DOI: 10.15514/ISPRAS-2023-35(1)-11

A Systematic Mapping Study on Software Testing in the DevOps Context

[1] *B. Pando, ORCID: 0000-0002-8133-631X <brian.pando@unas.edu.pe>*
[2] *A. Dávila, ORCID: 0000-0003-2455-9768 <abraham.davila@pucp.edu.pe>*

[1] *National Agrarian University of La Selva,*
Tingo María, Huánuco, Peru
[2] *Pontificia Universidad Católica del Perú,*
Lima, Perú, 15088

Abstract. DevOps is a philosophy and framework that allows software development and operations teams to work in a coordinated manner, with the purpose of developing and releasing software quickly and cheaply. However, the effectiveness and benefits of DevOps depend on several factors, as reported in the literature. In particular, several studies have been published on software test automation, which is a cornerstone for the continuous integration phase in DevOps, which needs to be identified and classified. This study consolidates and classifies the existing literature on automated tests in the DevOps context. For the study, a systematic mapping study was performed to identify and classify papers on automated testing in DevOps based on 8 research questions. In the query of 6 relevant databases, 3,312 were obtained; and then, after the selection process, 299 papers were selected as primary studies. Researchers maintain a continuing and growing interest in software testing in the DevOps context. Most of the research (71.2%) is carried out in the industry and is done on web applications and SOA. The most reported types of tests are unit and integration tests.

Keywords: DevOps; software testing; systematic mapping study

For citation: Pando B., Dávila A. A Systematic Mapping Study on Software Testing in the DevOps Context. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 1, 2023. pp. 163-188. DOI: 10.15514/ISPRAS-2023-35(1)-11

Acknowledgments. Authors recognize reviews from members of Grupo de Investigación y Desarrollo en Ingeniería de Software – Pontificia Universidad Católica del Perú GIDIS-PUCP).

Систематический обзор литературы по тестированию программного обеспечения в контексте DevOps

[1] *Б. Пандо, ORCID: 0000-0002-8133-631X <brian.pando@unas.edu.pe>*
[2] *А. Давила, ORCID: 0000-0003-2455-9768 <abraham.davila@pucp.edu.pe>*

[1] *Национальный аграрный университет Ла-Сельвы,*
Перу, Уануко, Тинго Мария
[2] *Папский католический университет Перу,*
Перу, 15088, Лима

Аннотация. DevOps – это философия и инфраструктура, которые позволяют группам разработчиков и эксплуатации программного обеспечения работать скоординированно с целью быстрой и дешевой разработки и выпуска программного обеспечения. Однако, как сообщается в литературе, эффективность и преимущества DevOps зависят от нескольких факторов. В частности, было опубликовано несколько результатов исследований по автоматизации тестирования программного обеспечения, которая является краеугольным камнем фазы непрерывной интеграции в DevOps. Эти работы нуждаются в идентификации и классификации. В нашем исследовании консолидируется и классифицируется существующая литература по автоматизированному тестированию в контексте

DevOps. Для исследования было проведено систематическое сопоставление литературных источников на основе 8 исследовательских вопросов. Путем выполнения запросов к шести уместным базам данных было получено 3312 статей. После процесса отбора 299 статей были выбраны в качестве основных. Исследователи сохраняют постоянный и растущий интерес к тестированию программного обеспечения в контексте DevOps. Большая часть исследований (71,2%) проводится в производственной сфере и затрагивают веб-приложения и SOA. Наиболее распространенными типами тестов являются модульные и интеграционные тесты.

Ключевые слова: DevOps; тестирование программного обеспечения; систематический обзор литературы

Для цитирования: Пандо Б., Давила А. Систематический обзор литературы по тестированию программного обеспечения в контексте DevOps. Труды ИСП РАН, том 35, вып. 1, 2023 г., стр. 163-188. DOI: 10.15514/ISPRAS-2023-35(1)-11

Благодарности. Авторы признательны за отзывы членам Группы исследований и разработок в области программной инженерии Папского католического университета Перу.

1. Introduction

The software market constantly demands strategies that allow it to deal with changes quickly [1], [2]. However, these strategies must maintain quality and avoid the costs of application downtime and failure [3]. Although agile methods are presented as a good alternative; these do not close the cycle until the delivery and operation of the software [4]. In this context, the DevOps philosophy and framework extends the agile methodology to deliver applications quickly and frequently [5], improving performance and costs [6], and taking care of the product quality [7], [8], [9]. So, with the support of top management [10], DevOps can represent a great opportunity for companies of any size to gain a foothold in the market [11]. For this reason, various companies have been adopting it [12] or have adopted plans [13]. Also, DevOps is a key factor in the microservices architecture [14].

In the field of the software industry, the introduction of the term DevOps, in 2008 [15], made it possible to articulate a set of practices that had already been taking place. In particular, the continuous integration practice that is based, among others, on automated tests [16], which represents one of the vital factors for its adoption [17], despite long-standing efforts to resolve this challenge [18], [19]. On the other hand, in the academic field, various literature review studies have been carried out where: (i) it is pointed out that the concept of DevOps is not completely defined [20]; (ii) the definitions, practices and benefits of DevOps are categorized [21]; (iii) the relevant aspects are determined [22], [23]; (iv) the factors that interrupt its adoption are identified [24]; (v) the influence on the product is presented [7]; and, (vi) in [2], a strong need to respond quickly to the market is reported and that DevOps helps to address this problem.

