Journal of Software Engineering Research and Development, 2019, 7:6, doi: 10.5753/jserd.2019.472

Characterization of software testing practices: A replicated survey in Costa Rica

Christian Quesada-López [**Universidad de Costa Rica, Universidad Estatal a Distancia** | ***cristian.quesadalopez@ucr.ac.cr; cquesadal@uned.ac.cr***]
Erika Hernandez-Agüero [**Universidad Estatal a Distancia** | ***ehernandez@uned.ac.cr***]
Marcelo Jenkins [**Universidad de Costa Rica** | ***marcelo.jenkins@ucr.ac.cr***]

Abstract

Software testing is an essential activity in software development projects for delivering high quality products. In a previous study, we reported the results of a survey of software engineering practices in the Costa Rican industry. To make a more in-depth analysis of the specific software testing practices among practitioners, we replicated a previous survey conducted in South America. Our objective was to characterize the state of the practice based on practitioners' use and perceived importance of those practices. This survey evaluated 42 testing practices grouped in three categories: processes, activities, and tools. A total of 92 practitioners responded to the survey. The participants indicated that: (1) tasks for recording the results of tests, documentation of test procedures and cases, and re-execution of tests when the software is modified are useful and important for software testing practitioners. (2) Acceptance and system testing are the two most useful and important testing types. (3) Tools for recording defects and the effort to fix them (bug tracking) and the availability of a test database for reuse are useful and important. Regarding the use and implementation of practices, the participants stated that (4) Planning and designing of software testing before coding and evaluating the quality of test artifacts are not a regular practice. (5) There is a lack of measurement of defect density and test coverage in the industry; and (6) tools for automatic generation of test cases and for estimating testing effort are rarely used. This study gave us a first glance at the state of the practice in software testing in a thriving and very dynamic industry that currently employs most of our computer science professionals. The benefits are twofold: for academia, it provides us with a road map to revise our academic offer, and for practitioners, it provides them with a first set of data to benchmark their practices.

Keywords: *Software Testing, Industry Practices, Survey, Costa Rica, Replication, Empirical Software Engineering*

1 Introduction

Software testing is an essential activity in software development projects, for delivering high quality products, but it is a costly activity in the software development life cycle (Garousi and Zhi, 2013). Software testing represents, on average, around 35% of the total budget of a development project (Dias-Neto et al., 2017). Testing practices play a significant role in the development process, they represent a quality assurance strategy for the identification of defects in the software applications before its deployment (Juristo et al., 2004).

Software testing has been a focus of attention for the industry. For example, the International Software Testing Qualifications Board (ISTQB, https://www.istqb.org/) aims to continually improve and advance the software testing profession by defining and maintaining a Body of Knowledge that allows testers to be certified based on best practices, connecting the international software testing community, and encouraging research. ISTQB promotes the value of software testing as a profession to individuals and organizations and has performed studies to observe the perception of practitioners on testing. After the "2013 ISTQB Effectiveness Survey", in which they collected feedback on the impacts of testing certifications, in 2015 ISTQB conducted a worldwide survey on Software Testing Practices with 3,281 responses from testing practitioners from 89 countries. ISTQB survey reveals significant findings for the professional practice:

- The budgets assigned to testing are large and keep on growing and ranges between 11% and 40%.
- The agile methodologies are being adopted ahead of traditional ones that emphasize the need to have appropriate testing processes and techniques for Agile.
- The segregation of duties has become a standard practice where in 84% of the cases the test team does not report to develop.
- The test tools for defect tracking, test execution, test automation, test management, performance testing, and test design are widely adopted.
- Some level of test automation is a trending topic with a with 72% of adoption.
- Testing requires a wide range of skills and competencies.
- There are important career paths available for testers and test managers.
- The decision of when to stop testing is mainly based on requirements coverage.
- Exploratory testing is the most adopted test techniques.
- Performance, usability, and security are the top three non-functional testing activities.
- There are several improvement opportunities in testing practices such as test automation, test process, communication, and test techniques.

Afterward, the 2017-2018 ISTQB Worldwide Software Testing Practices Report collected more than 2,000 responses from 92 countries. It reported findings mostly in parallel with the previous survey and revealed the following: (1) main improvement areas in software testing were test automation, knowledge about test processes, and communication between development and testing. (2) The top five test design techniques are use case testing, exploratory testing, boundary value analysis, checklist-based, and error guessing. (3) Trending topics will be test automation, agile testing, and security testing. (4) New technologies that could affect testing are security, artificial intelligence, and big data. Finally, (5) non-testing skills expected are soft skills, business and domain knowledge, and business analysis skills.

Currently, there is a gap between knowledge in academia and the software testing practices used in industry (Dias-Neto et al., 2017). Moreover, there is a knowledge deficiency for testing topics in practice activities (Scatalon et al., 2018). Garousi and Felderer (2017) state that the level of joint industry-academia collaborations in Software Engineering is very low compared to the number of activities in each of the two communities. Comparing the focus areas of industry and academia in software testing, results show that the two groups are talking about quite different things. As an example, practitioners talk about test automation referring to automating the test execution phase and academics on automated approaches (mostly focused on test-case generation and test oracles) (Garousi and Felderer, 2017). Moreover, researchers tend to be more interested in theoretically challenging issues, but test engineers in practice are more looking for options to improve the effectiveness and efficiency of testing (Garousi and Felderer, 2017; Garousi et al., 2017).

Besides, there is a wide spectrum of testing practices conducted by different software teams (Garousi and Zhi, 2013) and a little evidence in the literature regarding the use and importance of such practices in industry (Dias-Neto et al., 2017). The characterization of testing practices used in industry can help professionals, researchers, and academics to better understand the challenges faced by the software engineering profession (Garousi and Zhi, 2013).

To characterize testing practices in the software industry, a large number of surveys have been conducted in different countries. Garousi and Zhi (2013), and Dias-Neto et al. (2017) summarized previous surveys on software testing practices. In particular, Dias-Neto et al. (2017) identified surveys conducted to characterize the adoption of software testing practices, tools, and methods.

The earliest identified surveys to characterize aspects of the testing process were from the United States of America in (Beck and Perkins, 1983; Gelperin and Hetzel, 1988; Torkar and Mankefors, 2003). After that, other surveys were identity in United States (Wojcicki and Strooper, 2006; Kassab et al., 2017; Kassab, 2018). A set of replications surveying testing practices in Canada was conducted from 2004 to 2017 (Geras et al., 2004; Garousi and Varma, 2010; Garousi and Zhi, 2013; Garousi et al., 2017) and some studies surveying testing practices in South America was conducted from 2006 to 2018 (Dias-Neto et al., 2006; De Greca et al., 2015; Dias-Neto et al., 2017; Robiolo et al., 2017; Scatalon et al., 2018). Four more surveys were conducted in Australia and New Zealand between 2004 and 2012 (Ng et al., 2004; Chan et al., 2005; Sung and Paynter, 2006; Wojcicki and Strooper, 2006; Kirk and Tempero, 2012).

Additionally, other studies surveying different aspects related to testing practices were conducted in Finland (Taipale et al., 2005, 2006; Kasurinen et al., 2010; Pfahl et al., 2014; Smolander et al., 2016; Hynninen et al., 2018; Raulamo-Jurvanen et al., 2019), Spain (Fernández-Sanz, 2005; Fernández-Sanz et al., 2009), Sweden (Runeson, 2006; Grindal et al., 2006; Engström and Runeson, 2010), Korea (Park et al., 2008; Yli-Huumo et al., 2014), Netherlands (Vonken et al., 2012), Norway (Deak et al., 2013; Deak and Stålhane, 2013), Belgium (Pérez et al., 2013), Turkey (Garousi et al., 2015), Sri Lanka (Vasanthapriyan, 2018), and Bangladesh (Bhuiyan et al., 2018). Finally, other studies surveying different aspects related to testing practices were conducted in different countries (Chan et al., 2005; Causevic et al., 2010; Rafi et al., 2012; Lee et al., 2012; Greiler et al., 2012; Pham et al., 2013; Daka and Fraser, 2014; Kanij et al., 2014; Deak, 2014; Ghazi et al., 2015; Kochhar et al., 2015; Lima and Faria, 2016; Rodrigues and Dias-Neto, 2016; Garousi et al., 2017; Kochhar et al., 2019).

In Costa Rica, previous surveys had been conducted to characterize software engineering practices. In our previous work (Quesada-López and Jenkins, 2017, 2018), we replicated a survey based on (Garousi et al., 2015, 2016) where we identify the most common practices, methods, and tools in professional practice and their related challenges. Moreover, we conducted a cross-factor correlation analysis of development and testing engineering practices versus practitioner demographics. In (Aymerich et al., 2018), the authors conducted a survey on development practices based on the HELENA study (Kuhrmann et al., 2017). They studied development approaches, practices, and methods in the industry. To analyze the specific software testing practices among practitioners in our country, we replicated previous surveys conducted in South America (Dias-Neto et al., 2006; De Greca et al., 2015; Dias-Neto et al., 2017; Robiolo et al., 2017).

Further replications in different countries are still needed to allow the comparison of industry trends in software testing practices (Garousi and Zhi, 2013; Dias-Neto et al., 2017). The results of these surveys can support evidence on testing practices in the software engineering community (Garousi and Zhi, 2013).

The objective of our study was to characterize a set of software testing practices with respect to their use and importance from the point of view of practitioners of software organizations in Costa Rica. In this work, we replicated the previously surveys in (Dias-Neto et al., 2006; De Greca et al., 2015; Dias-Neto et al., 2017; Robiolo et al., 2017) with 92 practitioners from our country. As stated in (Dias-Neto et al., 2017), we were interested in understanding the testing practitioners' use and perceived importance of software testing practices. In addition, we wanted to compare the results of our study with the results of the previous surveys. Thus, to facilitate the comparison between previous studies and this replication, we used the same questionnaire used in (Dias-Neto et al., 2017).

