Journal of Software Engineering Research and Development, 2022, 10:8, doi: 10.5753/jserd.2022.786

A survey on the practices of software testing: a look into Brazilian companies

Italo Santos [**University of São Paulo** | ***italo.santos@usp.br***]
Silvana M. Melo [**University of Grande Dourados** | ***silvanamelo@ufgd.edu.br***]
Paulo S. L. Souza [**University of São Paulo** | ***pssouza@icmc.usp.br***]
Simone R. S. Souza [**University of São Paulo** | ***srocio@icmc.usp.br***]

Abstract

[Context:] Software testing is essential for all software development, and techniques and criteria have been proposed to ensure its quality in different application domains. **[Objective:]** This survey aims at the identification of software testing practices in Brazilian industries, towards an overview of the latest testing techniques, selection processes, challenges faced, tools and metrics used by testers. **[Methodology:]** Survey questions were carefully designed for providing relevant information to both industry and academy and evaluated by testers for improving the quality of the survey. **[Results and Conclusions:]** Our study provides insights into the current software testing practices in Brazilian software companies. The results show testers select a testing technique according to the project scope under development; however, most companies have shown a lack of importance and priority regarding to the testing activity. Some challenges raised will foster new research topics, outlined by the needs faced by testers in practice.

Keywords: *Survey, Software testing practices, Software quality assurance, Brazil*

1 Introduction

Software testing is applied to the industry to assure quality through a direct analysis of the software in execution and provides realistic feedback on software behavior, thus complementing other techniques. Beyond the apparent straightforwardness of the checking of a sample of runs, software testing embraces a variety of activities, techniques, and actors, and poses many complex challenges (Bertolino, 2007).

Testing is indispensable for all software development and an integral part of software engineering, in which efforts are made towards a running program with a few possible inputs, thus checking whether the program behaves as specified. The number of possible inputs can be infinite for most programs, thus the challenge is to use a reasonable and cost-effective number of tests while also maximizing the test suite's fault detection capability.

Surveys have received much attention in research and practice as a tool to systematically analyze opinions, experiences, and expectations among the population studied (Torchiano et al., 2017). A survey is a method for the collection and summarization of evidence from a large representative sample of an overall population of interest (Molléri et al., 2016). In software engineering, surveys are one of the most common research methods for empirical investigations (Kasunic, 2005). The literature reports several surveys related to software testing practices in different countries, such as Australia (Ng et al., 2004), Canada (Geras et al., 2004; Garousi and Varma, 2010; Garousi and Zhi, 2013), Sweden (Engström and Runeson, 2010), and Brazil (Dias-Neto et al., 2017).

According to Wohlin et al. (2011), the collaboration between industry and academia can provide improvement and innovation in the industry and helps to guide new academic research. Therefore, we conducted a survey of Brazilian companies to identify software testing practices, gain knowledge, and obtain an overview of the latest testing techniques, selection processes, challenges faced, tools, and metrics used by testers. The key motivation is to collect evidence about how practitioners conduct the testing activity, how they select testing techniques to apply to a particular project and we raised the main challenges faced by practitioners bringing important results so that the software testing community can engage and develop new solutions.

This study portrays the software testing scenario in Brazilian technology companies and provides relevant information towards benefiting testers and fostering new research topics. The survey includes responses from 185 testing practitioners. It has detected the main trends in the software testing industry and areas of strengths and weaknesses. As a result, Brazilian software companies perform more testing related to the system level, which demonstrates concern about testing the product only in the final phase of development. The testing technique selection process is conducted by different positions in a company, and a systematic process must be defined to aid decision-making. Functional and structural testing are the most common testing techniques due to their high efficiency. Selenium is the most used testing tool to automate the system testing task. The proportion of testers in Brazilian software industries is usually much smaller in relation to the number of developers in the companies, which may not be an ideal scenario, since a tester applies tests to check the quality of the developers' work. Most companies (78%) do not develop research on testing activity. Our study has summarized the main challenges indicated by the respondents regarding the execution of testing activities (e.g., Lack of importance and priority of the testing activity, Lack of knowledge and training, Automation process, Tools, Schedule).

The remainder of the paper is structured as follows: Section 2 presents some concepts of software testing; Section 3 discusses some related work on testing practices; Section 4 describes the survey methodology, goal, design, execution, and sample size; Section 5 reports on a detailed analysis of the results; Section 6 is devoted to a more in-depth discussion of our findings; Section 7 addresses the threats to validity; finally, Section 8 provides the conclusions.

2 Background

To identify and eliminate faults present in the software product, VV&T - Verification, Validation, and Testing activities aim to ensure the quality of the software produced. Software testing aims to detect faults and analyze if the software runs as expected (Myers et al., 2011).

According to Abran et al. (2004), software testing is usually performed at different levels throughout the development and maintenance processes. The categorization of levels is based on either the object of testing (target), or the purpose (objective). SWEBOK divides the testing target into three levels (e.g., unit, integration, and system). Moreover, testing is applied because of specific objectives, which are more or less explicitly stated and show varying degrees of precision. The objective of testing includes, but is not limited to, reliability measurement, identification of security vulnerabilities, usability evaluation, and software acceptance, for which different approaches can be employed (Bourque et al., 2014).

For each testing level, specific testing techniques can be applied. The mainly testing techniques categories described at SWEBOK, according to the information used to derive test cases are: functional, structural and fault-based testing.

Structural or white-box testing operates at a low-level design and implementable code and is applied at all levels of the system development (e.g., unit, system, and integration testing) (Saglietti et al., 2008).

Functional or black-box testing is based solely on requirements and specifications. It requires no knowledge of the internal paths, structure, or implementation of the software under test (Copeland, 2004).

Mutation testing is a commonly used fault-based technique that uses knowledge about common mistakes made by developers to generate mutant programs, which are similar to the original but with minor syntactical changes produced by the mutant operators (Delamaro et al., 2017).

Testing techniques provide standards for the systematic design of test cases that exercise both the internal logic of software components and their input and output domains (Myers et al., 2011).

Vegas and Basili (2005) claim that the choice of a testing technique to be adopted in a software testing project is mainly based on the knowledge of the test practitioner and often does not consider all testing techniques available in academia.

Aiming at understanding the gap between academia and industry related to knowledge and application of testing approaches, previous studies conducted surveys, which bring finds about software testing practices from different countries in real software development environments (Geras et al., 2004; Groves et al., 2000; Dias-Neto et al., 2017; Garousi and Zhi, 2013).

3 Related work

This section discusses some work on software testing practices that have contributed to our research and collection of data on software testing practices in Brazilian companies.

Geras et al. (2004) conducted a regional survey on software testing and software quality assurance techniques in the province of Alberta. The results show software organizations tend to train low personnel on testing-related topics. Such a practice exerts a two-fold impact, i.e., it detects trends, which reduces and identifies quality causes (e.g., lack of testing), and reveals the difficult adoption of methodologies (e.g., extreme programming) by organizations. Differently from their study, our survey includes all companies located in Brazil and aims at a better understanding of the selection of testing techniques.

Another survey into software testing practices was conducted by Ng et al. (2004) in Australian industries. It focused on five significant aspects of software testing, namely testing methodologies and techniques, automated testing tools, software testing metrics, testing standards, and software testing training and education. The results enabled the development of practices in software testing and some observations and recommendations for the future of software testing in Australia for both industry and academia. Our study aims to understand topics such as selection testing technique processes and identify challenges faced by testers

Towards an understanding of the regression testing practices in Sweden, Engström and Runeson (2010) conducted a qualitative survey with 15 industries and validated the outcomes in an online questionnaire with 32 respondents. The main finding was regression testing practices significantly vary among organizations at different stages of a project. The survey also highlighted the importance and challenges of automation. Our study presents findings related to general testing issues, rather than specific to regression testing.

Garousi and Varma (2010) replicated a survey into software testing practices in the Canadian province of Alberta five years after the original study (Geras et al., 2004) to analyze possible changes. The criteria used for the design of the questions were relevant to the industry and alignment with the testing knowledge area of the Software Engineering Body of Knowledge (SWEBOK). The results showed almost all companies performed unit and system testing, and more organizations used observations and experts' opinions to conduct usability testing. JUnit and IBM Rational were the most widely used testing tools, and in comparison to the original study, more companies were devoting efforts to pre-release testing.

Lee et al. (2012) surveyed companies and experts involved in software testing to identify the current practices and opportunities for improvements in software testing methods and tools (STMTs). They selected companies with prominent positions in the world market and assumed their testing practices would be better than others. They analyzed such practices and the results showed low or limited use of STMTs, difficulties caused by the lack of STMTs, demand for interoperability support between methods and tools of software development and testing, and need of guidance for the evaluation of STMTs or description of their capabilities.

Garousi and Zhi (2013) presented a newer version of the previous study (Geras et al., 2004), which surveyed Canadian software testing practices. They reused the questions from the 2004 survey and added four new ones. The results showed the importance of testing-related training had increased, and functional and unit testing were the two common testing types of most attention and effort. Moreover, mutation testing had drawn attention among Canadian companies, and NUnit and Web application testing tools had overtaken the popularity of JUnit and IBM Rational. Most Canadian firms spent less than 40% of their efforts (e.g., budget and time) on testing during development.