Since software testing is a critical factor for the adoption of DevOps [25], it should be reviewed how it is being applied in the reported cases. For this reason, this paper consolidates and classifies the literature on applied software testing in a DevOps context. The paper is organized as follows: in Section 2, the fundamental aspects of this study are presented; in Section 3, the Systematic Mapping Study (SMS) is described; in Section 4, the results of the SMS are presented; and, in Section 5, the conclusions are established.

2. Background

In this section, DevOps and software testing are briefly presented; as well as the works related to this study.

2.1 DevOps

DevOps integrates the teams that are usually separated (development and operations), focusing on delivering value quickly and continuously, based on 4 dimensions [22]: collaboration, automation, measurement and monitoring. In DevOps [4], it has extended the already known practices of agile methods, distributing them in 3 phases: construction phase, deployment phase, and operation phase. In addition, it incorporates some existing practices such as: continuous integration [26], continuous deployment [27], continuous delivery [28], and continuous testing [29].

2.2 Software Testing in Agile and DevOps Context

Software testing [30] are activities in the software development process to determine that the software has the expected behavior under a list of test cases. Tests can be categorized, according to [31]: (i) object of the test (unit, integration and system); and (ii) test objective (acceptance, installation, alpha, beta, regression, performance, security, load, recovery, bottom-out, interface, configuration, usability, and interaction).

In the agile context, agile tests have shown their benefits [32], [33], being necessary that the software-testers are present from the collection of requirements [34] and maintain fluid communication, both formal and informal, with the programmers [35].

3. Research Metodology

In this study, a Systematic Mapping Study (SMS) was performed. The SMS proposed by [36] is a research technique to identify and characterize all available studies on a given topic, using a reliable and verifiable methodology.

3.1 Scope and Research Questions

Software testing is one of the pillars to encourage good results in DevOps contexts [5], [8], and on which various publications have been made that require identification, studied and classified. For this reason, an SMS was performed with the purpose of identifying the levels of software tests that are being used in these contexts, as well as the authors, their evolution and the regions where the subject is being investigated, among others. The research questions and considerations for the answers are:

RQ-1 What is the evolution of the publication of papers on software testing in the DevOps contexts? The year of publication was taken as relevant data.

RQ-2 What kind of research has been done in software testing in DevOps? The types of research, adapted from [37], are: (i) survey/interview, (ii) case study, (iii) multiple case study, (iv) replication study, (v) review or literature mapping, and, (vi) background theory.

RQ-3 What kinds of proposals have been presented on software testing in DevOps? The types of proposals are an emerging classification and can be: methods, tools, frameworks.

RQ-4 What levels of software testing are used in DevOps? The possible test levels, depending on the object of the test, are: unit, integration, user, security and load/performance [31].

RQ-5 What programming languages and software testing tools are used in DevOps? Possible answers, at least initially, are: Java, C, PHP, JS, Xunit, Selenium.

RQ-6 In what types of applications are software testing used in the DevOps context? The possible answers, at least initially, are: web, desktop, console, mobile.

RQ-7 What infrastructure tools are used for software testing in DevOps? Possible answers are: Jenkins, Travis, Docker, AWS, Azure.

RQ-8 In what types of activities do software testing occur in DevOps? Possible answers are: Continuous Integration, Continuous Deployment, Continuous Delivery. Also, are security tests mentioned?

3.2 Search Query

Searches were performed according to a generated search string of the population (P) and intervention (I) as suggested [36]. The terms related to (P) are: DevOps, Continuous Integration, Continuous Testing, Continuous Deployment, and Continuous Delivery. The term related to I is: test. Then, the search string stayed as "P and I": "(DevOps OR "continuous integration" OR "continuous deployment" OR "continuous delivery" OR "continuous testing") AND test*". Although a string in English was searched, papers written in Spanish and Portuguese were also considered. Also, to allow for as many results as possible, the date was not restricted. The digital databases are: IEEE Xplore, SCOPUS, ScienceDirect. ACM Digital Library, Web of Science and Willey, selected for their scientific relevance and access to them.

3.3 Data Selection

The selection process was defined in four stages, where the inclusion criteria (IC) and exclusion criteria (EC) are applied (see Table 1); and according to [36] the quality assessment is omitted since relevant digital databases were chosen. The defined selection process has the following stages:

- In the first stage, obtaining the metadata, the EC.1 and IC.2 criteria are used, and the Parsif.al web application to facilitate some operations, such as discarding duplicate papers in the different databases.
- In the second stage, the title is read and EC.2 is applied, to rule out papers that are not related to the subject of software testing in the DevOps contexts.
- In the third stage, reading the summaries, IC.2, IC.3, EC.3 is applied.
- In the fourth stage, a quick reading is made of the content of the study to determine its relevance to the subject of software testing in DevOps contexts and criteria IC.2, IC.3, EC.3 and EC.4 are applied. Likewise, at this stage, the papers to which the full text is not available (EC.5) are withdrawn.

