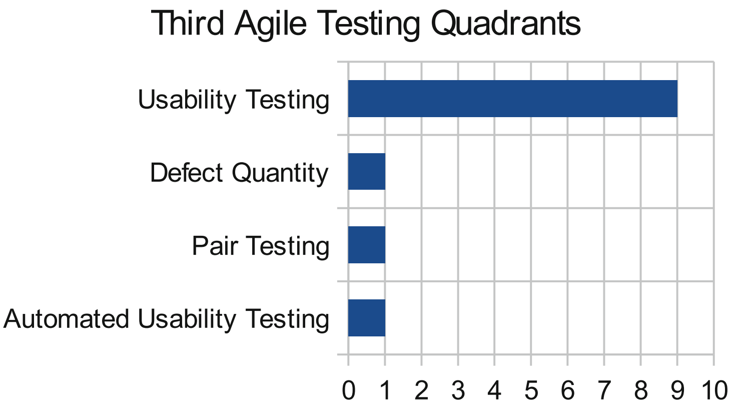

Fig. 9. Techniques related to the Third Agile Testing Quadrant

SRQ4: Which techniques related to the Fourth Quadrant of Agile Testing (Q4) are used?

A total of 3 studies reported testing techniques with the aim of criticizing the product with a technological focus, as shown in Fig. 10. These studies demonstrate the use of Performance Testing and Security Testing techniques.

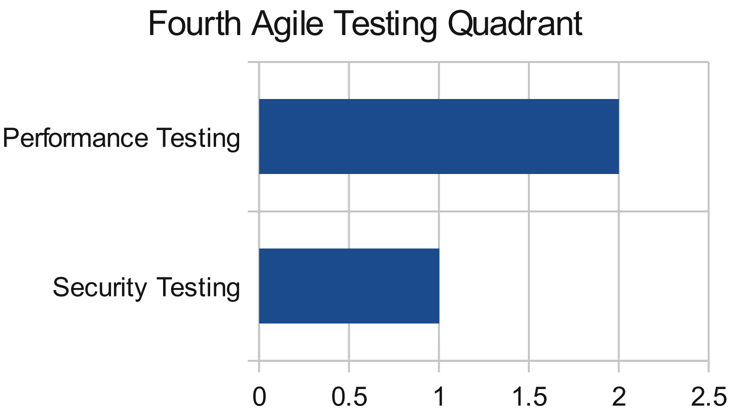

Fig. 10. Techniques related to the Fourth Agile Testing Quadrant

Tests from this quadrant evaluate the product under production conditions, where peaks of use and malicious attacks can lead to serious organizational damage. However, these tests, due their purely technical nature, are often postponed until the final stages of the projects [6]. Lucena and Tizzei [25] highlighted the problems caused by this type of decision:

> “When the software is finally deployed in the real customer environment one often finds integration and performance issues which could be prevented earlier. Late software deployments also limit useful feedback for the development team and delays customer return of the investment.”

5 Final Considerations

This section summarizes the findings of this study and provide insights about the state of agile testing in Brazil.

5.1 Results

The frequency of techniques revealed by this study are similar to global surveys [1]. The Q1 quadrant tests proved to be able to increase the internal quality of the code, technically enabling programmers to incorporate the agile principle of “embrace change”.

In agile projects, it is common that the requirements understanding becomes a problem, due to the inexhaustive documentation and rapid iterations. It was shown that techniques such as usability inspection and automated functional testing are able to mitigate this problem by ensuring that the correct product is being built.

The high number of papers reporting the use of usability testing can be justified by the fact that this technique is part of a set of user-centered design techniques. It was observed how the structured application of usability testing, especially when supported by tools, allows the detection of problems that seriously diminish the value brought by the software [22, 23].

Similar to Mohammed et al. [26], the studies related the Q4 quadrant have highlighted common problems of priority management. Due to their technical nature, security and performance issues are usually delayed until the end of the project. The problems detected at this stage are extremely costly and seriously reduce the value of the developed product.

5.2 Research Limitations

Approximately 12% of the papers selected by the snowballing process could not be analyzed in full text because they were not available for download in the UFPE’s network. Possibly some relevant work was not included in the analysis.

Furthermore, this systematic mapping used only one source of information on the investigated researchers, named the Lattes Curriculum. Although Lattes is the standard for scientific research in Brazil, using it as the single source of information is twofold limiting. Firstly, the Lattes curriculum is primarily used for academic researches. The low number of non-academic papers discussed on Sect. 4.1 can be a result of this limitation. Second, it increases the likelihood of missing published works not included in researches’ curriculum, specially for non-academic papers. Aggregating the Lattes results with other sources of information, such as ResearchGate, and direct contact with the researches would mitigate the risk of missing published papers related to agile testing.

5.3 Future Studies

The snowballing strategy presented in Sect. 3.2 was based on the work of Melo et al. [27]. We intend to validate quantitatively the effectiveness of this strategy in comparison with the technique presented by Wohlin [8].

The results of this systematic mapping will be used as source of information for the construction of a survey in practitioners, which will be conducted in the Porto Digital of the city of Recife. The goal is to compare the use of agile testing techniques in the Brazilian academic environment and the practitioners in this hub of development.