Previously, we had researched the software engineering practices of the industry in Costa Rica (Quesada-López and Jenkins, 2017, 2018). In this paper, we extend our previous study on software testing practices (Quesada-López et al., 2019) by extending the analysis performed. Besides, we conducted a literature search to identify past surveys on software testing practices in the industry. We describe the survey's planning, design, execution, analysis of the collected data, and the comparison with previous surveys conducted in Brazil, Uruguay, and Argentina to discuss the use and importance of software testing practices. Finally, to get feedback about the significance and usefulness of the survey results from the practitioners' perspective, we made two presentations of the study to different groups of professionals.

This study gave us a first glance at the state of the practice in software testing in a thriving and very dynamic industry that currently employs most of our computer science professionals. The benefits are twofold: for academia, it provides us with a road map to revise our academic offering, and for practitioners, it provides a baseline to benchmark their current practices.

The paper is structured as follows: Section 2 presents the related work. Section 3 describes the survey replication process. Section 4 analysis the results of the survey. Finally, Section 6 outlines our conclusions and future work.

2 Related work

Several survey studies have been conducted on the subject of software testing practices in different countries and scales (Garousi and Zhi, 2013). This section summarizes identified past surveys on software testing practices in the industry. These studies mainly aim to characterize the state of the practice in the software testing industry, identifying trends and opportunities for improvement and training (Dias-Neto et al., 2017).

To identify past surveys on software testing practices in the industry, we conducted a literature search. First, we conducted an exploratory search using Scopus and using the search string "TITLE-ABS-KEY(("software") AND ("testing practices" OR "quality assurance practices") AND ("survey" OR "questionnaire"))".

Additionally, we applied the snowballing technique (Wohlin, 2014) on two surveys previously published (Garousi and Zhi, 2013; Dias-Neto et al., 2017). Their cited references were searched using Google Scholar.

The inclusion criteria included only papers describing software testing surveys based on titles, keywords, abstracts, and analysis. The list includes papers on software engineering practices that report results on specific software testing practices.

Table 1 briefly summarizes the surveys on testing practices. The paper reference, scale and region (or target community), target audience, number of respondents, and survey goal and focus area are listed. This table was based on Garousi and Zhi (2013); Dias-Neto et al. (2017) and updated with identified surveys in our search. In Table 1, papers reported in Garousi and Zhi (2013) were marked with (*) and papers reported in Dias-Neto et al. (2017) were marked with (**). Papers in both studies were marked with (***). The following reports were excluded because their research goal and method were not comparable to the others surveys (Andersson and Runeson, 2002; Runeson et al., 2003).

The studies attempt to identify and characterize different software testing practices, processes, tools, and methods in different contexts. Many surveys were conducted since 2006, denoting the interest in surveys on software testing industry. In the last decade, one survey was published in 2009, four surveys were published in 2010, five surveys in 2012, the same quantity in 2013 and 2014, four surveys were published in 2015, three surveys in 2016, five surveys in 2017 and 2018, and finally, three surveys were published in 2019, as listed in Table 1. The main surveys' goals reported were:

- To characterize the adoption of software testing practices, processes, tools, and methods in different contexts (Beck and Perkins, 1983; Gelperin and Hetzel, 1988; Torkar and Mankefors, 2003; Geras et al., 2004; Ng et al., 2004; Chan et al., 2005; Wojcicki and Strooper, 2006; Dias-Neto et al., 2006; Kasurinen et al., 2010; Garousi and Varma, 2010; Kirk and Tempero, 2012; Garousi and Zhi, 2013; Pérez et al., 2013; Daka and Fraser, 2014; Yli-Huumo et al., 2014; De Greca et al., 2015; Garousi et al., 2015; Ghazi et al., 2015; Smolander et al., 2016; Kassab et al., 2017; Quesada-López and Jenkins, 2017; Dias-Neto et al., 2017; Robiolo et al., 2017; Hynninen et al., 2018; Vasanthapriyan, 2018).
- To characterize the strengths and issues of software testing, and the opportunities for the improvement of software testing, including the critical factors of success in different aspects of software testing (Runeson, 2006; Engström and Runeson, 2010; Causevic et al., 2010; Rafi et al., 2012; Lee et al., 2012; Greiler et al., 2012; Pfahl et al., 2014; Kochhar et al., 2015; Rodrigues and Dias-Neto, 2016; Bhuiyan et al., 2018; Kochhar et al., 2019).
- To analyze what factors may influence the selection of software testing practices (Fernández-Sanz et al., 2009; Greiler et al., 2012; Deak et al., 2013; Pham et al., 2013; Pérez et al., 2013; Deak and Stålhane, 2013; Pfahl et al., 2014; Deak, 2014; Kochhar et al., 2015; Lima and Faria, 2016; Kochhar et al., 2019; Raulamo-Jurvanen et al., 2019).
- To analyze software testing practices and the level of maturity in the industry (Fernández-Sanz, 2005; Grindal et al., 2006; Park et al., 2008).
- To compare practitioners' software testing practices and the state of art (Sung and Paynter, 2006; Causevic et al., 2010; Engström and Runeson, 2010; Vonken et al., 2012; Rafi et al., 2012; Scatalon et al., 2018).
- To characterize training needs and skills needed in software testing (Ng et al., 2004; Chan et al., 2005; Kanij et al., 2014; Vasanthapriyan, 2018).
- To identify research directions in software testing (Taipale et al., 2005, 2006; Smolander et al., 2016; Garousi et al., 2017).

Studies reported the gap between software testing state of the art and state of the practice (Ng et al., 2004; Dias-Neto et al., 2006; Sung and Paynter, 2006; Causevic et al., 2010; Engström and Runeson, 2010; Rafi et al., 2012; Lee et al., 2012; Yli-Huumo et al., 2014; Garousi et al., 2017; Scatalon et al., 2018; Vasanthapriyan, 2018; Scatalon et al., 2018).

Software testing is still reported as a time consuming and expensive phase in software development (Beck and Perkins, 1983; Ng et al., 2004; Dias-Neto et al., 2006). The automation of software testing has continued its growth and there are opportunities for automated software testing research (Ghazi et al., 2015; Hynninen et al., 2018; Kochhar et al., 2019; Raulamo-Jurvanen et al., 2019).

3 Replication process

In the following subsections, we provide details about the methodology for conducting the replication.

Replication studies are beneficial to evaluate the validity of prior study findings. Successful replications increase the validity and reliability of the outcomes observed in the original study and are an essential part of the experimental paradigm to produce generalizable knowledge (Carver et al., 2014). Combined results from a family of replications are interesting because all studies are related and could investigate related questions. The aggregation of replication results will be useful for software engineers to draw conclusions and consolidate the findings (Carver, 2010; Juristo and Gómez, 2010; Carver et al., 2014). A close replication study attempts to recreate the known conditions of the original study and is very similar to the original study. Close replications are often used to establish whether the original outcomes are repeatable (Lindsay and Ehrenberg, 1993).

Our study is an external replication of four previously conducted surveys in South America (Dias-Neto et al., 2006; De Greca et al., 2015; Dias-Neto et al., 2017; Robiolo et al., 2017). Dias-Neto et al. (2006) analyze the answers of 36 practitioners from 13 Brazilian organizations to identify the software testing practices used by the organizations and its importance. Greca et al. (2015) replicated the original survey with 18 practitioners in Argentina. Dias-Neto et al. (2017) conducted the same survey in Brazil and Uruguay with 150 practitioners. They surveyed different companies from Southern/Brazil (56 participants), Northern/Brazil (50 participants) and Uruguay (44 participants). Robiolo et al. (2017) surveyed 25 practitioners from 25 organizations of the public sector.

In this study, we reported the responses from 92 practitioners from Costa Rica. The study includes a detailed analysis of the data collected, and its comparison with previous studies, in accordance with the recommendations and guidelines in (Carver, 2010; Carver et al., 2014). This study is descriptive (Linåker et al., 2015) and is intended to compare and extend previous results (Carver et al., 2014), highlighting the similarities and differences in the use and importance of testing practices in different countries. The authors of the original study did not take part in the replication process. However, in our replication, we reused the survey goal, research questions, questionnaire, and analysis procedure reported in (Dias-Neto et al., 2017; Robiolo et al., 2017).

3.1 Goal and research questions

The objective of the study formulated using the Goal, Question, Metric (GQM) approach (Basili et al., 1994) was to characterize testing practices based on the practitioners' use and perceived importance in the context of software organizations in Costa Rica. The survey evaluated 42 testing practices grouped in three categories: processes, activities, and tools. We studied the following research questions:

- RQ1: What are the software testing practices used by practitioners in their organizations?
- RQ2: What are the most important software testing practices according to the opinion of testing practitioners?

3.2 Survey design

To address the study's goal and research questions, we conducted a survey to gather the opinions from practitioners.

3.2.1 Target population and sampling

The target population is the practitioners applying testing practices in software organizations in Costa Rica. The practitioners were sampled by convenience. They were contacted through the University of Costa Rica and the State Distance University, two of the most important public universities in our country. E-mail distribution lists were used to support the recruitment of participants.

3.2.2 Instruments used to collect data

We applied the questionnaire designed in (Dias-Neto et al., 2017) to collect the information. The instrument was divided into three parts: (1) profile and demographics, (2) the use of testing processes, activities and tools; and (3) perceived importance of testing processes, activities, and tools. The instrument evaluated 42 testing practices grouped in three categories: testing processes (practices related to the adopted test processes in the software organization), testing activities (practices concerned with the procedures performed during the software testing), and testing tools (practices concerned with tools supporting the software testing). We used the Spanish version of the instrument. In order to validate the questionnaire (concepts, language, and practices), we conducted five survey pilots. Table 2 details the list of questions of the instrument.

The participants were asked to fill out the job position, experience in software testing, academic degree, certifications in testing, development methodology, programming language expertise, software platform used for development, company's size, and quality team configuration.

Participants were asked to fill the entire questionnaire with the 42 testing practices according to the use level in their current organization and the perceived importance of a testing practice. Dias-Neto et al. (2017; 2006) defined a five point Likert scale to express the gradual increase in the level of use and importance of a testing practice, as shown in Table 3. As in the previous study, each practitioner answered only one option for the level of use and importance for each software testing practice.

Table 1. Summary of previous surveys on software testing practices.