Table 1. Inclusion Criteria (IC) and Exclusion Criteria (EC)

Id	Criteria
IC.1	IC.1 Paper in indexed journals or conferences whose memories are indexed.
IC.2	IC.2 Paper with content in English, Spanish or Portuguese.
IC.3	IC.3 Paper that focuses on software testing in the DevOps context.
EC.1	EC.1 Duplicate article.
EC.2	EC.2 Paper outside the topic of software and DevOps.
EC.3	EC.3 Paper that does not mention software testing levels or strategies.
EC.4	EC.4 Secondary or tertiary articles.
EC.5	EC.5 Paper whose content is not available.

To extract the data, a file was created (see Table 2) to be used in a spreadsheet and collect the data from the papers on it.

Table 2. Structure of the data extraction form

Data	Detail	Question
Id Study	Unique identifier of the study created for the MSL.	General
Title	Title of the paper.	RQ-1
Author	List of authors of the paper.	RQ-1
The year	Year in which the paper was published.	RQ-1
Type of publication	Journal or conference where the paper was published.	RQ-1
Country	Country of affiliation of the authors.	RQ-1
Research type	Categorizes the type of research of the paper.	RQ-2
Context	Categorizes between the academic or industrial context of the paper.	RQ-2
Domain	Categorizes the business domain where the item was applied.	RQ-2

Type of proposal	Categorizes the type of proposal of the paper, if applicable.	RQ-3
Test Level	Categorizes the test levels mentioned in the paper.	RQ-3, RQ-4
Continuous phase	Categorizes the continuous phase mentioned in the paper.	RQ-4
Method	Identifies the method or good development practices.	RQ-4
Testing tool	Identifies the testing tool used.	RQ-5
Version Control	Identifies the tool used for code version management.	RQ-5
Programming language	Programming language mentioned in the paper.	RQ-5, RQ-6
Type App	Type of software developed in the paper.	RQ-6
Architecture type	Type of the architecture of the application developed in the paper.	RQ-6
Infrastructure tool	Collects the infrastructure tools used in the research presented in the paper.	RQ-7
Security	Identifies if the paper mentions the security tests	RQ-8
Teams in DevOps	Identifies if the paper addresses Devs, Ops or both teams.	RQ-8

4. Results

The searches in the considered databases were carried out between June and July 2021. For each database, the search string was adapted according to its own rules (see Table 3). Of the 3,312 papers found, it was processed stage by stage until reaching a total of 299 primary studies. The process was based on the inclusion and exclusion criteria according to the study planning. Table 4 shows the number of papers that remained after each stage. In addition, 15 (5%) papers were withdrawn because the full text was not available, even after having searched different sources. The list of primary studies is available in Appendix A.

Table 3. Database search string

Source	Search string	Quantity
IEEE	(("All Metadata":Devops) OR ("All Metadata":"Continuous Integration") OR ("All Metadata":"Continuous Deployment") OR ("All Metadata":"Continuous Delivery") OR ("All Metadata":"Continuous Testing")) AND (("All Metadata":Test*))	529
Scopus	TITLE-ABS-KEY ((devops OR "Continuous Integration" OR "Continuous Deployment" OR "Continuous Delivery" OR "Continuous Testing") AND test*)	1,561
ACM	Title: ((Devops OR "Continuous Integration" OR "Continuous Deployment" OR "Continuous Delivery" OR "Continuous Testing") AND Test*) OR Abstract:((Devops OR "Continuous Integration" OR "Continuous Deployment" OR "Continuous Delivery" OR "Continuous Testing") AND Test*) OR Keyword:((Devops OR "Continuous Integration" OR "Continuous Deployment" OR "Continuous Delivery" OR "Continuous Testing") AND Test*)	246
Science Direct	Title-keyword-abstract (Devops OR "Continuous Integration" OR "Continuous Deployment" OR "Continuous Delivery" OR "Continuous Testing") AND Test	462
Web of Science	TITLE-ABS-KEY ((devops OR "Continuous Integration" OR "Continuous Deployment" OR "Continuous Delivery" OR "Continuous Testing") AND test*)	432
Willey	TITLE-ABS-KEY ((devops OR "Continuous Integration" OR "Continuous Deployment" OR "Continuous Delivery" OR "Continuous Testing") AND test*)	82
Total		3,312

Table 4. Search results by stage

Procedure	Selection Criteria	Total
First stage	EC.1, IC.1	1,179
Second stage	EC.2	928
Third stage	IC.2, IC.3, EC.3	344
Fourth Stage	IC.2, IC.3, EC.3, EC.4, EC5	299

4.1 RQ1 What is the evolution of the publication of papers on software testing in the DevOps contexts?

From the selected primary studies, from 2011 to Jun-2021 (see Figure 1a), it is observed that the level of publications has been increasing from the beginning, which shows the importance of software testing in DevOps contexts and that coincides with those indicated by [38]. In addition, this growth is expected to continue in the following years.