References

1. Sauce Labs: Testing Trends in 2017: A Survey of Software Professionals. https://saucelabs.com/news/sauce-labs-releases-third-annual-state-of-testing-survey-results. Accessed 5 June 2017
2. Agile Manifesto: Manifesto for Agile Software Development. http://www.agilemanifesto.org. Accessed 5 June 2017
3. Viana, V.: Um Método para Seleção de Testes de Regressão para Automação. Dissertação de Mestrado pelo Centro de Informática da UFPE (2006)
4. Williams, L., Cockburn, A.: Agile software development: it's about feedback and change. Computer **6**, 39–46 (2003)
5. Marick, B.: Agile testing directions: tests and examples. http://www.exampler.com/old-blog/2003/08/21.1.html#agile-testing-project-1. Accessed 5 June 2017
6. Crispin, L., Gregory, J.: Agile Testing: A Practical Guide for Testers and Agile Teams, 1st edn. Addison-Wesley, Reading (2009)
7. Kitchenham, B.A., Charters, S.: Guidelines for performing systematic literature reviews in software engineering, vol. 2.3, EBSE-2007-01, Keele, UK (2007)
8. Wohlin, C.: Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: 18th International Conference on Evaluation and Assessment in Software Engineering, pp. 1–10 (2014)
9. Goldman, A., Katayama, E.: Retrato da comunidade acadêmica de métodos ágeis no Brasil. In: Workshop Brasileiro de Métodos Ágeis. WBMA, pp. 1–10 (2011)
10. Mena-Chalco, J.P., Marcondes Jr., C.R.: ScriptLattes: an open-source knowledge extraction system from the Lattes platform. J. Brazilian Comput. Soc. **15**(4), 31–39 (2009)
11. Beck, K.: Extreme Programming Explained: Embrace the Change, 1st edn. Addison-Wesley, Boston (2000)
12. Fowler, M.: Refactoring: Improving the Design of Existing Code, 1st edn. Addison-Wesley, Boston (1999)
13. Filho, M.C., Vasconcelo, J.L., Santos, W.B., Silva, I.F.: Um Estudo de Caso sobre o Aumento de Qualidade de Software em Projetos de Sistemas de Informação que Utilizam Test Driven Development. In: SBSI, pp. 315–326 (2012)
14. Gouveia, V., Júnior, N.L., Furtado, A.P., Junior, I.F., Furtado, F.: Avaliando as métricas ágeis utilizadas pelas empresas certificadas no nível F do MPS.Br. In: WAMPS, pp. 85–93. (2016)
15. Lucena, P., Tizzei, L.P.: Applying software craftsmanship practices to a scrum project: an experience report. In: WASHES, pp. 73–84 (2016)
16. De Castro, A.M.F.V., Macedo, G.A., Collins, E.F., Dias-Neto, A.C.: Extension of Selenium RC tool to perform automated testing with databases in web applications. In: AST, pp. 125–131 (2013)

17. Villanes, I.K., Costa, E.A.B., Dias-Neto, A.C.: Automated mobile testing as a service (AM-TaaS). In: SERVICES 2015, pp. 79–86 (2015)
18. Braz, A., Rubira, C.M.F., Vieira, M: Development of complex software with agile method. In: Agile Conference, pp. 97–101 (2015)
19. Vaz, V.T., Conte, T., Travassos, G.H.: Empirical Assessments of a tool to support Web usability inspection. CLEI Electron. J. **16**(3), 6–22 (2013)
20. Rivero, L., Kawakami, G., Conte, T.U.: Using a controlled experiment to evaluate usability inspection technologies for improving the quality of mobile web applications earlier in their design. In: SBES, pp. 161–170 (2014)
21. Neves, C.E., Campos, F.: Teste de Usabilidade do Módulo de Atendimento da Ferramenta Desktop de Gestão de Clínicas de Diagnóstivo por Imagem Uniclinika Ris. In: CONERG, pp. 117–135 (2014)
22. De, R.Q., Ribeiro, C.P., Monteiro, R., Marcio, M.: Análise de Usabilidade da Homepage de um Equipamento de Eletroencefalograma: Emotiv Epoc. In: ERGODESIGN, pp. 47–60 (2014)
23. Nascimento, I., Gaspar, W., Conte, T., Gadelha, B., Elaine, H.: Melhor prevenir do que remediar: Avaliando usabilidade e UX de software antes de levá-lo para a sala de aula. In: SBIE, pp. 806–815 (2016)
24. Falcao, C., Lemos, A.C., Soares, M.: Evaluation of natural user interface: a usability study based on the leap motion device. In: AHFE Conference, pp. 5490–5495 (2015)
25. Ali, S., Briand, L., Hemmati, H., Panesar-Walawege, R.: A systematic review of the application and empirical investigation of search-based test case generation. IEEE Trans. Softw. Eng. **5**, 1–22 (2008)
26. Mohammed, N.M., Niazi, M., Alshayeb, M., Mahmood, S.: Exploring software security approaches in software development lifecycle: a systematic mapping study. Comput. Stand. Interfaces **50**, 107–115 (2017)
27. Melo, C., Santos, V., Katayama, E., Corbucci, H., Prikladnicki, R., Goldman, A., Kon, F.: The evolution of agile software development in Brazil: education, research, and the state-of-the-practice. J. Braz. Comput. Soc. **19**(4), 523–552 (2013)