Paper reference	Scale/region	Target audience	Number of respondents	Goal/focus area
Beck and Perkins (1983)	Dallas-Fort Worth, USA	Computer users	63	To analyze the usage of software engineering techniques, tools, and methods. They analyzed testing and validation activities in the software life cycle (*).
Gelperin and Hetzel (1988)	Washington, USA	Not reported	Not reported	To characterize major test process models, methodologies, and describe some of the changes associated with testing growth (**).
Torkar and Mankefors (2003)	USA, Sweden	Software development organization	91	To explain to what extent software testing had been used when reusing software components (**).
Geras et al. (2004)	Alberta, Canada,	Software development organization	60	To characterize test practices and software quality assurance techniques (***).
Ng et al. (2004)	Australia	Senior software practitioners	65	To determine testing techniques, tools, metrics, standards, and whether the training courses in software testing adequately cover the testing methodologies and skills required (**).
Fernández-Sanz (2005)	Spain	Professional practitioners	102	To analyze testing practices and the level of maturity in testing.
Taipale et al. (2005)	Finland	Software testing researchers	10	To identify research directions in software testing (**).
Chan et al. (2005)	5 countries	Software testing practitioners	34	To characterize software testing practices, and the levels of software testing education and training (**).
Wojcicki and Strooper (2006)	USA, Australia	List at cs.oswego.edu and IBM	35	To analyze the state of practice of verification and validation technology, the decision process for use, and cost-effectiveness for concurrent programs (**).
Runeson (2006)	Sweden	Software developers	15	To characterize the strengths and issues of unit testing (**).
Grindal et al. (2006)	Sweden	Not reported	12	To characterize organizations' testing maturity (**).
Sung and Paynter (2006)	New Zealand	Software testers	62	To compare software testing practices with the authors' software testing framework (**).
Dias-Neto et al. (2006)	Brazil	Software developers	36	To characterize the state of the practice of software testing in Brazil (**).
Taipale et al. (2006)	Finland	Industry specialists	40	To determine the current situation and improvement needs in software testing.
Park et al. (2008)	Korea	Software professionals in defense industry	38	To identify test maturity, testing practices, and characteristics of software development in the Korean defense industry.
Fernández-Sanz et al. (2009)	Spain	Software professionals	127	To analyze what factors may influence testing practices.
Engström and Runeson (2010)	Sweden	Software developers	32	To characterize the gap between the state of the art and practice of regression testing practices.
Kasurinen et al. (2010)	Finland	Software Testers and Test Managers	31	To identify the state of the practice on software test automation (**).
Causevic et al. (2010)	Not Reported	Researchers	83	To identify obstacles between the available (state-of-the art) and preferred (state-of-the-practice) practices by software testing practitioners (**).
Garousi and Varma (2010)	Alberta, Canada	Software developers	53	To replicate Geras et al. (2004) on software testing techniques and analyze possible changes (***).
Rafi et al. (2012)	Not reported	Software developers	115	To characterize the benefits and limitations of software testing automation (**).

Continued on next page

Table 1 – continued from previous page

Paper reference	Scale/region	Target audience	Number of respondents	Goal/focus area
Lee et al. (2012)	Not Reported	Executives	33	To identify the current practices and opportunities for the improvement of software testing tools and methods (**).
Greiler et al. (2012)	Not reported	EclipseCon participants	151	To discover how testing is performed, why testing is performed in a certain way and what test-related issues the community is facing (**).
Kirk and Tempero (2012)	New Zealand	Software developers	195	To understand what practices are used in software testing (***).
Vonken et al. (2012)	Netherlands	Development organizations	99	To determine whether there is a gap between the current state-of-the-practice and state-of-the-art in software engineering (*).
Deak et al. (2013)	Norway	Computing students	33	To identify the interest and desire to work in software testing among engineering and computer science students (**).
Deak and Stålhane (2013)	Norway	Not reported	23	To characterize the factors that can influence the creation of a software testing department or the investment in software testing personnel (**).
Garousi and Zhi (2013)	Canada	Software developers	246	To characterize Canadian testing practices (***).
Pham et al. (2013)	Not reported	Software developers of GitHub	569	To characterize how the testing behavior is influenced by the peculiarities of social coding environments (**).
Pérez et al. (2013)	Belgium	Development professionals	63	To assess the state of the practice in software quality with respect to software quality, and how these practices vary across companies.
Pfahl et al. (2014)	Finland and Estonia	Software Developers	61	To study how software engineers understand and apply the principles of exploratory testing, as well as the specific advantages and difficulties they experience (***).
Daka and Fraser (2014)	29 countries	Software Developers	246	To characterize how software developers use unit testing techniques (**).
Kanij et al. (2014)	22 countries	Software testers	104	To characterize skills of software testers affecting software testing (**).
Deak (2014)	Not reported	Software testers	26	To characterize the impact of the development methodology on testers motivation (**).
Yli-Huumo et al. (2014)	South Korea	Software development professionals	34 companies	To explore software development methods and quality assurance practices used by software industry.
De Greca et al. (2015)	Argentina	Software developers	18	To characterize the state of the practice in software testing in Argentina, a replication of Dias-Neto et al. (2006) (**).
Garousi et al. (2015)	Turkey	Software professionals	202	To characterize techniques, tools and metrics used by practitioners and the challenges faced. They included the analysis of the types of software testing practices, the latest techniques, tools, and metrics used and the challenges faced by practitioners (**).
Ghazi et al. (2015)	Not reported	Practitioners from LinkedIn and Yahoo Groups	27	To explore the testing of heterogeneous systems with respect to the usage and perceived usefulness of testing techniques used for heterogeneous systems from the point of view of industry practitioners.
Kochhar et al. (2015)	Not reported	Software developers in GitHub and Microsoft	210	To understand the common testing tools used by Android developers as well as the challenges faced by them when they test their apps.

Continued on next page

Table 1 – continued from previous page

Paper reference	Scale/region	Target audience	Number of respondents	Goal/focus area
Lima and Faria (2016)	Portugal	Software testing professionals	147	To assess the relevance of distributed and heterogeneous systems in software testing practice, the features to be tested, the test automation and tools, and desired features in test automation.
Rodrigues and Dias-Neto (2016)	Not reported	Software testing researchers and practitioners	33	To evaluate the critical factors of success in software test automation life cycle.
Smolander et al. (2016)	Finland	Software industry specialists	55	To understand the current situation and improvement needs in software test automation.
Kassab et al. (2017)	Penn State Great Valley, USA. LinkedIn	Professionals	67	To examined how software professionals used testing.
Quesada-López and Jenkins (2017)	Costa Rica	Software practitioners	278	To characterize engineering practices including the analysis of the software testing practices, a replication of Garousi et al. (2015).
Dias-Neto et al. (2017)	Brazil and Uruguay	Software testing practitioners	150	To understand the perception of practitioners regarding the use and importance of software testing practices, a replication of Dias-Neto et al. (2006); De Greca et al. (2015).
Robiolo et al. (2017)	Argentina	Software professionals in Public sector	25 organizations	To analyze use and importance of software testing practices, a replication of Dias-Neto et al. (2006); De Greca et al. (2015); Dias-Neto et al. (2017).
Garousi et al. (2017)	Canada, Turkey, Denmark, Austria, Germany	Practitioners	105	To characterize challenges and research topics that industry wants to suggest to software testing researchers.
Hynninen et al. (2018)	Finland	Industry practitioners	33	To explore industry practices concerning software testing and to assess how they test their products and what process models they follow, a continuation study of Taipale et al. (2006); Kasurinen et al. (2010).
Kassab (2018)	Not reported	Software professionals	72	To discover the actual practices for software testing and quality assurance activities for software in safety-critical systems.
Bhuiyan et al. (2018)	Bangladesh	IT personnel	47 organizations	To identify the challenges along with the practices of software quality assurance and testing.
Scatalon et al. (2018)	Brazil	Software professionals	90	To identify knowledge gaps in software testing between undergraduate courses and what professionals actually applied in industry after graduating.
Vasanthapriyan (2018)	Sri Lanka	Software development professionals	152 from 3 software companies	To determine software testing practices, testing methodologies and techniques, automated tools, testing metrics, testing training and academic collaboration with software industry.
Kochhar et al. (2019)	27 countries	Software practitioners	261	To investigate what make good test cases and to describe characteristics of good test cases and testing practices.
Raulamo-Jurvanen et al. (2019)	Finland	Testing professionals	89	To study how software practitioners evaluate testing tools.
This study (Quesada-López et al., 2019)	Costa Rica	Software practitioners	92	To characterize the state of the practice based on the perception of practitioners on the use and importance of software testing practices, a replication of Dias-Neto et al. (2006); De Greca et al. (2015); Dias-Neto et al. (2017); Robiolo et al. (2017).

3.2.3 Data analysis

For each testing practice, we collected the use and importance level based on the opinions of the professionals. The equations were based on Dias-Neto et al. (2017).

First, the responses of the professionals were differentiated by assigning a weight for each participant according to their experience, academic degree, and certifications on testing (Eq. 1). Second, we multiplied each answer by the weight of the participant and computed the total value for a testing practice (Eq. 2). Finally, we obtained a normalized value for the levels of use and importance that oscillates between 0% and 100% (Eq. 3). We applied the following formulas:

$$W(i) = \frac{DT(i)}{MdDT} + \frac{TT(i)}{MdTT} + f(i) + g(i) + h(i) \quad (1)$$

Where: $W(i)$ is the total weight for participant i. $DT(i)$ is the number of years of experience for participant i in software development. $TT(i)$ is the number of years of experience for participant i in software testing. $MdDT$ and $MdTT$ are the median of DT and TT. $f(i)$ is the highest academic degree for participant i (0-High school, 1-Undergraduate, 2-Specialization, 3-Master, 4-Ph.D). $g(i)$ is the self-assigned expertise level by the participant i (0-None, 1-Low, 2-Medium, 3-High, 4-Excellent). h(i) is the number of testing certifications reported by the participant i.

$$T(j) = \sum_{i=1}^{N} (Answer(i,j) * W(i)) \quad (2)$$

Where: $T(j)$ is the total value obtained for use and importance regarding the testing practice j. $Answer(i,j)$ is the answer value (1–5) relating to the use and importance by the participant i for the testing practice j.

$$N(j) = \frac{T(j)}{\sum_{i=1}^{N} W(i) * 5)} \quad (3)$$

Where: $N(j)$ is the normalized value for use and importance of testing practice j and $\sum_{i=1}^{N} W(i) * 5)$ is the maximum possible value for testing practice j.