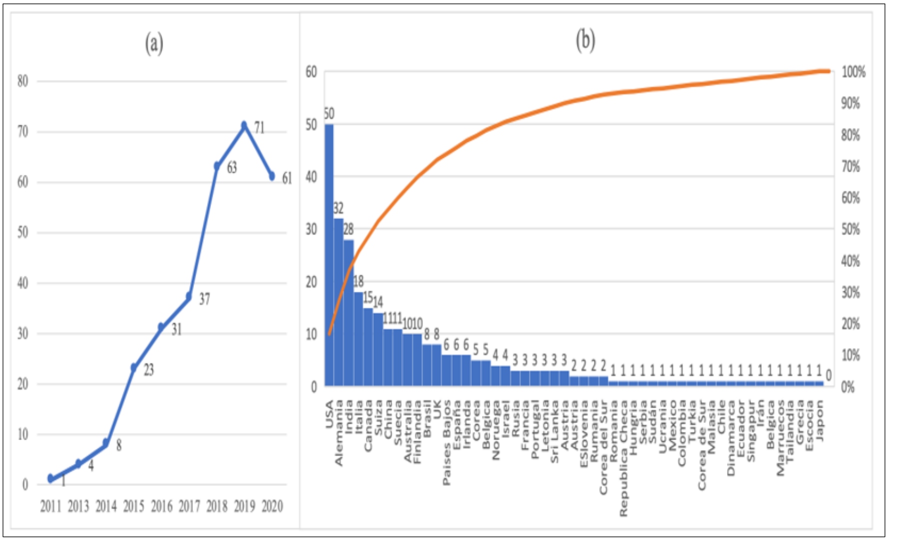

Fig 1. Evolution of publications per year (a), and publications by country (b) in DevOps software testing

Although the topic of DevOps is of global importance, it can be seen (see Figure 1b) that according to the Pareto rule 80% of the studies are concentrated in 16 countries: USA (16.7%), Germany (10.7%), India (9.4%), Italy (6%), Canada (5%), Switzerland (4.7%), China (3.7%), Sweden (3.7%) Australia (3.3%), Finland (3.3%) and Brazil (2.7%), UK (2.7%), the Netherlands (2%), Spain (2%), Ireland (2%), Korea (1.7%) and Belgium (1.7%).

On the other hand, the publication media where they have been published 4 or more primary studies are 14 media and are presented in Table 5.

Table 5. Frequency of primary studies by means of communication, which have 4 or more publications

Venue	Count
Lecture Notes in Computer Science	11
Communications in Computer and Information Science	9
CEUR Workshop Proceedings	9
International Conference on Software Engineering	9
ACM International Conference Proceeding Series	7
International Workshop on Quality-Aware DevOps (QUDOS)	7
IEEE Software	5
Euromicro Conference on Software Engineering and Advanced Application (SEAA)	5
Information and Software Technology	5
IEEE International Conference on Software Maintenance and Evolution (ICSME)	5
Advances in Intelligent Systems and Computing	4
International Conference on Software Testing, Verification and Validation (ICSTW)	4
International Conference on Software Analysis, Evolution, and Reengineering (SANER)	4
Journal of System and Software	4

4.2 RQ2 What types of research have been done on software testing in DevOps?

From the primary studies, on types of research (see Figure 2a), there are two predominant types of research (78.6%): 136 study cases (45.5%) and 99 experiments (33.1%); which are mostly reported in the industry. This orientation, towards the more empirical side, makes sense, since the cases and experiments of integrating Dev and Ops work teams materialize in real projects. This result coincides with the study by [39], who also found a high percentage (20%) of papers at the industry level. Of the remaining group of research types, it can be pointed out that those related to opinion-research allow concepts, ideas, lessons to be proposed when dealing with software testing in DevOps. Likewise, the result of the research context shows that 213 (71.2%) according to Figure 2b, are papers in the industry, compared to 29 (9.7%) are papers in academia; which reinforces the idea of the previous result.

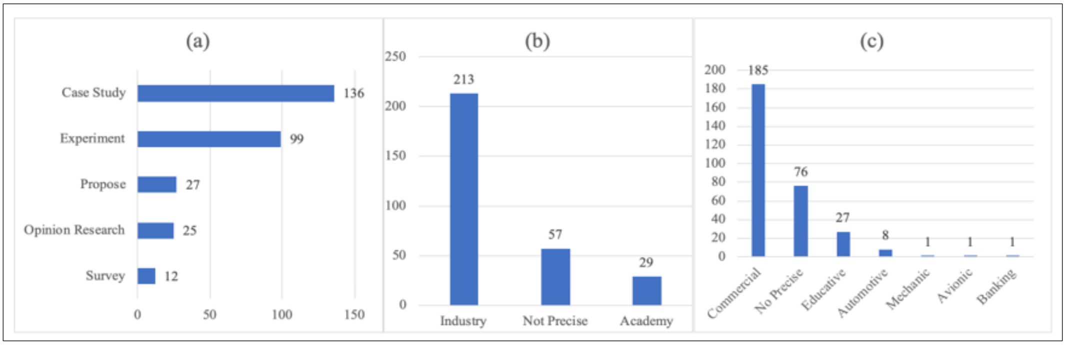

Fig 2. Distribution of primary studies of software testing in the DevOps context, by: (a) research type, (b) research context, and (c) application domain

Finally, from the perspective of the application domain (see Figure 2c), 185 (61.8%) papers have been applied to commercial solutions, that is, applications to sell products, rent services, etc. Likewise, an interesting focus is seen in the education sector, where 27 (9%) primary studies have focused on applications for education (support for the teaching/learning process).