Kassab et al. (2016) surveyed software professionals to determine the use of testing practices. The participants represented 18 industries located in 9 different countries, and the main findings included characteristics of projects, practices, organizations, and practitioners related to software testing.

Dias-Neto et al. (2017) conducted a replicated survey in two South American countries (Brazil and Uruguay) to understand the perception of testing practitioners regarding the use and importance of software testing practices. The main findings include system and regression testing are the two test types deemed most useful and important, tools to support automation of test case generation and execution or code coverage are still poorly used in their organizations, documentation of test artifacts (plan, cases, procedures, results) are useful and important for software testing practitioners.

Santos et al. (2020a), conducted a survey focused on mobile testing techniques. Their results show that most companies focus on developing native applications. The test level identified as more performed is the system test. Also, participants indicate that the correct choice of testing techniques directly influences the final product quality, and testers do not follow a defined process to select testing techniques. The testing tools most used to automate are Cucumber and Selenium.

We highlight the importance of surveys for the identification of software testing practices and their efficiency regarding the results achieved. Our work differs from the others because we aim to investigating the practices of testing technique selection in Brazilian industries. Moreover, we have listed the main challenges faced by testers while performing their activities.

4 Survey methodology

A survey is a comprehensive research method for the collection of information on descriptions, comparisons, explanations, knowledge, attitudes, and behavior (Fink, 2003). A survey collects quantitative, subjective (individual's opinions, attitudes, and preferences), and objective data (e.g., demographic information). The research process (Figure 1) follows the guidelines defined by Kitchenham and Pfleeger (2008).

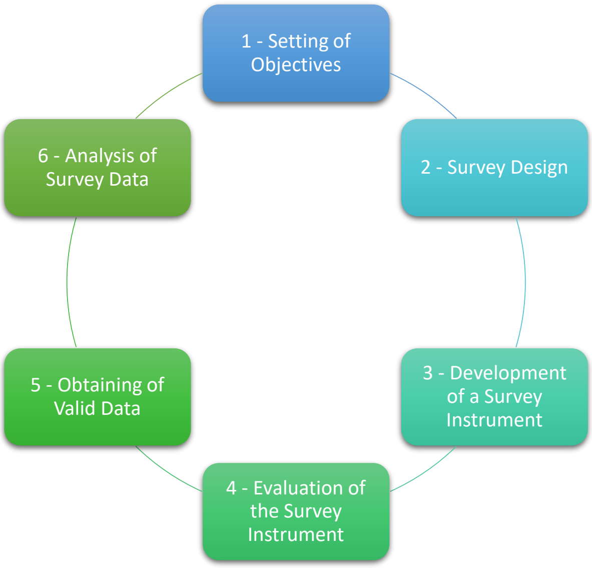
1 - Setting of Objectives
2 - Survey Design
3 - Development of a Survey Instrument
4 - Evaluation of the Survey Instrument
5 - Obtaining of Valid Data
6 - Analysis of Survey Data

Figure 1. Overview of the survey research process.

1. **Setting of objectives:** each objective is a statement of the expected outcomes or a question to be answered by the survey;
2. **Survey design:** since the type of survey used is a cross-sectional study, the participants are asked about their previous experiences at a particular fixed point in time;
3. **Development of a survey instrument:** such a development involves four steps, namely (i) search for relevant literature, (ii) construction of an instrument, (iii) evaluation of the instrument, and (iv) documentation of the instrument;
4. **Evaluation of the survey instrument:** often called pretesting, this evaluation pursues the following goals: (i) checking of whether the questions are understandable, (ii) assessment of the likely response rate and effectiveness of the follow-up procedures, (iii) evaluation of the reliability and validity of the instrument, and (iv) assurance of match of data analysis techniques and expected responses;
5. **Obtaining of valid data:** a sample, which is a subset of a population, is defined in this phase. The answers of this group should represent the entire group. When choosing a sample to be surveyed, we must keep in mind three aspects of survey design, namely (i) avoidance of bias, (ii) appropriateness, and (iii) cost-effectiveness;
6. **Analysis of survey data:** it is the final step, in which all data collected are gathered, and relevant information is extracted, so that the questions defined in the survey objectives can be answered.

4.1 Survey goal and research questions

The approach used in this survey was based on the Goal, Question, Metric (GQM) methodology and the template proposed in Caldiera et al. (1994) and Briand et al. (1996). Our survey aims to identify software testing practices in Brazilian industries to highlight strengths and weaknesses and foster work/research collaboration between academia and industry towards an overview of the latest testing techniques, selection processes, challenges faced, tools and metrics used by testers. The survey was conducted between September 2018 and December 2018 and it is available online[1]. Table 1 shows

[1] https://github.com/italo-07/surveyTeste

the research questions raised.

4.2 Survey design and questions

Our survey was based on other relevant studies (Geras et al., 2004; Ng et al., 2004; Engström and Runeson, 2010; Garousi and Varma, 2010; Lee et al., 2012; Anand et al., 2013; Garousi and Zhi, 2013) and some of the questions designed by Geras et al. (2004), and new ones related to our specific objectives were adopted.

Testers currently working in the industry were asked to evaluate the set of questions proposed. Torkar and Mankefors (2003) state feedback improve the set of questions prior to the application of a survey. The goal behind this phase was to ensure the terminology used would be familiar to a reasonable ratio of the audience. According to Garousi and Zhi (2013), occasionally, the software testing terminology used in academia versus industry may be slightly different or even confusing. The feedback received from testers was used for the validation of the survey questions.

The study employs the characterization topics presented in SWEBOK (Bourque et al., 2014). Therefore, the survey comprises the following categories: (i) testing levels, (ii) testing techniques (selection techniques), and (iii) software testing tools and testing process management. We have added three other categories, namely (i) respondents' personal and professional profiles, (ii) research & training, and (iii) challenges, which contain information on the respondents' profile and investigate research initiatives within companies towards identifying challenges faced by testers.

Survey questions are designed to provide relevant information to the industry and academy. We derived them from each of the eight research questions previously presented, which have composed a set of 34 questions, of which 7 (20%) are qualitative (in italics), and 27 (80%) multiple choices are quantitative. To help readers identify the research questions, we labeled the corresponding categories in the first column and the survey questions in the second column (Table 2).

The first category is related to respondents' personal and professional profile (Q1-Q11), and we are interested in understanding the level of education, undergraduate course, companies' name, time of experience, current position, types of software projects the companies usually test, and company size. The questions would provide an overview of the scenario of software testing in Brazil.

The second category (Q12–Q13) refers to the testing level (e.g., unit, integration, and system testing) and objectives of testing (e.g., acceptance, reliability, regression, performance, security, stress, interface, and usability testing). The survey investigates the use of testing levels by Brazilian industries and respondents' experience.

The third category is related to testing techniques and selection processes (Q14-Q23), i.e., to those who decide on the testing techniques to be used in a project under test, the way the respondents select testing techniques, their experience with other testing techniques, such as mutation or concurrent testing, and what should be considered in a selection process.

The fourth category details the testing tools and testing process management (Q24-Q29) towards a better understanding of testing tools and identification of testing practices of management applied to test teams.

The fifth category, i.e., research & training (Q30-Q33), investigates whether companies develop research on software testing and provide formal training to their testing team, and the way testers seek new knowledge to solve their technical problems. Finally, the sixth category (Q34) asks the respondents to list three challenges faced in their work routine.

4.3 Survey application

We used the search management application Google Forms[2] to design and host our online survey, which was available for four months, from September to December 2018. According to the ethics guidelines, the respondents could remove their answers from the study at any time, and researchers agreed on aggregating information from the survey and publishing a summary of the results.

To reach the maximum number of responses, invitations were sent to (i) email lists used by the Brazilian Computer Society (SBC)[3], which concerns professionals of the computer science community in Brazil, (ii) email lists used at the University of Sao Paulo for undergraduate and graduate students, and (iii) email lists of major Brazilian software companies (e.g., Brazilian Association of Software Companies (ABES)[4], Association of Brazilian Companies of Information Technology - ASSESPRO[5], Brazilian Association of Information and Communication Technology Companies - BRASSCOM[6], Porto Digital[7], and Association of Technology Companies of Sancahub - Sancahub[8]). The authors also attended scientific conferences in Brazil and requested the participants to answer the survey.

4.4 Population and sample size of respondents

The survey was available for four months, and 201 responses were obtained. To ensure the quality of the collected data, the respondents informed their emails for further contact towards the confirmation of their information. They were also asked the following question after the first category of questions (Q1-Q11): Do you have knowledge of software testing?. If the answer is no, the survey ends. The purpose here is to be sure the respondents fit the target audience, i.e., testers, developers who perform testing during development, and professionals who work in software quality. The survey data included responses from 185 testers in Brazilian companies.