For each testing practice, the use and importance were analyzed and compared with previous studies, and the correlation between use and importance perceived was evaluated. For this study, we replicated the analysis proposed in (Dias-Neto et al., 2017). The most used/important software testing practices, the differences between regions, and the difference between the levels of use and importance perceived by practitioners were analyzed. Finally, the existence of a significant correlation between the levels of use and importance for each evaluated practice was tested.

3.3 Survey execution

The electronic questionnaire was implemented using LimeSurvey (www.limesurvey.org) and it was available in a Survey Server at the University of Costa Rica for a period of two months, from September to October 2018. Participants were asked to complete the survey online. All participants were invited to participate anonymously and voluntarily by email. We sent e-mail invitations directly to the professionals through contact lists of the universities.

Practitioners could withdraw at any time, and only summarized and aggregated information were published. Similar to experiences in previous studies (Quesada-López and Jenkins, 2017, 2018), some participants leave questions unanswered and others leave the questionnaire without completing it. Only the completed answers were considered for the analysis of results. After data pre-processing, the responses of 92 professionals were analyzed.

3.4 Threats to Validity

This work is subject to the threats to the validity reported for this type of studies including previous replications and the results must be interpreted carefully. We discuss the validity concerns based on Wohlin et al. (2012) classification.

3.4.1 Internal validity

This threat is related to the quantity and representativeness of the sample. The practitioners were sampled by convenience, reported as common practice for survey studies in software engineering (Molléri et al., 2016; Ghazi et al., 2017), and in previous surveys listed in Section 2. Besides, the survey could not necessarily represent all the Costa Rican industry. Although we achieved a relatively high number of respondents compared with previous surveys (Dias-Neto et al., 2017; Robiolo et al., 2017), it was not possible to evaluate the representativeness of the sample. We were not able to obtain a reliable estimation of the total number of practitioners in the software industry of Costa Rica. Our participants were mainly invited through the Universidad Estatal a Distancia and Universidad de Costa Rica network and partners in Costa Rican software development organizations. Many practitioners out of our contact were not probably properly represented in the survey sample. Moreover, we were informed that some practitioners working in transnational software companies could not answer the questionnaire for confidentiality issues with their companies. The original testing practices lists in the original study were not modified to allow the replication. The original practices could be outdated from the current state of the art and practice. Moreover, some testing practices in Costa Rica's context could be missed or omitted. First, we believe that the set of practices is still representative in the testing research field (Dias-Neto et al., 2017). Second, we conducted five survey pilots with professionals in Costa Rica to validate the questionnaire (concepts, language, and practices).

3.4.2 Construct validity

The testing practices lists were based on a previous survey instrument (Dias-Neto et al., 2017, 2006). The analysis of the levels of use and importance has already been used in the evaluation of the performance of organizations. We counted the votes for each question and then made statistical analysis. We used the weight function based on Dias-Neto et al. (2017) to compare the results across studies. The weight function

Table 2. Survey Questionnaire.

Id	Question
D01	Job position
D02	Experience in software testing
D03	Academic degree
D04	Certifications in testing
D05	Development methodology
D06	Programming language expertise
D07	Software platform used for development
D08	Company's size
D09	Quality team configuration
P01	Documentation of test plan
P02	Documentation of test procedures and cases
P03	Recording the results of test execution
P04	Measurement and analysis of the test coverage
P05	Use of methodology or process
P06	Analysis of identified defects
P07	Identification and use of risks for planning and executing software tests
P08	Planning/Designing of testing before coding
P09	Monitoring adherence to the test process
P10	Re-execution of tests when the software is modified
P11	Evaluation of the quality of test artifacts
P12	Setting a priori criteria to stop the testing
P13	Reporting evaluation of a test round
A01	Definition of a responsible professional or team
A02	Application of unit tests
A03	Application of integration tests
A04	Application of system tests
A05	Application of acceptance tests
A06	Application of regression tests
A07	Application of exploratory tests
A08	Application of performance tests
A09	Application of security tests
A10	Registration of the time spent on testing
A11	Measurement of the effort/cost of testing
A12	Storage of records (log) of the executed tests
A13	Measurement of the defect density
A14	Conducting training on software testing
A15	Separation of testing and development activities
A16	Storage of test data for future use
A17	Analysis of faults patterns (trend)
A18	Availability of human resources allocated full time for testing
A19	Selection of test techniques according to the project's features
T01	Availability of a test database for reuse
T02	Use of tools for automatic execution of test procedures or cases
T03	Use of tools for automatic generation of test procedures or cases
T04	Use of test management tools to track and record
T05	Use of tools to estimate test effort and/or schedule
T06	Use of test management tools to enact activities and artifacts
T07	Use of tools for recording defects and the effort to fix them (bug tracking)
T08	Use of coverage measurement tools
T09	Continuous integration tools for automated tests
T10	Selection of test tools according to project characteristics
	D: Demographics. P: Testing processes. A: Testing activities. T: Testing tools.

Table 3. Level of use and importance.

L	Level of use	L	Level of importance
1	Not Applied: the practice is outside the scope of the organization's software projects.	1	Not important: the practice is not necessary for software projects.
2	Not used: the practice is within the scope of the organization, but it is not used in any software project.	2	Low value: the practice has low importance to use in software projects.
3	Infrequent use: the practice is not frequently used in the organization's software projects.	3	Limited value: the practice can be adequate to use in software projects.
4	Common use: the practice is used in most of the organization's software projects.	4	Significant value: the practice is recommended to use in software projects.
5	Standard use: the practice is used in all organization's software projects.	5	Essential value: the practice must be used in all software projects.
	L: Likert Scale.		

should be carefully analyzed to interpret the results. The analysis showed differences in the levels of use and importance of software testing practices. The characteristics of the organizations could affect these results. We informed participants of the survey that we will not collect any personal information so that professionals will remain anonymous.

3.4.3 Conclusion validity

The analysis procedure to obtain the level of use and importance according to the characteristics of each participant was based on previous surveys (Dias-Neto et al., 2017, 2006). The analysis procedure is a weighted average, where the weight function is based on qualitative aspects representing each subject (Dias-Neto et al., 2017). The model of use and importance was based on a previous empirical evaluation of the software practices (Dias-Neto et al., 2006). The trade-off of using this type of analysis is that the information from the extremes can be lost (Dias-Neto et al., 2017). All conclusions in this study are traceable to data.

3.4.4 External validity

The survey reflects the practitioners' interpretation of importance and use. The answers could not necessarily represent the reality of testing practices and could reflect subjectivity. Aspects such as self-awareness and difference of training of the participants could influence responses. The results show a correlation between the levels of use and importance. It could indicate that practitioners find those practices usable and important, but they could not distinguish between the use and importance or they see no value in the difference (Dias-Neto et al., 2017). In this study, we analyzed correlations between testing practices and we did not intend to establish any causal relationship.

4 Analysis of results

4.1 Demographics of the participants

In this survey, 92 complete answers were analyzed. Our participants could indicate more than one job position: 54% (50) of the practitioners reported one position, 23% (21) two positions, 8% (14) reported 3 and 4 positions, and 7% (7) reported up to 7 positions.

Table 4 presents the quantity (Q) and the percentage (%) of participants per position and company's size (S1: less than 10 employees, S2: 10-49 employees, S3: 50-100 employees, S4: more than 100 employees). Participants claimed to be mostly project managers (18%), analysts (17%), developers (16%), and quality managers (14%). In addition, participants reported being software engineers (9%), test analysts (8%), testers (8%), quality engineers (6%), and software architects (3%). Around 36% of participants are working on quality/testing. However, 32% (29) of the participants reported that both development and quality teams perform testing activities, 34% (31) reported that only quality teams perform testing, and 26% (24) reported that the development teams perform testing activities.

With respect to organizations size, 50% (46) of participants work in organizations with more than 100 employees, 16% (15) in organizations with 50-100 employees, 22% (20) work in organizations with 10-49 employees, and 12% (11) in organizations with less than 10 employees.

Table 4. Participants per position and company's size.

Position	Q	%	S1	S2	S3	S4
Project Man.	32	18	7	8	3	14
Analyst	31	17	4	6	6	15
Test Analyst	15	8	1	4	1	9
Architect	6	3	-	-	-	6
Quality Man.	14	8	1	2	2	9
Test Leader	10	6	1	3	-	6
Developer	29	16	3	5	5	16
Tester	15	8	2	4	-	9
Quality Eng.	11	6	1	1	2	7
Software Eng.	16	9	2	6	1	7
Total	92	100	11	20	15	46
		%	12	22	16	50

Participants reported on average, 11.5 years of experience in the software industry, and 5.5 years of experience in software quality and testing. Only 20% (18) of the participants hold a software testing certification. Some 15% (14) of practitioners are ISTQB Certified Testers, 3% (3) are Certified Test

Figure 1. Distribution of respondents' weight.

Manager (CTM), and 1% (1) is a Certified Software Quality Engineer (CSQE).

Participants reported the level of experience in testing, 33% (30) of the participants indicated a medium level of experience, 27% (25) indicated a high level, 21% (19) indicated a low level, 15% (14) an excellent level, and 4% (4) indicated no experience in testing.

Finally, participants reported their academic degree, 49% (45) hold a university degree, 36 (33%) a master's degree, 14% (13) have a technical specialization, and only 1% (1) holds a Ph.D.

In total, 59% (54) of the practitioners claim to apply agile methodologies, 26% (24) traditional methodologies and 15% (14) use a hybrid development methodology. The most used programming languages are .Net in C# and Visual Basic (35%), Java (24%), C/C++ (11%), PHP (9%), and Python (9%).

4.1.1 Participants' influence

Dias-Neto et al. (2017) observed that some participants could influence the results of the testing practices with their answers (experience and academic degree, as defined in Eq. 1). In this section, we analyzed the influence of each participant in this survey. The distribution of participants' weight ranges from 1.20 to 15.00 ($M = 6.63, Md = 6.50, S.D. = 2.92$). The 25th percentile was 4.80, the 50th percentile was 6.50, and the 75th percentile was 8.17. The normality test shows a normal distribution. The p-value for the Shapiro-Wilk test indicates that the values representing the influence (weight) of the participants were normally distributed ($p > 0.05$).