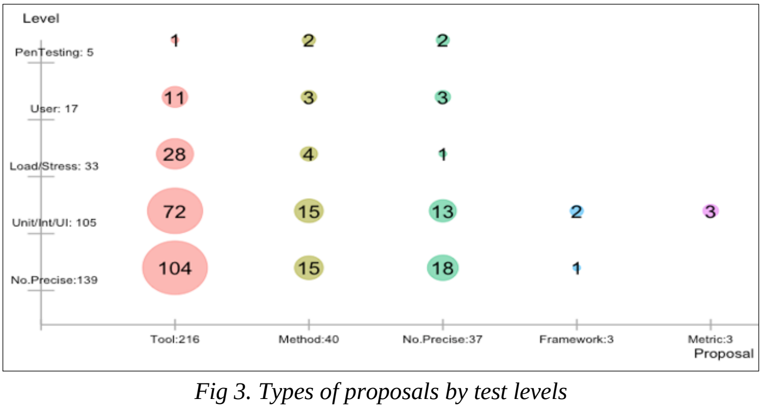

Fig 3. Types of proposals by test levels

4.3 RQ-3 What kinds of proposals have been presented on software testing in DevOps?

In Figure 3, it can be seen that 216 (72.2% primary studies) propose tools to support DevOps contexts, incorporating software testing as part of them. Furthermore, 40 (14%) and 3 (1%) papers propose methods and frameworks respectively to support testing work. These results are in agreement with the results obtained in the study by [40], they point out that tools and frameworks have been proposed and that most are based on unit tests and automated integration.

4.4 RQ-4 What levels of software testing are used in DevOps?

In relation to the levels of software testing used in DevOps (see Figure 4a), the response of "not precise" are 139 papers (46.5%). Despite this, these works do indicate that software testing is a DevOps necessity, but they do not specify the levels of testing in the DevOps context. In the case of the primary studies, which do indicate the levels of proof, it follows that: (i) 122 papers (35.1%) have reported unit and user interface tests; (ii) 33 papers (11%) have reported load and stress; and, (iii) the rest are user tests and penetration testing (pen-testing). The work of [41] and [42] agree that unit and integration tests are among the most studied. Likewise, [41] adds functional, load and stress tests as the most studied with 63.6% of the total studies reviewed; and, they consider that security tests are much less studied with 3.6%. According to reviews from [43] and [44], GUI and accessibility tests are still pending challenges in continuous contexts.

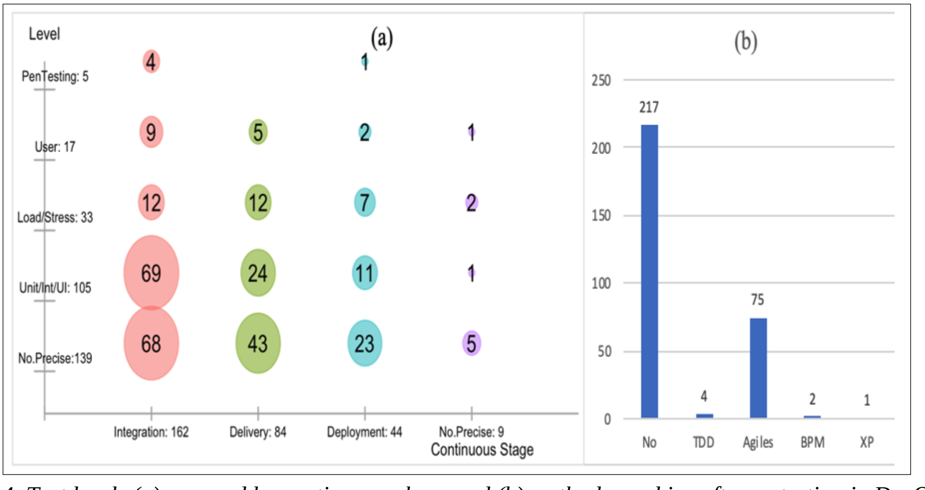

Fig 4. Test levels (a) grouped by continuous phase and (b) methods used in software testing in DevOps

According to this Figure 4a, in relation to the opportunity in the use of software tests in DevOps, it can be pointed out that 162 papers (54.2%) have been applied during continuous integration; which, at first glance, turns out to be the natural space for testing. However, 84 (28.1%) papers have also been identified that have used tests to solve activities in continuous delivery and 44 (14.7%) in continuous deployment, which shows that 42.8% of the tests are outside continuous integration.

According to Figure 4b, in relation to the software development methodology, from the primary studies, it has been determined as "not precise" in 217 (72.6%) papers. In the other cases, it shows 75 (25.1%) papers used agile methodologies, and more explicitly points to TDD and XP with 5 (1.7%) papers, considering both. In particular, in the case of TDD studies, they consider the method important for the success of software testing in DevOps. This suggests that, for now, although TDD is a very good method, there are few studies in this type of context. Similarly, the studies by [43] and [39] consider that TDD would help to better conceptualize testing strategies and mitigate system design errors for help continuous testing.

4.5 RQ-5 What programming languages and software testing tools are used in DevOps?

Due to the nature and objectives of the primary studies, in many cases, programming languages, testing support tools, and version control tools are not required. In the case of programming languages (see Figure 5), it is observed that Java is the most reported language with 90 (30%) papers. In the case of test support tools, Junit with 25 (8.4%) and Selenium with 13 (4.3%) papers are the most reported. Finally, in the case of version control tools, Git is mentioned in 179 (59.9%) of papers.