Table 3 shows the total number of emails sent to the companies. 1269 companies were listed in the first column and represented Brazilian companies' associations of technology, of which 207 mail addresses failed. We believed the emails reported on the website had not been updated, therefore, we sent emails to only 1062 companies. Since 185 respondents out of 1062 companies listed contributed to the survey, a rough measure of the sample size equals 17.41% (185/1062).

[2] https://www.google.com/forms/

[3] http://www.sbc.org.br/34-listas-eletronicas

[4] www.abessoftware.com.br/associados/socios/

[5] http://assespro.org.br/

[6] https://brasscom.org.br/

[7] http://www.portodigital.org/home

[8] http://sancahub.org/

Table 1. Research questions - GQM methodology

Goal	Question	Metric
Investigate commonly used testing levels.	**RQ1 -** What are the most used testing levels in Brazilian software companies?	Set of options with the testing levels.
Understand the process of testing technique selection and familiarity of testers with the existing testing techniques.	**RQ2 -** How do you select a testing technique to test a software product?	Open questions for the collection of respondents' opinions.
	RQ3 - How often does a tester use structural, functional, mutation, and concurrent program testing?	Set of options with the software testing techniques.
Investigate the testing tools and technologies used and the way testing teams are managed.	**RQ4 -** What are the most used testing tools?	Set of tool options and possibility of selection of more than one and insertion of other tools names.
	RQ5 - How are the testing teams and processes managed?	Set of options with the process management.
Find out whether companies are concerned with research and provide training, and the way testers seek new knowledge.	**RQ6 -** Do Brazilian industries develop research on software testing?	Open questions for the collection of respondents' opinions.
	RQ7 - How do testers seek knowledge on software testing? Does their company offer any training?	Set of options with actions for seeking knowledge.
Identify research challenges.	**RQ8 -** What challenges do testers face in their work routine?	Open questions for the collection of respondents' opinions.

Table 2. Survey list of questions

Categories (RQ)	Survey Questions
Respondents' personal and professional profile	1- How old are you? 2- What is your gender? 3- What is your highest level of education? 4- If you have taken an undergraduate course, please inform its name. 5- What is your email address? *6- What is the name of the company where you work or have worked with software testing?* 7- In what state is your company located? 8- What is (are) your current position(s) in your company? 9- How many years of experience do you have in software testing? 10- What types of software projects does your company develop? 11- What is the size of your company (number of employees)?
Testing levels RQ1	12- What type of tests do you work or have you worked on? 13- Are you engaged in other activities in your company? If so, which ones?
Testing techniques, Selection process RQ2 RQ3	14- Do you decide on the testing techniques to be used on the software in your company? 15- If you answered no to the previous question, please inform the one who decides on them. 16- In your opinion, what makes a tester not consider choosing new testing techniques? *17- How do you select a testing technique for a project?* *18- In your opinion, what is important in the selection of a testing technique for a project?* *19- What do you think prevents your company from adopting new techniques and methodologies?* 20- Are you familiar with the white box test? If so, how often do you use it in your projects? 21- Are you familiar with the black box test? If so, how often do you use it in your projects? 22- Are you familiar with the mutation test? If so, how often do you use it in your projects? 23- Are you familiar with concurrent testing? If so, how often do you use it in your projects?
Testing tools and process management RQ4 RQ5	*24- What tools do you use to automate the testing activity?* *25- In your current, or latest project, what mechanisms are (were) used for generation of test cases?* 26- Does your testing team use any measures as a guide for planning or designing tests? 27- Is testing separated from development and production environments? 28- What is (was) the proportion of testers for developers (t:d) in your current, or latest software project? 29- Does your development team(s) refer to itself as agile or traditional?
Research & Training RQ6 RQ7	30- Is your company involved in research on software testing towards the development of new ways/techniques for testing software systems? 31- Does your company offer any training to the testing team? 32- If you answered yes to the previous question, please specify the training. 33- How do you usually gain knowledge in the testing area for solving technical problems in the work environment?
Challenges RQ8	*34- Provide a list of the top 3 testing challenges related to your projects.*

Our study comprised a significant number of responses in comparison with other surveys into software testing. The average number of responses was 62. The most extensive survey (Garousi and Zhi, 2013) comprehended 246 responses, whereas the lowest (Lee et al., 2012) provided 14 responses. Our survey was well above the average of responses and almost approached the survey of the highest number.

Table 3. Total of invitations sent

	Number of contacts	Failed to send email	Sent
ABES	1023	155	868
ASSESPRO	169	39	130
BRASSCOM	15	8	7
PortoDigital	48	3	45
SancaHub	14	2	12
Total	1269	207	1062

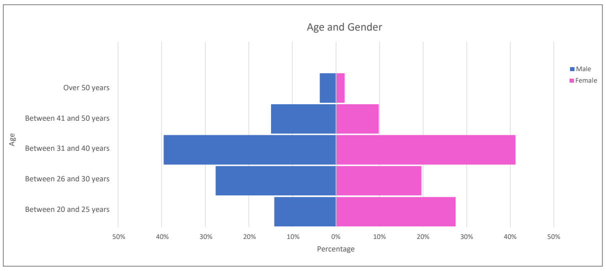

Figure 2. Respondents' age and gender distribution.

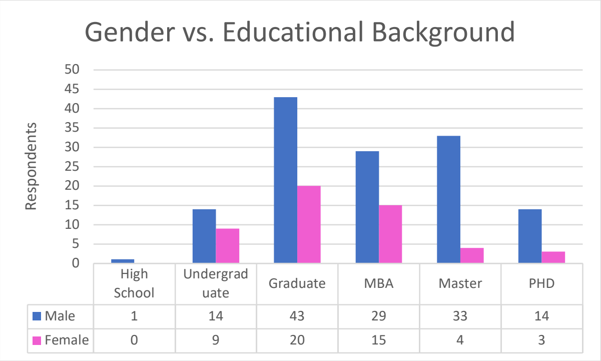

Figure 3. Relationship between gender and educational background.

5 Analysis of the survey results

5.1 Respondents' personal and professional profile

The participants' ages were divided into intervals. Most respondents (40%) are between 31 and 40 years old, whereas the second most representative group (25%) is aged 26 and 30. Other groups range between 20 and 25 years (18%), 41 and 50 years (14%), and over 50 years (3%). According to the range, the profile of participants is diversified, which contributes to the strength of the answers collected.

Regarding gender, 72% of the respondents are male and 28% female. We intended to identify the relationship between gender and age of respondents and how its representation is portrayed in the software testing scenario within Brazilian industries. Figure 2 represent this distribution and we can note that the majority of testers in Brazilian industries are male with age between 31 and 40 years and the distribution is similar for both genders.

Concerning respondents' level of education, 87% had graduated, and were divided into the following categories: graduate (34%), master of business administration - MBA (24%), masters (20%) and PhD (9%). The other respondents are undergraduates (12%), and (1%) who graduated from high school.

In Figure 3 is shown the distribution of women and men related to your academic background. The presented results show that the distribution is proportional between the genders, with a greater number of men in the pos-graduated level.

A higher level of education increases the competitiveness among IT companies and guarantees a higher quality of work. However, a proper theoretical and technical training in the area prepares professionals to better perform their activities.

Concerning the undergraduate courses informed, we grouped the data according to areas (Table 4). The majority (173 respondents) obtained an undergraduate degree in areas of science, technology, engineering and mathematics (STEM), and technological courses (e.g., analysis and systems development, computer networks, information technology management) and enrolled in baccalaureate programs (e.g., information systems, computer science, computer engineering). Only (10 respondents) had their base education in applied social sciences courses (e.g., administration, social service, psychology).

Table 4. Undergraduate courses attended by the respondents.

Course	**Percentage**
Exact sciences, engineering and technology	94%
Applied social sciences	5%
Did not take upper course	1%

We also identified the respondents according to their companies location and grouped them into the five regions in Brazil (Figure 4). The region with the most significant number of participants was the Southeast region (59%), which is the second largest region of the country and is composed of large technological centers, located in cities such as São Paulo, Minas Gerais, and Rio de Janeiro. Secondly, the Northeast region (21%) is a region comprised of 9 states and has technological centers of great recognition such as the Porto Digital located in Recife. Moreover, in the third place the South region (11%), which has poles of innovation and technology. Finally, the Midwest region (7%) and the Northern region (2%). Hence, we expected that the most significant number of respondents would come from the Southeast region, due to the higher concentration of software companies.

The position of the highest number of respondents was Analyst (82 responses), followed by the developer (33 responses) and tester (23 responses). Other positions were manager and leader (e.g., project, QA, test, technical, TI).