Figure 1 shows the weight distribution through a dispersion and box-plot graph. Two outliers were identified (experts), the weights were 14.00 and 15.00 respectively. Both of them are project managers, with 30 years of experience in the IT industry, and 20 years of experience in Testing. Their highest academic degree is a Master's degree and the first one is a Certified Test Manager (CTM). In our analysis, we used the answers of all participants.

4.1.2 Participants among surveys

In this study, we compare the results of surveys conducted in Argentina, Brazil, Uruguay, and Costa Rica. Table 5 presents the percentages of the positions reported in each previous survey (Dias-Neto et al., 2017; Robiolo et al., 2017) and this study. We present the percentages of Northern Brazil (NBR, n=50), Southern Brazil (SBR, n=56), Uruguay (UY, n=44) (Dias-Neto et al., 2017), Argentina (AR, n=25) (Robiolo et al., 2017), and Costa Rica (CR, n=92). The positions (%) reported are: Analysts (P1), Architects (P2), Developers (P3), Project Managers (P4), Quality Managers (P5), Test Analysts (P6), Test Leaders (P7), and Testers (P8).

In Brazil and Uruguay, 66% of the respondents are working on quality/testing (Quality Manager, Test Leader, Test Analyst, and Tester) and 34% in development activities (Analyst, Architect, Developer, and Project Manager). In the Northern Brazil region 84% are working on quality/testing, in Southern Brazil region 59%, and in Uruguay 57% (Dias-Neto et al., 2017). In contrast, Argentina reported only 16% of the respondents working on quality/testing and 84% in other development activities (16% were not reported) (Robiolo et al., 2017). In Costa Rica, 36% of the respondents are working on quality/testing, including 6% reported as quality engineers.

Table 5. Participants per position (%).

Survey	P1	P2	P3	P4	P5	P6	P7	P8
NBR	12	-	-	4	6	47	14	16
SBR	14	2	4	21	5	38	11	5
UY	7	2	16	18	14	-	7	36
AR	16	-	12	40	-	4	8	4
CR	17	3	16	18	14	8	-	8

In the same way, Table 6 the percentage of respondents by the company's size. The company's size (%) are: Less than 10 (S1), 10 - 49 (S2), 50 - 99 (S3), and more than 100 (S4). We can observe that with the exception of Argentina (AR), most of the answers come from professionals from organizations with more than 100 employees.

Table 6. Participants per company's size (%).

Survey	S1	S2	S3	S4
NBR	10	14	16	60
SBR	9	30	21	39
UY	5	23	20	52
AR	36	24	16	24
CR	12	22	16	50

In the next sections, we present the analysis of the results of the use and importance of the evaluated software testing practices. First, we present the analysis of the use and perceived importance of testing practices. Second, we analyze the correlation between use and perceived importance, Third, the results between use and perceived importance based on "more used" and "more important", "less used" and "less important", "more used" and "Less important", and "less used" and "more important" are discussed. Finally, we compare the results among replications.

4.2 Analysis of the use and perceived importance of testing practices

Table 7 presents a heat map with the results of the use and importance of software testing practices. The first column contains the results of our study and the other four columns the results of the previous studies. The most used and perceived important (P. I.) testing practices in process (P), activities (A), and tools (T) were marked in green, and the least used and important ones were marked in red. The greener color means the practice is deemed useful and/or important, the redder mean the practice is not considered important or not implemented. We present the results of Costa Rica (CR), Argentina (AR) (Robiolo et al., 2017), Northern Brazil (NBR), Southern Brazil (SBR), and Uruguay (UY) (Dias-Neto et al., 2017).

For each testing practice, we could observe some trends by analyzing the use and important across the replications. In all five countries/regions, there is a set of used and important practices (P02: Documentation of test procedures and cases, P03: Recording the results of test execution, P10: Re-execution of tests when the software is modified, A01: Definition of a responsible professional or team, A03: Application of integration tests, A04: Application of system tests, A05: Application of acceptance tests, T01: Availability of a test database for reuse, and T07: Use of tools for recording defects and the effort to fix them-bug tracking), and a set of less used and considered less important practices (P08: Planning/Designing of testing before coding, A10: Registration of the time spent on testing, A11: Measurement of the effort/cost of testing, A13: Measurement of the defect density, A14: Conducting training on software testing, and A17: Analysis of faults patterns-trends).

4.2.1 Use of testing practices

The results of the use of software testing practices per country/region are presented. By analyzing the green patterns in Table 7, we can conclude that the three most used testing processes reported were: the recording of test cases results (P03), the documentation of test procedures and cases (P02), and the re-execution of tests when the software is modified (P10). In the case of testing activities, the three most used were the application of acceptance testing (A05) and system testing (A04), and the definition of a responsible professional or team (A01). Finally, the three most used testing tools were those for recording defects and the effort to fix them - bug tracking (T07), a test database for reuse (T01), and management tools to track and record the results (T04).

On the other hand, the processes for planning/designing of testing before coding (P08), the evaluation of the quality of test artifacts (P11), and the measurement and analysis of the test coverage (P04) were reported as the three least used. The measurement of the defect density (A13), the analysis of faults patterns – trends (A17), and the registration of the time spent on testing (A10) were reported as the three least used activities. Finally, the three least used tools were the tools for automatic generation of test procedures or cases (T03), coverage measurement tools (T08), and tools to estimate test effort and/or schedule (T05).

4.2.2 Importance of testing practices

The importance perceived by the participants on the software testing practices per country/region is presented in Table 7. By observing the green patterns, we can conclude that the three most perceived important testing processes were: the task of recording the results of tests cases (P03), the documentation of test procedures and cases (P02), and the re-execution of tests when the software is modified (P10). These processes were also the most used by practitioners. In the case of testing activities, the three perceived as most important were the application of acceptance testing (A05), the application of integration tests (A03), and the storage of records (logs) of the executed tests (A12). Besides, system testing (A04), and a definition of a responsible professional or team (A01) were perceived as important. Finally, the three most important testing tools were: tools for recording defects and the effort to fix them - bug tracking (T07), tools for automatic execution of test procedures or cases (T02), and a test database for reuse (T01). The management tools to track and record the results (T04) were also perceived as important.

Likewise, the processes for test artifacts quality (P11), for planning/designing of testing before coding (P08), and for reporting evaluation of a test round (P13) were perceived as the three least important. The measurement of the defect density (A13), the application of exploratory tests (A07), and the analysis of faults patterns – trends (A17) were perceived as the three least important activities. The perceived as the three least important tools were the tools to estimate test effort and/or schedule (T05), coverage measurement tools (T08), and tools for automatic generation of test procedures or cases (T03).

4.3 Analysis of correlation between use and perceived importance

Table 8 presents the Spearman's rho correlation coefficient between the use and perceived importance of each testing practice (two-tail test with $p<0.01$). In this case, there was a positive correlation between the use and perceived importance, and all correlations were statistically significant. The values above 0.5 were considered as highly correlated and are marked in bold. A high correlation means that the participants either: (1) deemed the practice useful and important, or (2) deemed the practice not useful and not important.

Our results show that although there is a correlation between the values of use and perceived importance, only 18 of 42 practices are highly correlated (P01: Documentation of test plan, P02: Documentation of test procedures and cases, P03: Recording the results of test execution, P09: Monitoring adherence to the test process, P12: Setting a priori criteria to stop testing, P13: Reporting evaluation of a test round, A01: Definition of a responsible professional or team, A04: Application of system tests, A06: Application of regression tests, A07: Application of exploratory tests, A10: Registration of the time spent on testing, A11: Measurement of the effort/cost of testing, A12: Storage of records (log) of the executed tests, A13: Measurement of the defect density, T01: Availability of a test database for reuse, T05: Use of tools to estimate test effort and/or schedule, T06: Use of test management tools to enact activities and artifacts, T07: Use of tools for recording defects and the effort to fix them-bug tracking). In the following section, we compare the relation between use and importance.

4.4 Analysis between use and perceived importance

Dias-Neto et al. (2017) analyze the level of use and perceived importance dividing the test practices into two equal groups of the total 42 practices. Table 9 presents the "More used" and "More important", and the "Less used" and "Less important" testing practices according to the answers of Costa Rican practitioners. To classify the practices, the top 21 most used practices and the top 21 most perceived as important

Table 7. Comparison on the use and perceived importance of testing practices.