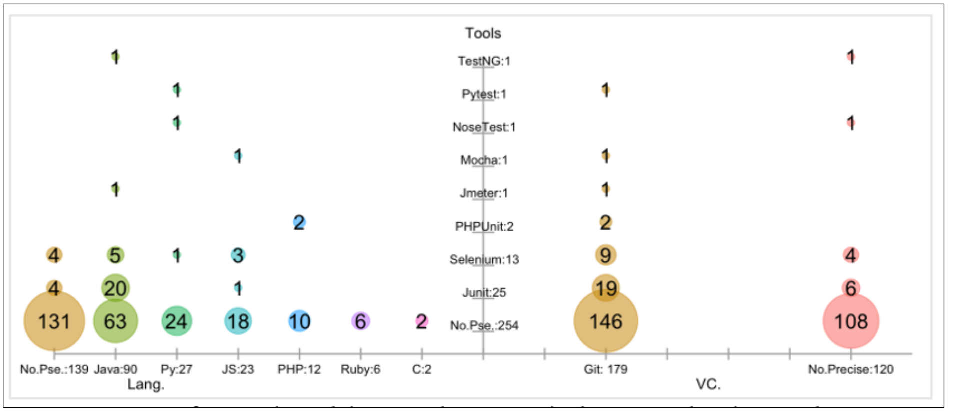

Fig 5. Software testing tools in DevOps by programming languages and version control

In the review of [39], it is agreed that Junit, Selenium and Git are the most frequent tools in the DevOps software testing application. In addition [39], considers NUnit among the most frequent, however, of the selected primary studies, no reference to said tool was found.

According to Figure 6a, Java is the most used language over time with an average of 13 papers per year, while Python has been considered in recent years, with an average of 4 papers per year as presented in Figure 6b.

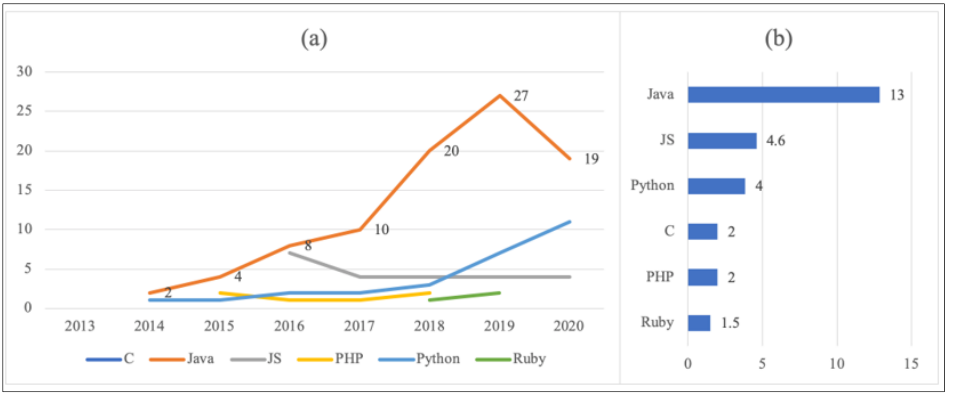

Fig 6. Programming languages in software testing over time (a) and average per year (b)

4.6 RQ-6 In what types of applications and architectures is software testing used in the DevOps context?

In relation to the types of applications where software tests are used in DevOps (see Figure 7a), reported in the primary studies, web applications with 219 (71.9%) papers have to be the most reported applications, and to a lesser extent, mobile applications with 13 (4.3%) papers. The identified console applications are reported for cases in which they apply machine learning concepts and use this type of application to display the results. In relation to the types of architecture (see Figure 7b), the primary studies indicate that 134 (44.8%) are of the MVC type and 52 (17.4%) are of the SOA type, and especially, of the latter, 14 studies report REST as a technology communication. Despite this, 85 (28.4%) papers which represent a high percentage that does not need it.

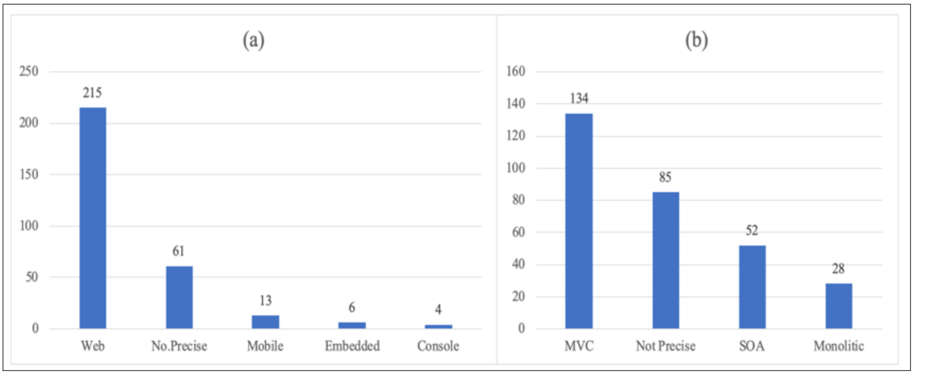

Fig 7. Type of applications (a) and architectures (b) in software testing in DevOps

For [39], 33% of their studies found are web applications, being the most frequent for DevOps software tests; and it also agrees that few researches, that is, 1.6%, are reported on embedded applications.