The years of experience in the software testing industry were divided into five categories (up to 1 year, between 1 and 2 years, between 2 and 5 years, between 5 and 10, and

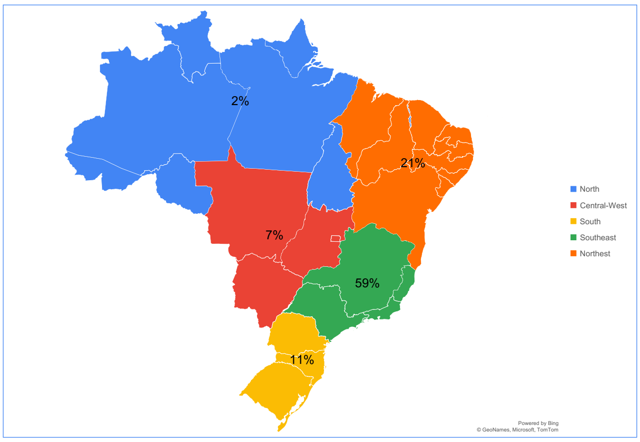

Figure 4. Geographical distribution of respondents.

above 10 years). Most candidates (76%) had two years' experience, characterized by solid experience in software testing, and some had up to 5 years' experience (27%).

To analyze the influence of the years of experience from the participant's industry positions, we compare this data at (Figure 5). Results show that there are respondents with all levels of experience in almost all positions. Although the majority of high positions as project manager and leader, concentrate respondents with more than five years of experience

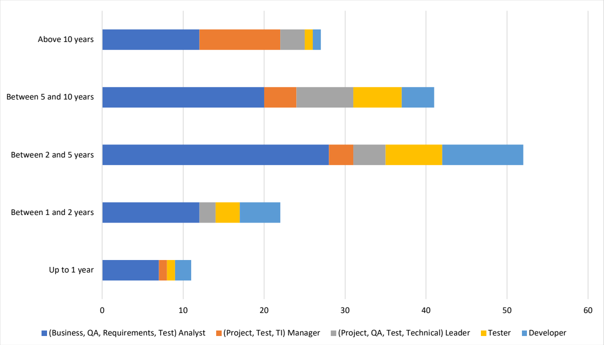

Figure 5. Relationship between years of experience and position.

In Brazil, most companies classify IT professionals as junior, senior, and expert, taking into consideration the experience time in the company, i.e., junior level encompasses up to 5 years' experience and, in general, senior level comprehends up to 9 years. Above 5 years' experience, the number of professionals is more significant in categories whose level of education associated with MBA, Masters, and PhD (Figure 6). The expert's level requires above 10 years' experience and well-established academic training, and expert professionals often work in management positions. Figure 6 shows the relationship between years of experience and the education level of each respondent.

Some types of software projects require more intensive tests (Figure 7). The category of web systems development (158 responses) showed the highest number of answers, followed by mobile development (105 responses) applications. The other systems indicated were maintenance (74), ERP (57) and banking (49). The participants reported other types of projects, such as games development, virtual reality systems, quality consulting, and educational systems.

The size of the companies the respondents work at is related to the number of employees as shown in Table 5. Most respondents worked in large companies with more than 200 employees (53%). Such organizations are more concerned with the quality of the software product developed and invest more resources in hiring professionals to perform testing activities. Moreover, from the data collected, we identify the companies that have the certification of Capability Maturity Model Integration - CMMI. This information is available at CMMI Institute[9], the model describes a five-level evolutionary path of increasingly organized and systematically more mature processes.

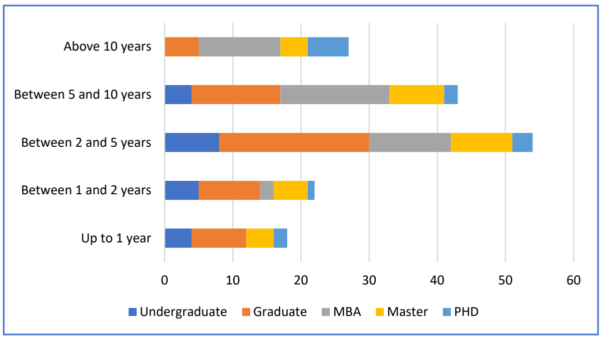

Figure 6. Relationship between years of experience and level of education.

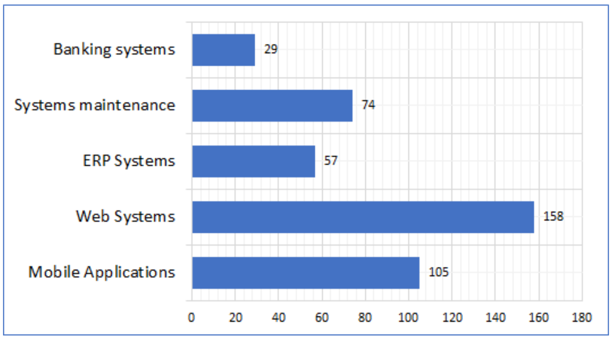

Figure 7. Types of software projects developed by companies.

Table 5. Company size (according to number of employees).

Number of employees	Respondents	Percentage
Up to 200	97	53%
Between 100 and 200	23	12%
Between 50 and 100	21	11%
Between 10 and 50	29	16%
Between 5 and 10	5	3%
Between 1 and 5	10	5%

5.2 RQ1. What are the most used testing levels in Brazilian software companies?

The participants' responses enabled the identification of the testing levels and objectives. In that case, respondents could choose more than one option. The most accomplished testing level (Figure 8) was the system test (136 responses), which tests the behavior of an entire system and is usually considered appropriate for the assessment of nonfunctional system requirements, such as security, speed, accuracy, and reliability. The second testing level was unit test (93 responses), which checks the functioning of software elements separately testable. The integration test (4 responses) was given the lowest number of responses.

[9] https://sas.cmmiinstitute.com/pars/pars.aspx

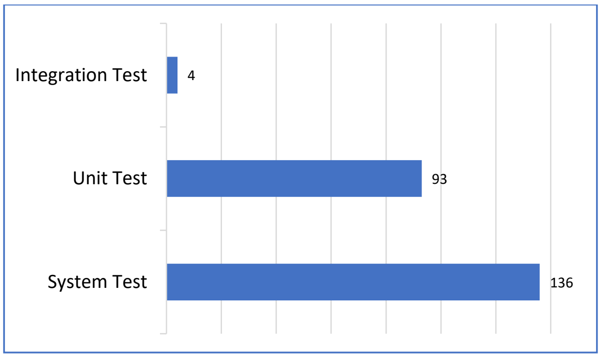

Figure 8. Testing levels used by respondents.

The most used testing objective (Figure 9) was the interface test (129 responses) checks whether the components in an interface behave as expected, and the acceptance test (124 responses) is applied when the product is almost finished, and the tester must check if the system developed has met the requirements.

A regression test (121 responses) is applied when the software has been updated and we need to verify if the new changes will influence the behavior of the unchanged part of the software. The usability test (117 responses), one of the most applied tests, evaluates the degree of ease of the software developed and contributes to the learning and use by the end-user. Other tests indicated were (i) stress test, which exercises the software's maximum capacity for determining its limits, and (ii) security test, which checks whether the software is protected from external attacks.

Although most of the respondents (132) have cited the automation test as one testing objective, it is not represented in Figure 9 because automation is not an objective. Instead, it is a way of conducting the testing activity as discussed by Dustin et al. (1999).

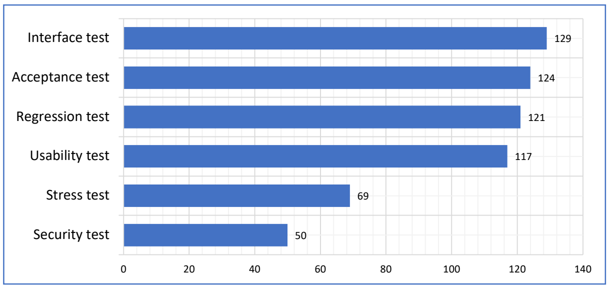

Figure 9. Testing objectives most used by the respondents.

5.3 RQ2. How do you select a testing technique to test a software product?

There are several testing techniques available, each one with different and often complementary features responsible for testing several aspects of the software. Choosing the testing technique should not only be based on subjective knowledge, but it should also incorporate objective knowledge, guided by elements that favor good choice (Santos et al., 2020c,d, 2019; Victor and Upadhyay, 2011). Our study identifies whether the respondents have the autonomy to choose a testing technique to test a software project.

Results show that the respondents selected (70%) of the testing technique to be applied in the software project. Those who did not select (30%) it informed the position responsible for the task (Figure 10). The aim is to summarize the positions within a company responsible for the testing technique selection. We identified three main positions that have this responsibility: QA Analysts (63%) is the main role that performs this task; Project Manager (23%) and Developers (14%). Through the identification of these roles, it is expected to conduct further research to propose a systematic way to select testing techniques to ensure quality and improve the selection process.