	CR (n=92)		AR (n=25)		NBR (n=50)		SBR (n=56)		UY (n=44)		
	Use	**P. I.**	**Use**	**P. I.**	**Use**	**P. I.**	**Use**	**P. I.**	**Use**	**P. I.**	
P01	76%	86%	67%	76%	78%	85%	69%	75%	79%	85%	*P01*
P02	84%	95%	72%	80%	86%	87%	81%	84%	83%	89%	*P02*
P03	85%	96%	76%	83%	91%	94%	86%	90%	90%	94%	*P03*
P04	64%	87%	57%	70%	65%	75%	65%	79%	66%	76%	*P04*
P05	77%	93%	58%	76%	75%	81%	73%	78%	74%	80%	*P05*
P06	72%	90%	59%	78%	71%	77%	73%	79%	71%	76%	*P06*
P07	66%	85%	53%	76%	68%	75%	63%	74%	65%	77%	*P07*
P08	55%	79%	52%	72%	62%	68%	59%	70%	58%	67%	*P08*
P09	67%	85%	47%	64%	64%	75%	58%	66%	61%	72%	*P09*
P10	83%	95%	74%	85%	90%	92%	87%	91%	86%	90%	*P10*
P11	58%	76%	48%	71%	63%	71%	60%	69%	62%	75%	*P11*
P12	72%	87%	50%	73%	65%	77%	63%	73%	64%	74%	*P12*
P13	68%	84%	58%	78%	72%	80%	62%	67%	74%	86%	*P13*
A01	82%	93%	79%	89%	90%	94%	84%	86%	89%	92%	*A01*
A02	79%	92%	79%	93%	71%	81%	79%	88%	78%	86%	*A02*
A03	81%	95%	77%	88%	79%	87%	83%	90%	86%	92%	*A03*
A04	83%	93%	81%	89%	88%	92%	89%	94%	88%	93%	*A04*
A05	88%	97%	82%	87%	83%	88%	82%	89%	85%	92%	*A05*
A06	79%	91%	74%	82%	84%	90%	83%	91%	84%	90%	*A06*
A07	69%	76%	60%	69%	85%	82%	77%	81%	82%	75%	*A07*
A08	71%	89%	73%	80%	70%	81%	68%	77%	71%	81%	*A08*
A09	71%	89%	80%	88%	70%	88%	68%	79%	74%	85%	*A09*
A10	63%	78%	52%	60%	66%	75%	71%	75%	70%	78%	*A10*
A11	64%	81%	45%	51%	62%	78%	69%	76%	65%	78%	*A11*
A12	81%	94%	76%	80%	78%	84%	79%	78%	83%	87%	*A12*
A13	56%	74%	47%	57%	47%	63%	56%	65%	49%	60%	*A13*
A14	65%	84%	55%	65%	64%	80%	62%	80%	66%	80%	*A14*
A15	77%	89%	81%	90%	82%	80%	82%	84%	85%	85%	*A15*
A16	78%	88%	71%	82%	78%	81%	70%	77%	77%	78%	*A16*
A17	58%	78%	67%	77%	63%	73%	57%	72%	60%	71%	*A17*
A18	78%	92%	75%	82%	78%	84%	77%	89%	83%	85%	*A18*
A19	70%	89%	71%	79%	80%	84%	70%	80%	84%	85%	*A19*
T01	77%	91%	79%	87%	80%	81%	78%	88%	85%	91%	*T01*
T02	71%	94%	54%	74%	68%	78%	69%	81%	66%	81%	*T02*
T03	58%	83%	53%	69%	58%	69%	62%	71%	62%	73%	*T03*
T04	74%	90%	66%	83%	82%	83%	75%	84%	84%	88%	*T04*
T05	62%	81%	50%	69%	64%	73%	65%	75%	63%	74%	*T05*
T06	73%	89%	56%	77%	71%	78%	68%	80%	71%	76%	*T06*
T07	82%	95%	65%	83%	86%	89%	84%	89%	88%	92%	*T07*
T08	61%	82%	48%	73%	57%	71%	62%	78%	54%	68%	*T08*
T09	66%	87%	48%	75%	59%	75%	70%	82%	61%	70%	*T09*
T10	67%	83%	56%	73%	69%	83%	66%	78%	71%	83%	*T10*

Table 8. Spearman's correlation between use and importance.

Id	Testing practice	r_s
P01	Documentation of test plan	**.585**
P02	Documentation of test proc. and cases	**.644**
P03	Recording the results of test execution	**.556**
P04	Measurement, analysis of test coverage	.393
P05	Use of methodology or process	.492
P06	Analysis of identified defects	.400
P07	Identification and use of risks	.447
P08	Plan/Design tests before coding	.372
P09	Monitoring adherence to the test process	**.602**
P10	Re-execution of tests when modified	.467
P11	Evaluation of the quality of test artifacts	.395
P12	Setting a priori criteria to stop testing	**.712**
P13	Reporting evaluation of a test round	**.537**
A01	Def. of a professional or team	**.516**
A02	Application of unit tests	.448
A03	Application of integration tests	.456
A04	Application of system tests	**.605**
A05	Application of acceptance tests	.472
A06	Application of regression tests	**.562**
A07	Application of exploratory tests	**.587**
A08	Application of performance tests	.306
A09	Application of security tests	.323
A10	Registration of the time spent on testing	**.565**
A11	Measurement of the effort/cost of testing	**.561**
A12	Storage of records (log) of the executed tests	**.585**
A13	Measurement of the defect density	**.532**
A14	Conducting training on software testing	.459
A15	Separation of testing and dev activities	.468
A16	Storage of test data for future use	.482
A17	Analysis of faults patterns (trend)	.411
A18	Availability of human resources full time	.476
A19	Selection of test techniques based on features	.450
T01	Availability of a test database for reuse	**.548**
T02	Automatic execution of test proc. or cases	.360
T03	Automatic generation of test proc. or cases	.355
T04	Test management tools to track and record	.453
T05	To estimate test effort and/or schedule	**.542**
T06	Test management tools to enact artifacts	**.545**
T07	Recording defects and the effort to fix them	**.518**
T08	Use of coverage measurement tools	.479
T09	Continuous integration for automated tests	.424
T10	Selection of test tools based on proj. charcs.	.450

practices were selected. The set of "most used, most important" practices represents the good practices in testing performed by Cost Rican practitioners. The set of "least used, least important" testing practices represent those that seem to be not relevant in the context of these organizations. Furthermore, these practices could represent gaps in knowledge about their benefits or simply a lack of organizational resources to put them into practice.

Table 10 presents the "More used" and "Less important", and the "Less used" and "More important" testing practices. The set of "most used, least important" testing practices includes the practices used by software practitioners but considered not as important as other practices. In this case, other used practices could generate more value in supporting testing activities. The set of "least used, most important" testing practices are those not used by practitioners in their software organizations, but perceived as important for their professional practice.

5 Discussion

The results of the use of software testing practices show that practitioners in our industry are currently implementing basic processes and tools for performing software testing, but at the same time, they are not using key metrics for assessing testing results or the quality of the testing products. This clearly represents an important area for improvement in our industry and a challenge for universities for teaching these concepts.

Second, although not perceived as important by practitioners, we believe that metrics (such as defect density) and processes such as analysis of fault patterns are key for software organizations that aspire to improve their processes and reach higher maturity levels. They may not be deemed important now, but they will gain more importance as the industry matures.

On the other hand, based on the analysis of the correlation between use and perceived importance, we agreed with (Dias-Neto et al., 2017) when they state that practitioners can find the practices they use daily to be important and therefore, either they cannot distinguish between the use and important or they do not see value in the distinction. In the following section, we compare the relation between use and importance.

Finally, based on the analysis between use and perceived importance, the set of "least used, least important" testing practices could represent gaps in knowledge about their benefits or simply a lack of organizational resources to put them into practice. These practices may point out the gaps between academia and industry and, for example, have to be addressed through practitioners' training courses and software process improvement plans to show the benefits of their application. The set of "least used, most important" can be complex or expensive to implement, they may have considerable training needs, or these organizations may not have the necessary tools to perform them.

5.1 Comparing the results among replications

To compare the results of this survey with previous studies Dias-Neto et al. (2017) the "More used" and "More important" testing practices, and the "Less used" and "Less important" testing practices were analyzed. Table 11 presents the "More used" and "More important" testing practices for each replication. Five testing practices are common in all surveys (P03: Recording the results of test execution, A01: Definition of a responsible professional or team, A03: Application of integration tests, A04: Application of system tests, A05: Application of acceptance tests), and four practices are common in four surveys (P2: Documentation of test procedures and cases, P10: Re-execution of tests when the software is modified, A15: Separation of testing and development activities, A18: Availability of human resources allocated full time for testing).

Table 9. Use and importance similarities between testing practices.

Id	"More used" and "More important"	Id	"Less used" and "Less important"
P02	Documentation of test procedures and cases	P04	Measurement and analysis of the test coverage
P03	Recording the results of test execution	P07	Identification and use of risks
P05	Use of methodology or process	P08	Planning/Designing of testing before coding
P06	Analysis of identified defects	P09	Monitoring adherence to the test process
P10	Re-execution of tests when modified	P11	Evaluation of the quality of test artifacts
A01	Definition of a responsible professional or team	P13	Reporting evaluation of a test round
A02	Application of unit tests	A07	Application of exploratory tests
A03	Application of integration tests	A10	Registration of the time spent on testing
A04	Application of system tests	A11	Measurement of the effort/cost of testing
A05	Application of acceptance tests	A13	Measurement of the defect density
A06	Application of regression tests	A14	Conducting training on software testing
A12	Storage of records (log) of the executed tests	A17	Analysis of faults patterns (trend)
A15	Separation of testing and dev activities	A19	Selection of test techniques based on features
A18	Availability of human resources full time	T03	Tools for automatic generation of test cases
T01	Availability of a test database for reuse	T05	Use of tools to estimate test effort and/or schedule
T04	Test management tools to track and record	T08	Use of coverage measurement tools
T06	Test management tools to enact artifacts	T09	Use of continuous integration tools for tests
T07	Tools for bug tracking and effort to fix them	T10	Selection of test tools according to project charcs.

Table 10. Use and importance similarities between testing practices.

Id	"More used" and "Less important"	Id	"Less used" and "More important"
P01	Documentation of test plan	A08	Application of performance tests
P12	Setting a priori criteria to stop testing	A09	Application of security tests
A16	Storage of test data for future use	T02	Automatic execution of test procedures or cases

Table 12 presents the "Less used" and "Less important" testing practices for each replication. Six testing practices are reported in four surveys (P07: Identification and use of risks for planning and executing software tests, P09: Monitoring adherence to the test process, A11: Measurement of the effort/cost of testing, T03: Use of tools for automatic generation of test procedures or cases, T05: Use of tools to estimate test effort and/or schedule, T08: Use of coverage measurement tools). These practices represent a gap between software testing state of the art (academia) and the state of the practice (practitioners) considering that the list of practices in the survey was defined considering the academic literature. In (De Greca et al., 2015), no practices were classified as less used and less important.

In Table 11 and Table 12, we only included practices of our survey, and practices with more than three occurrences across replications. We found no significant differences in practices perceived usefulness and importance between our survey and previous surveys. As in other countries, important practices are not being used in our software industry. This opens an interesting line of research to find out why they are not being used.