4.7 RQ-7 What tools are used for software testing in DevOps?

Regarding the tools, it can be pointed out that they are not reported in 111 (37.1%) of the studies (see Figure 8a). In the studies that are reported, Jenkins is present in 92 (30.8%) primary studies. This result coincides with the review by [39] who also found Jenkins to be the most studied tool. In the industry, Jenkins is known as a very versatile tool that allows you to automatically run tests written by the development team, whether they are unit, integration, UI, loading and others. Crossing these results with the years of publication, according to Figure 8b, it can be seen that Jenkins has been increasingly reported in primary studies since 2013. It is also observed, according to Figure 8c, in relation to the average of the publications of papers per year, which Docker has about 6.8 papers/year since 2016, AWS is 3.3 since 2018 and GitLab is 4.8 since 2017. This result shows that Docker is being recurrently reported in the selected primary studies. In the interviews conducted by [42], containerization is mentioned as one of the most studied solutions in continuous delivery.

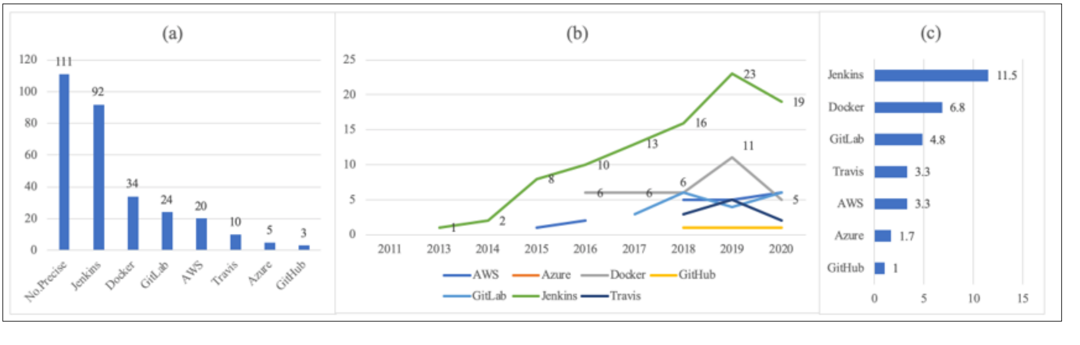

Fig 8. Software testing tools in DevOps (a) by years (b) and, distributed over time and average per year (c)

In Figure 9, it can be seen that Java appears in 40 (13.4%) primary studies, being used in conjunction with Jenkins, becoming the most frequent language for Jenkins. Furthermore, in the case of Java, 19 (21%) papers have been applied in industry and 3 (4%) in the academic context.

Figure 10 shows that 63 (21%) Jenkins primary studies have been studied in the industry and Docker with 34 (7.4%) is behind Jenkins. This shows that Jenkins is the most studied software testing tool in DevOps contexts.