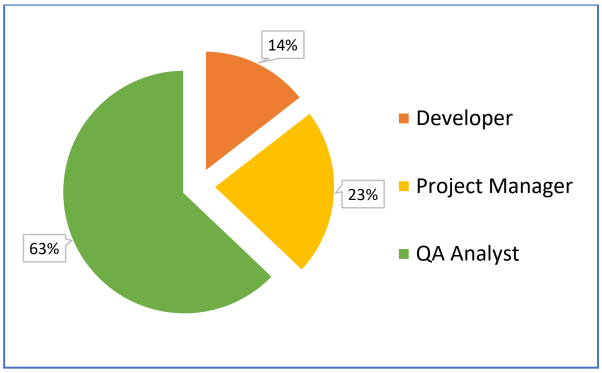

Figure 10. Positions responsible for the selection of testing techniques.

Factors that hamper the use of new techniques by testers (Figure 11) are very short deadlines (158 responses) and lack of knowledge (131). The time devoted to software testing may be very short within a project schedule, in comparison to the other activities in the development process, therefore, testers cannot apply (or even choose) new techniques for improving testing efficiency.

The cost of application of a technique (72 responses) is also an obstacle, once it is an expensive activity in the software development process and can consume a large part of the total costs of the project.

Low priority (58 responses) indicates the company is not concerned with seeking new testing techniques. Due to the efforts that must be devoted to a testing activity, companies always use the same techniques, which will become obsolete regarding advances in research in the area. The lack of technical documentation (50 responses) triggers an alert for researchers and companies that develop new techniques and should provide proper documentation to encourage their use and dissemination. Some testers declared satisfaction (43 responses) with the techniques used in previous projects; they feel no need to seek new ones, and the company does not allow (21 responses) their use.

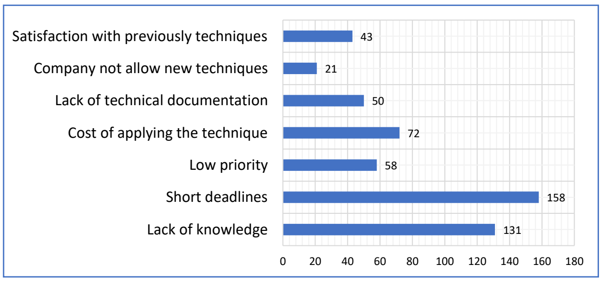

Figure 11. Factors that avoid the use of new testing techniques by testers.

The study investigates the way testers select a testing technique to be applied in a software project (Figure 12). The project scope (124 responses) supports decision-making regarding the selection of testing techniques. Testers analyze the project to be tested and choose a technique related to its characteristics. Another criterion is time (71 responses). The technique selected should be feasible to be applied according to the time defined in the schedule. Risk (26 responses) related to the implementation of the technique, involves the problems to applying the testing technique for the software under test. Costs (26 responses) are another factor considered, since testers are more worried about ensuring the best quality practice in the testing activity, and do not take costs into consideration. The complexity (20 responses) of the implementation of a technique is also indicated as a selection criterion. One of the fewest answers was about previous experiences (12 responses) of testers.

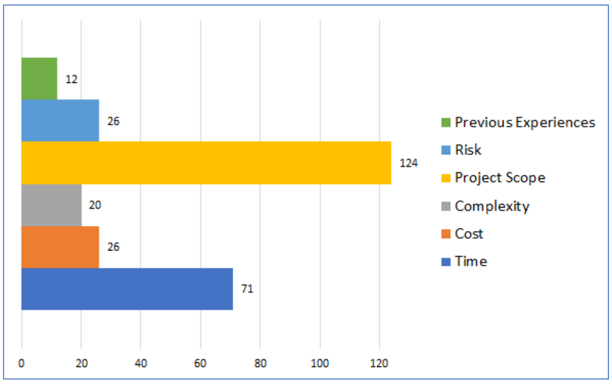

Figure 12. Criteria adopted for the selection of testing techniques.

5.4 RQ3. How often does a tester use structural, functional, mutation, and concurrent program testing?

We chose two well-known techniques (structural and functional testing) to investigate the familiarity of the respondents with some testing techniques (Figure 13), and two alternative ones (mutation and concurrent testing) to check their demand and use in Brazilian software industries.

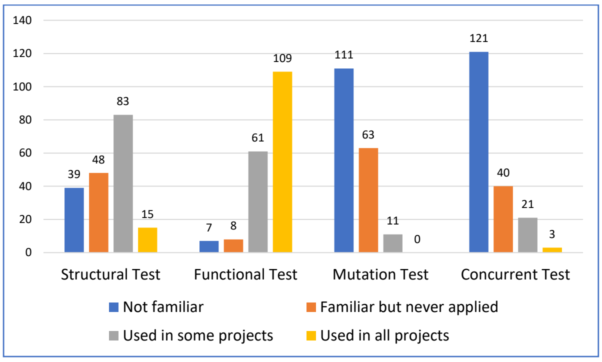

Figure 13. Respondents' familiarity with testing techniques.

According to the familiarity reported in the previous questions, the most used testing techniques identified during the execution of testing activities are functional (170 responses), and structural test (98).

Among Brazilian software industries, only a few respondents mentioned using mutation testing in some testing projects (11 responses); most of them reported they were not familiar (111 responses), while the second category of respondents indicated they were familiar, but had never applied it (63 responses). It would be a good practice if companies attempted to apply more mutation testing in their software projects to improving the testing activity.

The analysis of participants' familiarity with concurrent software testing seeks to evaluate the relevance of the area in the evolution of the computer research field. Both testing activity and research in concurrent software testing should be more investigated and widespread among testers. Most respondents are unfamiliar (121 responses) with this type of test. The second most indicated option revealed although the participants were familiar with the test, they had never applied it (40 responses). In contrast the mutation test, some respondents indicated they used this type of testing in all their projects (3 responses), and another part of respondents reported they used it in one of their projects (21 responses). The tendency to work with concurrent software testing has grown due to the evolution of computers, therefore, software companies should update themselves regarding this trend.

Brazilian software companies have shown a stronger tendency to apply more tests related to the functionality of the software developed and those that ensure the internal quality of the code.

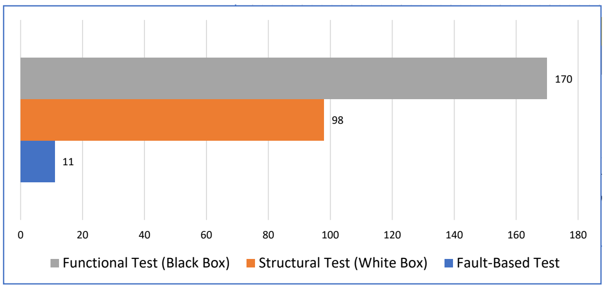

Figure 14. Most used testing techniques.

5.5 RQ4. What are the most used testing tools?

Tools and frameworks are commonly used to support the testing process (Figure 15). Selenium (27%), a portable framework used in web application testing that supports the execution and creation of system tests, received the most responses. Open-source framework JUnit (23%) supports the design of automated tests for Java language, facilitating the development and maintenance of the code, and open-source JMeter tool (15%) was designed to load functional test behavior and measure performance.

SoapUI (13%) is also an open-source tool used for web application testing. Among its functions are web services inspection, support for functional tests, load, and compliance tests. TestLink (10%) is an open-source tool that facilitates the management and maintenance of test cases and documents related to the software. SonarQube (9%) continuously inspects code quality to perform automatic revisions with static code analysis and detect bugs, smelly codes, and security vulnerabilities; it works with a variety of programming languages. Sikuli (3%) is an open-source tool that supports testing automation and automates the elements displayed on the screen.

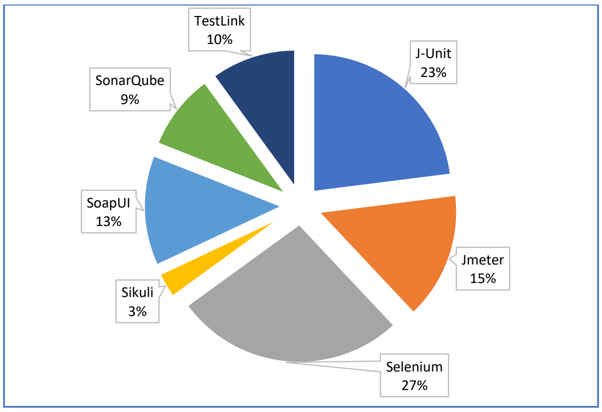

Figure 15. Automation tools and frameworks.

Concerning the design of test cases, user stories (122 responses) consist of a specification that captures what a user does or needs to do (Figure 16). It has been widely used with agile methodologies to support the definition of test cases. Tester skills (120 responses) indicate respondents usually rely on their skills from previous experience in designing efficient test in projects they participate in.

Other categories indicated are (i) bugs reported (102 responses), which shows respondents are based on the design of test cases from errors reported, (ii) code coverage (60 responses), which measures the degree at which the source code of a program is executed when a particular testing suite runs, (iii) boundary value analysis (59 responses), which concerns boundary conditions, i.e, situations directly above, and beneath the edges of input and output equivalence classes, (iv) equivalence partitioning (46 responses), which reduces the number of test cases to a manageable level while still maintaining reasonable testing coverage, and (v) control flow graphs (24 responses), which identify execution paths through a module of program code and create and execute test cases to cover those paths.