Our survey aggregated evidence previously reported and presented new evidence on the use and perceived importance of testing practices in the industry:

- There is a gap between software testing state of the art and state of the practice. This study identified a set of testing practices classified as "Less important" and "Less used" (Table 9), and the set of these "Less important" and "Less used" testing practices reported in multiple replications (Table 12).
- The findings support that organizations mainly use the ad hoc criteria to stop testing. In Dias-Neto et al. (2017); Robiolo et al. (2017) the practice P12: Setting a priori criteria to stop the testing is ranked low (the level of use ranked in the bottom 10th (65%), 10th (63%), 12th (64%) and 7th (50%) positions respectively). In the case of Costa Rica P12 was ranked 23rd (72%). The perceived importance received a total of 77% (8th), 73% (10th), and 74% (11th) in Dias-Neto et al. (2017), 73% (13th) in Robiolo et al. (2017), and 87% (17th) in Costa Rica.
- The application of unit tests (A02) is not within the three most used (71%, 79%, 78%) and important (81%, 88%, 86%) practices in any of the regions reported in Dias-Neto et al. (2017). However, in Robiolo et al. (2017) unit tests were reported as the most important practice (93%) and used (79%). In this study, unit testing was reported used (79%) and important (92%). According to the findings, we cannot conclude about the use and importance level of unit tests. Other testing practices, such as A03: Application of integration tests, A04: Application of system tests, A05: Application of acceptance tests, and A06: Application of regression tests were reported as used and important in multiple replications (Table 11).
- The findings indicated some level on the use and importance of automated testing. However, T03: Use of tools for automatic generation of test procedures or

Table 11. Comparison of “More used” and “More important” testing practices.

Id	“More used” and “More important”	This study	Robiolo et al. (2017)	Dias-Neto et al. (2017)	De Greca et al. (2015)	Dias-Neto et al. (2006)
P02	Documentation of test procedures and cases	✓	✓	✓		✓
P03	Recording the results of test execution	✓	✓	✓	✓	✓
P05	Use of methodology or process	✓				✓
P06	Analysis of identified defects	✓				
P10	Re-execution of tests when the software is modified	✓	✓	✓		✓
A01	Definition of a responsible professional or team	✓	✓	✓	✓	✓
A02	Application of unit tests	✓	✓		✓	
A03	Application of integration tests	✓	✓	✓	✓	✓
A04	Application of system tests	✓	✓	✓	✓	✓
A05	Application of acceptance tests	✓	✓	✓	✓	✓
A06	Application of regression tests	✓	✓	✓		
A12	Storage of records (log) of the executed tests	✓	✓			
A15	Separation of testing and dev activities	✓	✓		✓	✓
A16	Storage of test data for future use		✓		✓	✓
A18	Availability of human resources allocated full time for testing	✓	✓	✓	✓	
T01	Availability of a test database for reuse	✓	✓			
T04	Test management tools to track and record	✓	✓			
T06	Test management tools to enact activities and artifacts	✓				
T07	Tools for recording defects and the effort to fix them (tracking)	✓	✓			

Table 12. Comparison of “Less used” and “Less important” testing practices.

Id	“Less used” and “Less important”	This study	Robiolo et al. (2017)	Dias-Neto et al. (2017)	De Greca et al. (2015)	Dias-Neto et al. (2006)
P04	Measurement and analysis of the test coverage	✓	✓	✓		
P07	Identification and use of risks	✓	✓	✓		✓
P08	Planning/Designing of testing before coding	✓	✓	✓		
P09	Monitoring adherence to the test process	✓	✓	✓		✓
P11	Evaluation of the quality of test artifacts	✓	✓	✓		
P13	Reporting evaluation of a test round	✓				
A07	Application of exploratory tests	✓	✓			
A10	Registration of the time spent on testing	✓	✓			
A11	Measurement of the effort/cost of testing	✓	✓	✓		✓
A13	Measurement of the defect density	✓	✓	✓		
A14	Conducting training on software testing	✓	✓			
A17	Analysis of faults patterns (trend)	✓		✓		✓
A19	Selection of test techniques based on features	✓				
T03	Use of tools for automatic generation of test procedures or cases	✓	✓	✓		✓
T05	Use of tools to estimate test effort and/or schedule	✓	✓	✓		✓
T08	Use of coverage measurement tools	✓	✓	✓		✓
T09	Use of continuous integration tools for automated tests	✓	✓			
T10	Selection of test tools according to project characteristics	✓	✓			

cases was reported as "Less used" and "Less important" in Dias-Neto et al. (2017); Robiolo et al. (2017) and this study. Besides, the testing practices T02: Use of tools for automatic execution of test procedures or cases, and T09: Use of continuous integration tools for automated tests were categorized as "Less used". We cannot infer whether the level of use is lesser or higher than manual testing.

Finally, we confirmed some similarities highlighted by Dias-Neto et al. (2017) regarding industrial surveys: (1) testing automation is a concern, but it has not reached full adoption in industry, (2) the ad hoc has been reported as one of the main used criteria to stop testing, (3) the use of tools for recording defects and bug tracking are the most adopted, and (4) the most used testing levels are acceptance, integration, system, and unit testing.

5.2 Getting Feedback from Practitioners

To get some feedback about the significance and usefulness of this research from the practitioners' perspective, we made two presentations to different groups of professionals about our study results. After presentations, we asked them the following two questions: (1) Do you think that the data on this presentation provides value for your professional practice? (2) What would you like to see in future presentations?

For the first question, everyone who answered responded in the affirmative. They considered the results from the survey useful to keep up to date with industry trends and improve their own software processes. One person mentioned the importance of doing an informal benchmark with this initial data. A couple of them also mentioned the importance for academia to know these data for keeping updated their curricula and for better defining the exit profile of their graduates.

For the second question, the answers varied substantially. Some people would like to see presentations with specific examples or case studies on how to implement software testing practices in organizations. Others would like to have a presentation on guidelines about how to implement some of those practices in their own organizations. Others suggested having presentations about software testing metrics and tools (including the measurement of testing effectiveness), and how to implement them in small and medium organizations. Finally, one person suggested to hold an entire workshop on software testing and to include software security testing as the main issue.

6 Conclusions

This paper reported a survey study of software testing practices in the Costa Rican software industry and compared the results with previous studies conducted in South America. We characterized a set of testing practices with respect to their use and perceived importance from the point of view of 92 practitioners.

The main software testing practices reported in this survey were the recording of the results of tests, documentation of test procedures and cases, and re-execution of tests when the software is modified. Acceptance and system testing were the two most useful and important testing types. The tools for recording defects and the effort to fix them (bug tracking) and the availability of a test database for reuse were reported useful and important. In contrast, the planning and designing of software testing before coding and evaluating the quality of test artifacts were not a regular practice. Finally, there is a lack of measurement of defect density and test coverage in the industry; and tools for automatic generation of test cases and for estimating testing effort are rarely used.

A set of testing practices were common across different countries: the application of integration, system and acceptance tests, the recording of test execution results and the definition of a responsible professional, or team for testing. In contrast, our results confirm that the main testing limitations are the monitoring and measurement of tests and defects, the automatic generation of test cases, and procedures and the management of test coverage and effort. These last three are clear areas for process improvement.

Further studies in different countries and regions should be conducted to compare industrial trends in software testing practices. We believe this work could be used by organizations, practitioners, and academics to improve the state of the practice in our software industry. For future work, it could be interesting to make a comparison using the demographic data of the participants (such as types of projects, organizations' characteristics, and others) to find out if different demographics influence the results by country.

Acknowledgements

This work was partially supported by Universidad Estatal a Distancia Comiex-19-2017 and Universidad de Costa Rica project No. 834-B7-749. We would like to thank Guilherme Travassos, Santiago Matalonga, Martín Solari, Arilo Dias-Neto and Gabriela Robiolo for providing the earlier version of the questionnaire. We thank all practitioners of the survey for their participation.

References

Andersson, C. and Runeson, P. (2002). Verification and validation in industry-a qualitative survey on the state of practice. In *Proceedings International Symposium on Empirical Software Engineering*, pages 37–47. IEEE.

Aymerich, B., Díaz-Oreiro, I., Guzmán, J. C., López, G., and Garbanzo, D. (2018). Software development practices in costa rica: A survey. In *International Conference on Applied Human Factors and Ergonomics*, pages 122–132. Springer.

Basili, V., Gianluigi, C., and Rombach, D. (1994). The goal question metric approach. *Encyclopedia of software engineering*, pages 528–532.

Beck, L. L. and Perkins, T. E. (1983). A survey of software engineering practice: tools, methods, and results. *IEEE Transactions on Software Engineering*, (5):541–561.

Bhuiyan, S. A. R., Rahim, M. S., Chowdhury, A. E., and Hasan, M. H. (2018). A survey of software quality assurance and testing practices and challenges in bangladesh. *International Journal of Computer Applications*, 975:8887.

Carver, J. C. (2010). Towards reporting guidelines for experimental replications: A proposal. In *1st international workshop on replication in empirical software engineering*, pages 2–5. Citeseer.

Carver, J. C., Juristo, N., Baldassarre, M. T., and Vegas, S. (2014). Replications of software engineering experiments.

Causevic, A., Sundmark, D., and Punnekkat, S. (2010). An industrial survey on contemporary aspects of software testing. In *2010 Third International Conference on Software Testing, Verification and Validation*, pages 393–401. IEEE.

Chan, F., Tse, T., Tang, W., and Chen, T. (2005). Software testing education and training in hong kong. In *Fifth International Conference on Quality Software (QSIC'05)*, pages 313–316. IEEE.

Daka, E. and Fraser, G. (2014). A survey on unit testing practices and problems. In *2014 IEEE 25th International Symposium on Software Reliability Engineering*, pages 201–211. IEEE.

De Greca, F., Rossi, B. D., Robiolo, G., and Travassos, G. H. (2015). Aplicación y valoración de la verificación y validación de software: una encuesta realizada en buenos aires. In *Simposio Argentino de Ingeniería de Software (ASSE 2015)-JAIIO 44 (Rosario, 2015)*.

Deak, A. (2014). A comparative study of testers' motivation in traditional and agile software development. In *International Conference on Product-Focused Software Process Improvement*, pages 1–16. Springer.

Deak, A. and Stålhane, T. (2013). Organization of testing activities in norwegian software companies. In *2013 IEEE Sixth International Conference on Software Testing, Verification and Validation Workshops*, pages 102–107. IEEE.

Deak, A., Stålhane, T., and Cruzes, D. (2013). Factors influencing the choice of a career in software testing among norwegian students. *Software Engineering*, page 796.