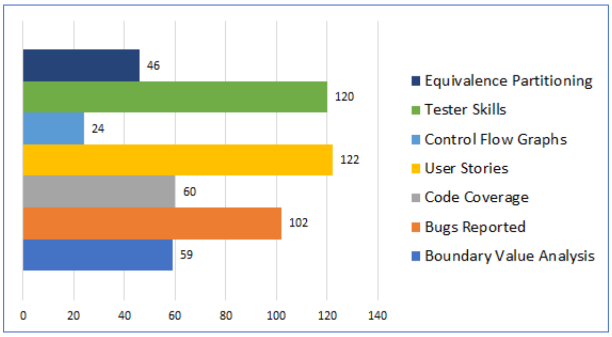

Figure 16. Forms of design test cases.

The design of test cases can consider measures to guide the activity (Figure 17). Priority level (116 responses) consists in prioritizing the creation of test cases fundamental for the correct functioning of the system. Code lines (37 responses) consist of tools for an automatic generation of test cases through the lines of code. Function points (33 responses) are a technique that establishes a measure of size, considering the creation of a test case according to the implemented functionality of the software. Complexity (20 responses) is a metric that identifies the cyclomatic complexity of the software and uses a strategy to test each path independently of the program. Therefore, the number of test cases generated corresponds to the cyclomatic complexity.

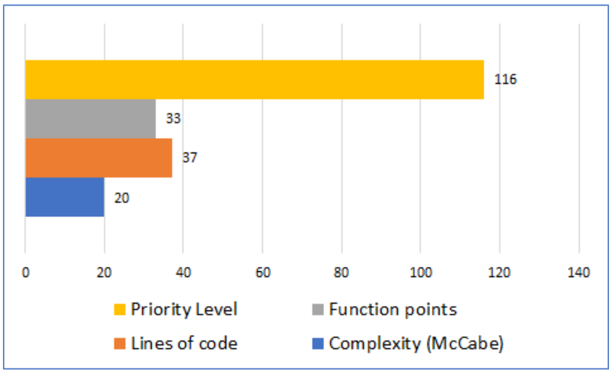

Figure 17. Measures for the design and plan of test cases.

5.6 RQ5. How are the testing teams and processes managed?

During the development and execution of testing activities, the testing team must have full autonomy to work on a version of the system that is not used by real users. Testers require the whole software available to perform their activities without the risk of modifying its functional version. The environment refers to the use of servers or different virtual machines for tasks of development, testing, and production (Table 6). Respondents indicated software and hardware are separated (58%), i.e., they might be divided into different virtual and physical servers as a safer option in case of equipment failure. Moreover, the separation among testing, development, and production environments does not apply (20%), which indicates no concern on the part of the testing team or the project management regarding the distribution of such environments. As a result, projects running in real-time can be impacted, and new tests must be applied.

Table 6. Characteristics of the testing environment.

Development environment	Responses	Percentage
Separate software and hardware	107	58%
Separate software	40	22%
Not applicable	37	20%

An important point is the identification of the proportion of testers for developers (t:d) (Figure 18). The proportion of most indicated was 1: 5 or fewer developers (68 responses), which highlights a scenario inefficient due that five developers would be working in new system functionalities and only one tester to perform testing activities in all the new functionalities developed.

A 2: 1 and higher ratio (24 replies) would be better if the testers split the several testing activities to ensure a better quality of their tasks and speed up the reporting of errors. The respondents indicated that there are no testers as the ratio of 0: 5 or fewer developers (20 responses), in this scenario developers, have the function of performing the tests in their codes, it is not a good practice because, without a professional focused to ensure the software quality, the final product developed can be compromised.

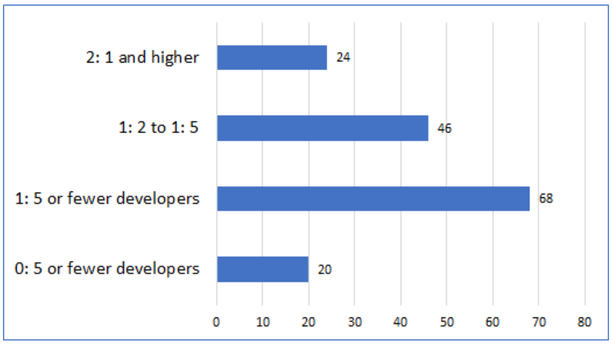

Figure 18. Proportion of testers to developers (t:d).

To understand how the development teams consider themselves, agile methodology (61%) seems to be dominant and quite popular among software companies (Figure 19). Traditional methodology (12%) received a small number of answers, which characterizes the migration of companies that used old software processes to new projects that apply agile methodologies. Other respondents indicated no specific distinction (27%), and that their companies could use both approaches. The type of development methodology used by companies must be known, since each model directly impacts the testing activity.

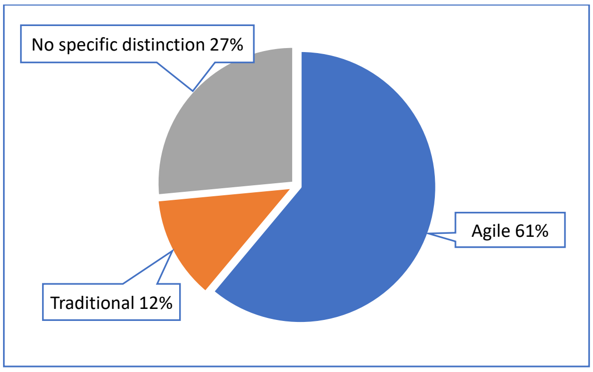

Figure 19. Characteristics of the development team.

5.7 RQ6. Do Brazilian industries develop research on software testing?

Many companies are not involved in research (78%), which is not an ideal scenario. The lack of research investments by companies is an issue to be addressed. Santos et al. (2020b), highlights some research opportunities to indicate new research directions at the intersection of testing and the software ecosystem that could be further explored for companies in collaboration with the industry. Research initiatives are essential for enabling significant advances in software testing. Some respondents indicated that 22% of companies invested in research, despite its importance for the evolution of the software testing area. These companies show a concern with the development of research and prioritize the research area through a specific department to conduct research in software testing. We investigate whether company size influences the amount of research investments. Figure ref fig: research-company shows that the company size and, consequently, its investment capacity does not affect the research initiative.

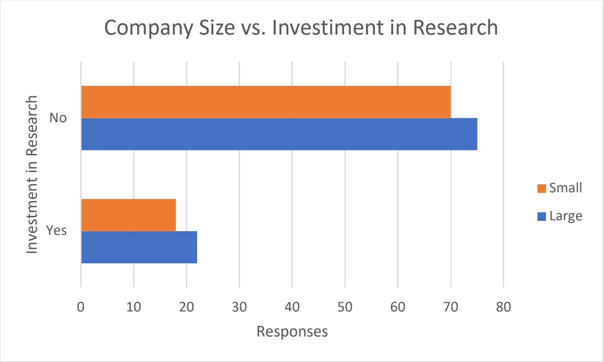

Figure 20. Research initiatives from companies.

5.8 RQ7. How do testers seek knowledge on software testing? Does their company offer any training?

Most companies (58%) offer no training to the testing team, and those that provide it (42%) aim at improving their employees' skills.

Among the training provided by companies, training on testing techniques (24 responses) is indicated as advantageous for spreading knowledge about new testing techniques among testers and improving the execution of their work activities. Web courses (20 responses) are offered through online platforms, and companies pay for their employees to access them. Internal training (20 replies) is provided when companies promote the sharing of internal knowledge among their employees. Training related to automation and tools (20 responses) aims to improve the forms of testing automation and find the available automation tools that can contribute to the automation activity.

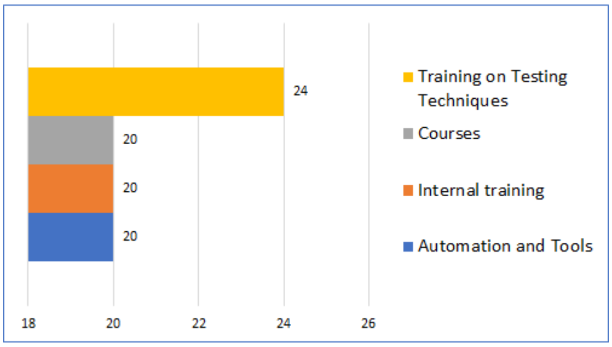

Figure 21. Type of training provided by respondents' companies.

Both IT professionals and testers commonly face challenges during the execution of their tasks. To comprehend how these professionals seek new information (Figure 22)., results indicate that testers usually seek new knowledge through conversations with more experienced testers (125 responses), which identifies the need for communication among employees of the same team, since the most experienced ones can share their knowledge with beginners, and vice versa, it will help to increase team integration. Other options indicated were reading scientific papers (106 responses) and attending courses (106 responses), testers acquire new content to help them solve their problems, which reinforces the importance of research and collaboration between industry and academy.