Dias-Neto, A., Natali, A. C. C., Rocha, A. R., and Travassos, G. H. (2006). Caracterização do estado da prática das atividades de teste em um cenário de desenvolvimento de software brasileiro. *V Simpósio Brasileiro de Qualidade de Software, Vila Velha, ES*.

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.

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.

Fernández-Sanz, L. (2005). Un sondeo sobre la práctica actual de pruebas de software en españa. *REICIS. Revista Española de Innovación, Calidad e Ingeniería del Software*, 1(2).

Fernández-Sanz, L., Villalba, M. T., Hilera, J. R., and Lacuesta, R. (2009). Factors with negative influence on software testing practice in spain: A survey. In *European conference on software process improvement*, pages 1–12. Springer.

Garousi, V., Coşkunçay, A., Betin-Can, A., and Demirörs, O. (2015). A survey of software engineering practices in turkey. *Journal of Systems and Software*, 108:148–177.

Garousi, V., Coşkunçay, A., Demirörs, O., and Yazici, A. (2016). Cross-factor analysis of software engineering practices versus practitioner demographics: An exploratory study in turkey. *Journal of Systems and Software*, 111:49–73.

Garousi, V. and Felderer, M. (2017). Living in two different worlds: A comparison of industry and academic focus areas in software testing. *IEEE Software*, (1):1–1.

Garousi, V., Felderer, M., Kuhrmann, M., and Herkiloğlu, K. (2017). What industry wants from academia in software testing?: Hearing practitioners' opinions. In *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*, pages 65–69. ACM.

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.

Gelperin, D. and Hetzel, B. (1988). The growth of software testing. *Communications of the ACM*, 31(6):687–695.

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

Ghazi, A. N., Petersen, K., and Börstler, J. (2015). Heterogeneous systems testing techniques: An exploratory survey. In *International Conference on Software Quality*, pages 67–85. Springer.

Ghazi, A. N., Petersen, K., Reddy, S. S. V. R., and Nekkanti, H. (2017). Survey research in software engineering: problems and strategies. *arXiv preprint arXiv:1704.01090*.

Greiler, M., Deursen, A. v., and Storey, M.-A. (2012). Test confessions: A study of testing practices for plug-in systems. In *Proceedings of the 34th International Conference on Software Engineering*, pages 244–254. IEEE Press.

Grindal, M., Offutt, J., and Mellin, J. (2006). On the testing maturity of software producing organizations. In *Testing: Academic & Industrial Conference-Practice And Research Techniques (TAIC PART'06)*, pages 171–180. IEEE.

Hynninen, T., Kasurinen, J., Knutas, A., and Taipale, O. (2018). Software testing: Survey of the industry practices. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1449–1454. IEEE.

Juristo, N. and Gómez, O. S. (2010). Replication of software engineering experiments. In *Empirical software engineering and verification*, pages 60–88. Springer.

Juristo, N., Moreno, A. M., and Vegas, S. (2004). Reviewing 25 years of testing technique experiments. *Empirical Software Engineering*, 9(1-2):7–44.

Kanij, T., Merkel, R., and Grundy, J. (2014). A preliminary survey of factors affecting software testers. In *2014 23rd*

Australian Software Engineering Conference, pages 180–189. IEEE.

Kassab, M. (2018). Testing practices of software in safety critical systems: Industrial survey. In *20th International Conference on Enterprise Information Systems, ICEIS 2018*, pages 359–367. SciTePress.

Kassab, M., DeFranco, J. F., and Laplante, P. A. (2017). Software testing: The state of the practice. *IEEE Software*, 34(5):46–52.

Kasurinen, J., Taipale, O., and Smolander, K. (2010). Software test automation in practice: empirical observations. *Advances in Software Engineering*, 2010.

Kirk, D. and Tempero, E. (2012). Software development practices in new zealand. In *2012 19th Asia-Pacific Software Engineering Conference*, volume 1, pages 386–395. IEEE.

Kochhar, P. S., Thung, F., Nagappan, N., Zimmermann, T., and Lo, D. (2015). Understanding the test automation culture of app developers. In *2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST)*, pages 1–10. IEEE.

Kochhar, P. S., Xia, X., and Lo, D. (2019). Practitioners' views on good software testing practices. In *Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Practice*, pages 61–70. IEEE Press.

Kuhrmann, M., Diebold, P., Münch, J., Tell, P., Garousi, V., Felderer, M., Trektere, K., McCaffery, F., Linssen, O., Hanser, E., et al. (2017). Hybrid software and system development in practice: waterfall, scrum, and beyond. In *Proceedings of the 2017 International Conference on Software and System Process*, pages 30–39. ACM.

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

Lima, B. and Faria, J. P. (2016). A survey on testing distributed and heterogeneous systems: The state of the practice. In *International Conference on Software Technologies*, pages 88–107. Springer.

Linåker, J., Sulaman, S. M., Maiani de Mello, R., and Höst, M. (2015). Guidelines for conducting surveys in software engineering.

Lindsay, R. M. and Ehrenberg, A. S. (1993). The design of replicated studies. *The American Statistician*, 47(3):217–228.

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 Measurement*, page 58. ACM.

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

Park, J., Ryu, H., Choi, H.-J., and Ryu, D.-K. (2008). A survey on software test maturity in korean defense industry. In *Proceedings of the 1st India software engineering conference*, pages 149–150. ACM.

Pérez, J., Mens, T., and Kamseu, F. (2013). A pilot study on software quality practices in belgian industry. In *2013 17th European Conference on Software Maintenance and Reengineering*, pages 395–398. IEEE.

Pfahl, D., Yin, H., Mäntylä, M. V., and Münch, J. (2014). How is exploratory testing used? a state-of-the-practice survey. In *Proceedings of the 8th ACM/IEEE international symposium on empirical software engineering and measurement*, page 5. ACM.

Pham, R., Singer, L., Liskin, O., Figueira Filho, F., and Schneider, K. (2013). Creating a shared understanding of testing culture on a social coding site. In *Proceedings of the 2013 International Conference on Software Engineering*, pages 112–121. IEEE Press.

Quesada-López, C., Hernandez-Aguero, E., and Jenkins, M. (2019). A survey of software testing practices in costa rica. In *Proceedings of the XXII Ibero-American Conference on Software Engineering (CibSE 2019). La Habana, Cuba, 23-27 Abril 2019*, pages 107–145.

Quesada-López, C. and Jenkins, M. (2017). Estudio sobre las prácticas de la ingeniería de software en costa rica: Resultados preliminares. In *Proceedings of the XX Ibero-American Conference on Software Engineering (CibSE 2017). Buenos Aires, Argentina, 22-23 May 2017*, pages 107–145.

Quesada-López, C. and Jenkins, M. (2018). Factores asociados a prácticas de desarrollo y pruebas de software en costa rica: Un estudio exploratorio. In *Proceedings of the XXI Ibero-American Conference on Software Engineering (CibSE 2018). Bogotá, Colombia, 23-27 Abril 2018*, pages 107–145.

Rafi, D. M., Moses, K. R. K., Petersen, K., and Mäntylä, M. V. (2012). Benefits and limitations of automated software testing: Systematic literature review and practitioner survey. In *Proceedings of the 7th International Workshop on Automation of Software Test*, pages 36–42. IEEE Press.

Raulamo-Jurvanen, P., Hosio, S., and Mäntylä, M. V. (2019). Practitioner evaluations on software testing tools. In *Proceedings of the Evaluation and Assessment on Software Engineering*, pages 57–66. ACM.

Robiolo, G., M, M., Rossi, B., and Travassos, G. H. (2017). Aplicación e importancia de las pruebas de software: una encuesta realizada en buenos aires en el Ámbito público. In *XX Ibero-American Conference on Software Engineering (CibSE 2017). Argentina, 22-23 May 2017*.

Rodrigues, A. and Dias-Neto, A. (2016). Relevance and impact of critical factors of success in software test automation lifecycle: A survey. In *Proceedings of the 1st Brazilian Symposium on Systematic and Automated Software Testing*, page 6. ACM.

Runeson, P. (2006). A survey of unit testing practices. *IEEE software*, 23(4):22–29.

Runeson, P., Andersson, C., and Höst, M. (2003). Test processes in software product evolution—a qualitative survey on the state of practice. *Journal of software maintenance and evolution: Research and practice*, 15(1):41–59.

Scatalon, L. P., Fioravanti, M. L., Prates, J. M., Garcia, R. E., and Barbosa, E. F. (2018). A survey on graduates' curriculum-based knowledge gaps in software testing. In *2018 IEEE Frontiers in Education Conference (FIE)*, pages 1–8. IEEE.

Smolander, K., Taipale, O., and Kasurinen, J. (2016). Software test automation in practice: Empirical observations. In *Data Structure and Software Engineering*, pages 107–145. Apple Academic Press.

Sung, P. W.-B. and Paynter, J. (2006). Software testing practices in new zealand. In *In Proceedings of the 19th Annual Conference of the National Advisory Committee on Computing Qualifications*, pages 273–282.

Taipale, O., Smolander, K., and Kälviäinen, H. (2005). Finding and ranking research directions for software testing. In *European Conference on Software Process Improvement*, pages 39–48. Springer.

Taipale, O., Smolander, K., and Kälviäinen, H. (2006). A survey on software testing. *6th International SPICE*.

Torkar, R. and Mankefors, S. (2003). A survey on testing and reuse. In *Proceedings 2003 Symposium on Security and Privacy*, pages 164–173. IEEE.

Vasanthapriyan, S. (2018). A study of software testing practices in sri lankan software companies. In *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pages 339–344. IEEE.

Vonken, F., Brunekreef, J., Zaidman, A., and Peeters, F. (2012). Software engineering in the netherlands: the state of the practice. *Technical Report Series TUD-SERG-2012-022*.

Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, page 38. Citeseer.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2012). *Experimentation in software engineering*. Springer Science & Business Media.

Wojcicki, M. A. and Strooper, P. (2006). A state-of-practice questionnaire on verification and validation for concurrent programs. In *Proceedings of the 2006 workshop on Parallel and distributed systems: testing and debugging*, pages 1–10. ACM.

Yli-Huumo, J., Taipale, O., and Smolander, K. (2014). Software development methods and quality assurance: Special focus on south korea. In *European Conference on Software Process Improvement*, pages 159–169. Springer.