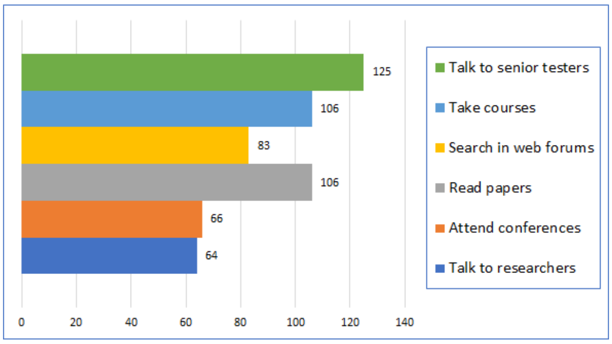

Figure 22. Actions for the obtaining of knowledge.

Other options were (i) search in web forums (83 responses), which consists of the sharing of questions with the testers' community to obtain help, (ii) attendance at conferences (66 responses) for networking and meeting other professionals in the field, and (iii) talking with researchers (64 responses) of the area towards finding solutions to problems. However, no simple communication is established between industry and academy.

6 Discussion on the survey findings

6.1 Correlations

In this section, we present a triangulation constructed from the obtained results. We are looking for correlations among the analyzed data. Some exciting finds are discussed below.

6.1.1 Software testing technique used versus type of project

We analyzed the correlation between testing techniques and software projects, seeking to understand which testing techniques are commonly used to test companies' different types of systems. The results are shown in Figure 23, where we can see that for all types of projects, functional testing is the most used. The second testing technique most used is the error-based test, except for web systems and ERP systems, where the structural test takes second place. On average, the structural testing presented similar results to the error-based test and the last site, as the least used technique was the model-based test. Despite that, all testing techniques are used at some intensity level independently of the software project type.

6.1.2 Software testing technique used versus company size

We evaluated the relationship between companies' size and the applied testing technique to understand which factors influence the choice of the testing technique (Figure 24. We can

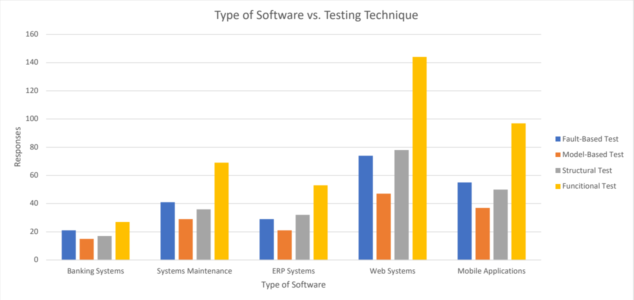

Figure 23. Relationship between type of software project and testing technique adopted by companies.

observe that functional testing is the most used technique, regardless of the company size. About other testing techniques, structural testing is more used in large companies while fault-based and model-based are few used techniques both in large and small companies.

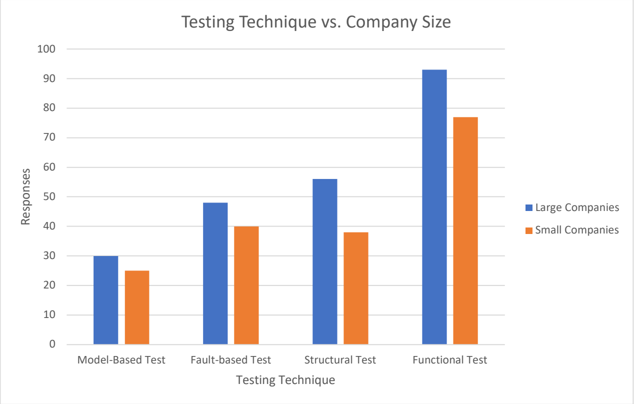

Figure 24. Relationship between company size and testing technique used in companies' projects.

6.2 Challenges

This study has summarized the main challenges indicated by the respondents regarding the execution of testing activities, which can contribute to the development of research in the software testing area (Figure 25).

- **Lack of importance and priority of the testing activity (71 responses):** the team involved in the project management does not value the correct execution of the testing activity and often neglects the activity establishing short deadlines for its development. Awareness of its importance should be raised by professionals in the technological area, and a possible solution would be the insertion of specific disciplines in technological undergraduate courses;
- **Lack of knowledge and training (51 responses):** respondents indicated some testers do not have the necessary knowledge to perform automation activities, since they are not familiar with programming, and companies do not provide them with specific training. The activity requires substantial expertise in programming.
- **Automation process (46 responses):** there is a need to automate the tests developed to improve efficiency in the testing activity. Automation is a challenge in testing practice because there are no clear guidelines that

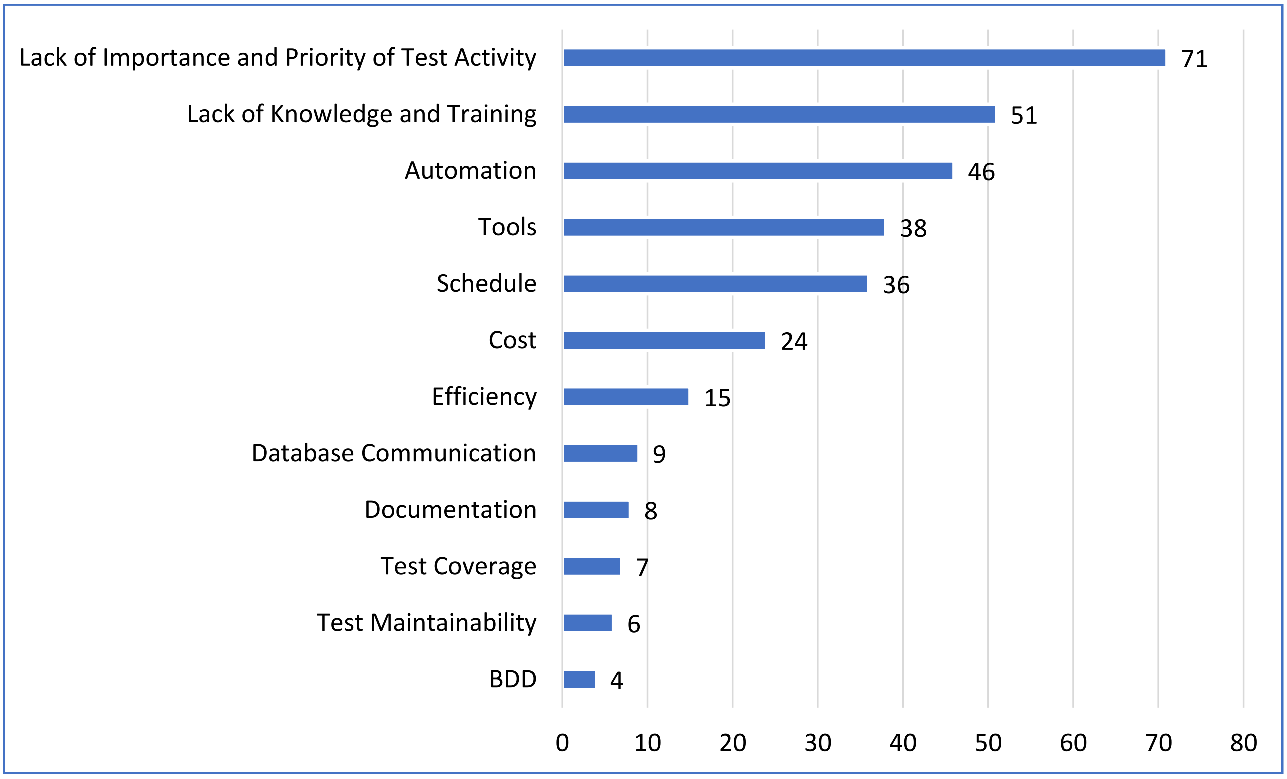

Figure 25. Challenges related to testing in software companies.

should be followed to help testers perform the automation process;

- **Tools (38 responses):** many testers use automated tools to support the automation process. The respondents reported difficulty using some tools and the challenge of finding one suitable for the activity. Consequently, several tools are used. The development of tools that offer a variety of functionalities for supporting the automation process and other testing activities is, therefore, fundamental;
- **Schedule (36 responses):** facing short deadlines dedicated to the testing activity within the project schedule, is a hard task; therefore, improvements in efficiency, such as activities and new proposals are required towards optimizing testing.

Other challenges indicated are (i) cost (24 responses) - new approaches must be developed for reducing the costs spent on testing activities, (ii) efficiency (15 responses) - it is related to the difficulty of performing the testing activity and achieving the best possible results, (iii) database communication (9 responses) - the generation of masses of data for use in automated tests requires improvements in the communication with the database during testing activities, (iv) documentation of testing activities (8 responses) - respondents reported difficulties in finding satisfactory documents to solve their problems of use of tools or new testing techniques, (v) test coverage (7 responses), which is a resource-intensive task, (vi) test maintainability (6 responses), which maintains the created test cases; their reuse in other programs is a topic to be further researched, and (vii) BDD (4 responses) - its adoption can be hampered by an incorrect implementation.

7 Threats to validity

Our results can potentially be affected by the methodology defined, and the threats are discussed towards guiding the interpretation.

1. *Conclusion validity:* regards the reliability and strength of the data provided by the respondents. To minimize it, we carefully summarized the data collected and excluded blank ones or those with incorrect information. The survey initially received 201 responses. To assure all respondents were aware of and worked with software testing, we asked a question to check whether they knew about software testing. If the answer was no, the survey was finished. 16 negative answers were given; therefore, they were excluded, thus leaving 185 responses to be analyzed.
2. *Internal validity:* regards influences that can affect the study to causality. To mitigate it, we followed the guidelines proposed by Kitchenham and Pfleeger (2008). Prior to designing the survey, we analyzed other relevant work, reviewed similar past surveys (mentioned in Section 4), reused some of their questions and created new ones, based on the specific objectives of our research.
3. *Construct validity:* concerns the generalization of the study results. Our scenario relates to this threat if we measure the real scenario of software testing practices in the Brazilian software industry. To minimize it, we categorized the data collected in each question for extracting relevant information and making assumptions about the topics addressed.
4. *External validity:* concerns the generalization of our re-

sults. The population and sample size of the respondents (Table 3) outlines the number of emails sent to the lists of companies accessed. Since 185 respondents out of 1062 companies listed contributed to the survey, a rough measure of the sample size equals 17.51% (185/1062). Our sampling rate has provided relevant results and contributed to our assumptions.

8 Conclusions and future work

This paper has reported the findings of a survey on software testing practices conducted in Brazilian software companies. Prior to its design, we intensively studied other surveys related to the topic. Our research objectives and questions were specified according to the GQM methodology.

The study presents, in detail, the results collected from the analyzed data, which enabled the identification of the software testing practices most used by testers and software quality professionals in the Brazilian companies. Testers select a technique according to the scope of the project. The most applied is functional and structural testing; the least used one is the mutation test, which was also pointed out as the technique most respondents had never applied. Selenium was indicated as the most used tool in testing activities. The challenges faced by testers were discussed and have raised interesting topics to be more deeply explored.

The need for surveys on software testing practices is clear, since they will help software companies improve both their testing strategies and the relationship between testing and the software quality of the products developed. Despite some limitations, our study can support this process.

As future work, we aim to portray a broad view of the industry as a whole and a more detailed picture of some companies. The research will be developed with more qualitative aspects involving formal interviews or focus groups, and companies will be visited to obtain more results for comparisons.

Acknowledgements

The authors acknowledge FAPESP (Sao Paulo Research Foundation), for the financial support under processes number 2018/10183-9 and 2019/06937-0, and PROPP/UFGD under SigProj project number 322855.1174.8276.1103 2019.

References

Abran, A., Moore, J. W., Bourque, P., Dupuis, R., and Tripp, L. L. (2004). *Guide to the software engineering body of knowledge: 2004 version SWEBOK*. IEEE Computer Society.

Anand, S., Burke, E. K., Chen, T. Y., Clark, J., Cohen, M. B., Grieskamp, W., Harman, M., Harrold, M. J., Mcminn, P., Bertolino, A., et al. (2013). An orchestrated survey of methodologies for automated software test case generation. *Journal of Systems and Software*, 86(8):1978–2001.

Bertolino, A. (2007). Software testing research: Achievements, challenges, dreams. In *2007 Future of Software Engineering*, pages 85–103. IEEE Computer Society.

Bourque, P., Fairley, R. E., et al. (2014). *Guide to the software engineering body of knowledge (SWEBOK (R)): Version 3.0*. IEEE Computer Society Press.

Briand, L. C., Differding, C. M., and Rombach, H. D. (1996). Practical guidelines for measurement-based process improvement. *Software Process: Improvement and Practice*, 2(4):253–280.

Caldiera, G., Basili, V. R., and Rombach, H. D. (1994). Goal question metric paradigm. *Encyclopedia of software engineering*, 1:528–532.

Copeland, L. (2004). *A practitioner's guide to software test design*. Artech House.

Delamaro, M., Jino, M., and Maldonado, J. (2017). *Introdução ao teste de software*. Elsevier Brasil.

Dias-Neto, A. C., Matalonga, S., Solari, M., Robiolo, G., and Travassos, G. H. (2017). Toward the characterization of software testing practices in south america: looking at brazil and uruguay. *Software Quality Journal*, 25(4):1145–1183.

Dustin, E., Rashka, J., and Paul, J. (1999). *Automated Software Testing: Introduction, Management, and Performance*. Addison-Wesley Longman Publishing Co., Inc., USA.

Engström, E. and Runeson, P. (2010). A qualitative survey of regression testing practices. In *International Conference on Product Focused Software Process Improvement*, pages 3–16. Springer.

Fink, A. (2003). *The survey handbook*, volume 1. Sage.

Garousi, V. and Varma, T. (2010). A replicated survey of software testing practices in the canadian province of alberta: What has changed from 2004 to 2009? *Journal of Systems and Software*, 83(11):2251–2262.

Garousi, V. and Zhi, J. (2013). A survey of software testing practices in canada. *Journal of Systems and Software*, 86(5):1354–1376.

Geras, A. M., Smith, M., and Miller, J. (2004). A survey of software testing practices in alberta. *Canadian Journal of Electrical and Computer Engineering*, 29(3):183–191.

Groves, L., Nickson, R., Reeve, G., Reeves, S., and Utting, M. (2000). A survey of software development practices in the new zealand software industry. In *Software Engineering Conference, 2000. Proceedings. 2000 Australian*, pages 189–201. IEEE.

Kassab, M., DeFranco, J., and Laplante, P. (2016). Software testing practices in industry: The state of the practice. *IEEE Software*.

Kasunic, M. (2005). Designing an effective survey. Technical report, Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst.

Kitchenham, B. A. and Pfleeger, S. L. (2008). Personal opinion surveys. In *Guide to advanced empirical software engineering*, pages 63–92. Springer.

Lee, J., Kang, S., and Lee, D. (2012). Survey on software testing practices. *IET software*, 6(3):275–282.

Molléri, J. S., Petersen, K., and Mendes, E. (2016). Survey guidelines in software engineering: An annotated review. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measure-*

ment, page 58. ACM.

Myers, G. J., Sandler, C., and Badgett, T. (2011). *The art of software testing*. John Wiley & Sons.

Ng, S., Murnane, T., Reed, K., Grant, D., and Chen, T. (2004). A preliminary survey on software testing practices in australia. In *Software Engineering Conference, 2004. Proceedings. 2004 Australian*, pages 116–125. IEEE.

Saglietti, F., Oster, N., and Pinte, F. (2008). White and grey-box verification and validation approaches for safety-and security-critical software systems. *Information security technical report*, 13(1):10–16.

Santos, I., C Filho, J. C., and Souza, S. R. (2020a). A survey on the practices of mobile application testing. In *2020 XLVI Latin American Computing Conference (CLEI)*, pages 232–241. IEEE.

Santos, I., Coutinho, E. F., and Souza, S. R. (2020b). Software testing ecosystems insights and research opportunities. In *Proceedings of the 34th Brazilian Symposium on Software Engineering*, pages 421–426.

Santos, I., Furlanetti, A. B., Melo, S. M., de Souza, P. S. L., Delamaro, M. E., and Souza, S. R. (2020c). Contributions to improve the combined selection of concurrent software testing techniques. In *Proceedings of the 5th Brazilian Symposium on Systematic and Automated Software Testing*, pages 69–78.

Santos, I., Melo, S. M., de Souza, P. S. L., and Souza, S. R. (2019). Testing techniques selection: A systematic mapping study. In *Proceedings of the XXXIII Brazilian Symposium on Software Engineering*, pages 347–356.

Santos, I., Melo, S. M., de Souza, P. S. L., and Souza, S. R. (2020d). Towards a unified catalog of attributes to guide industry in software testing technique selection. In *2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pages 398–407. IEEE.

Torchiano, M., Fernández, D. M., Travassos, G. H., and de Mello, R. M. (2017). Lessons learnt in conducting survey research. In *2017 IEEE/ACM 5th International Workshop on Conducting Empirical Studies in Industry (CESI)*, pages 33–39. IEEE.

Torkar, R. and Mankefors, S. (2003). A survey on testing and reuse. In *Software: Science, Technology and Engineering, 2003. SwSTE'03. Proceedings. IEEE International Conference on*, pages 164–173. IEEE.

Vegas, S. and Basili, V. (2005). A characterisation schema for software testing techniques. *Empirical Software Engineering*, 10(4):437–466.

Victor, M. and Upadhyay, N. (2011). Selection of software testing technique: A multi criteria decision making approach. In *International Conference on Computational Science, Engineering and Information Technology*, pages 453–462. Springer.

Wohlin, C., Aurum, A., Angelis, L., Phillips, L., Dittrich, Y., Gorschek, T., Grahn, H., Henningsson, K., Kagstrom, S., Low, G., et al. (2011). The success factors powering industry-academia collaboration. *IEEE software*, 29(2):67–73.