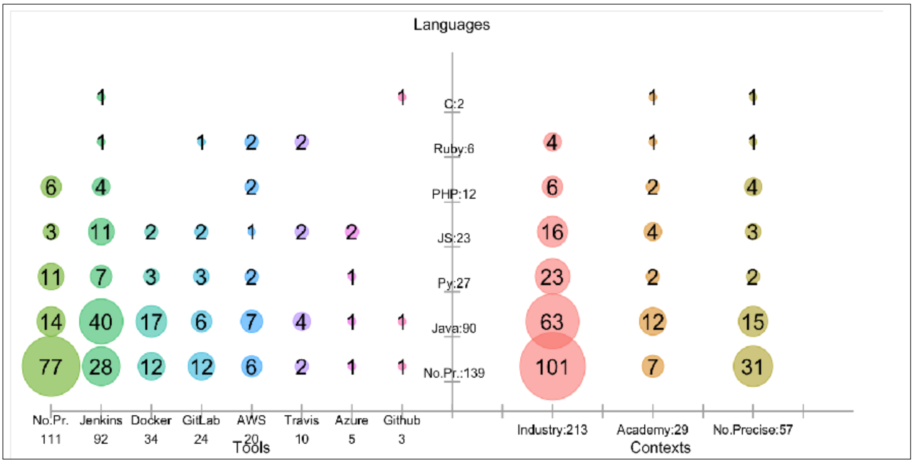

Fig 9. Programming languages and tools in DevOps software testing

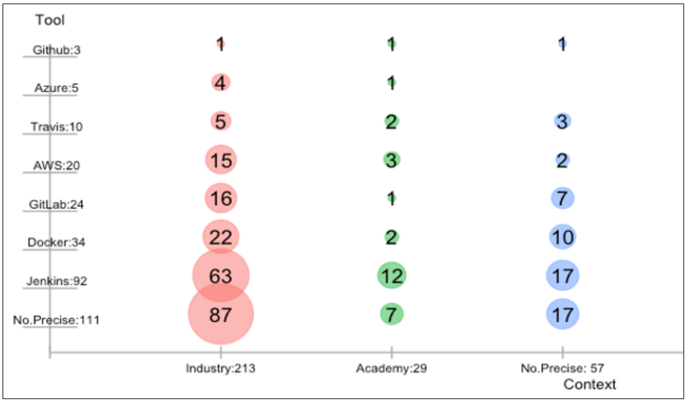

Fig 10. Tools in DevOps for software testing according to its context

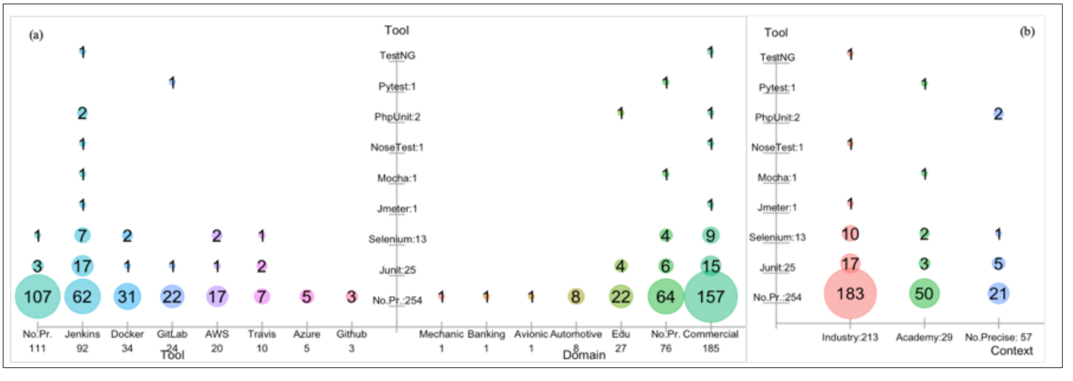

Fig 11. Test tools, infrastructure in DevOps (a) and application context (b)

Figure 11a shows that although Java was often used as a programming language, Junit was not necessarily mentioned in these studies. However, Junit does appear as the most mentioned testing tools in the primary studies. In addition, these, for the most part, 185 (61.8%) papers have been applied in commercial business domains. Figure 11b confirms that Junit is also applied in the industrial context.

4.8 RQ-8 In what types of activities do software testing occur in DevOps? Also, are safety tests mentioned?

According to Figure 12, the selected primary studies show that more than 230 (75%) have concerned themselves with both what is needed in development and in operation, be it with tools, methods, frameworks or suggestions. 60 (20%) papers have studied the specific activities of development teams. Finally, only 9 (3%) have focused solely on operating activities.

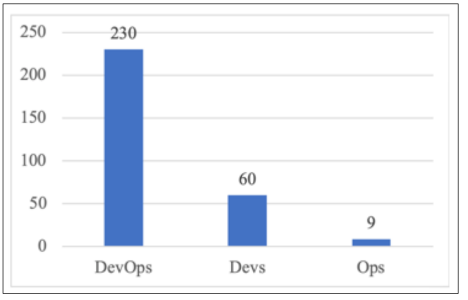

Fig 12. Software testing in DevOps phases

According to Figure 13, more than half of the papers found, that is 169 (56.6%), mention application security as an important factor in the DevOps contexts, despite the fact that there are only 15 application testing papers. penetration (see Figure 4a). These findings are in the same direction as that indicated by [45], [46] and [39], about the need to study more about the security issues in Devops contexts, also known as DevSecOps. This allows you to integrate these types of tests into your development tools.

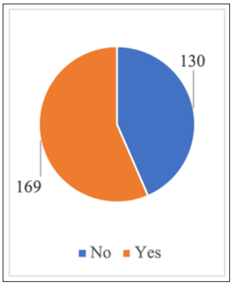

Fig 13. Mention of security in software testing in DevOps

4.9 Threats to Validity

The analysis of the threats to validity was based on the work and questions proposed by [47].

- **Study Selection Validation**. During the planning of the research, in order to ensure the proper identification of all relevant studies, the following was carried out: (i) a preliminary search to identify a relevant set of 20 "test" papers that allowed validating the research questions research, the search chain and selection process; then, (ii) Population and Intervention was used, according to [36], to structure a convenient search chain, actually an iterative task; (iii) a chain test was carried out with the "test" papers, and a check was made if the data obtained from said "test" papers allowed to answer the research questions; and (iv) it was established to work with 6 relevant digital databases.

  The selection was made using the methodology proposed by [36]. Duplicate papers were filtered in the exclusion criteria by DOI, title, authors and year. Inclusion/exclusion criteria were discussed by the authors based on similar research. At each stage, a general criterion was applied, that, when in doubt of acceptance or rejection, acceptance is chosen so that the paper is subsequently evaluated. This reloads the next stage, but reduces the risk of deleting relevant papers.

- **Data Validation**. Taking into account what was indicated in [36], it was decided to only work with relevant digital databases. These databases usually already have evaluation schemes for the journals and reports of events that they incorporate. In this context, it was decided not to make a quality assessment in the selection process.

  In the first 100 primary studies, a first consolidation was performed, and these studies were discussed between both authors. The evaluation also made it possible to note the relationship of the results with the subject under research. The classification schemes were proposed during the planning of the SMS and were refined, in some cases, during the data extraction. Additionally, the verification of the selection was carried out by the second author in a sample manner.
- **Research Validation**. Both authors are related to the research topic and the second author has more experience in secondary studies. The work carried out is replicable since all the data collected during the research are publicly accessible, phase by phase, as well as the general search string and the personalized ones for each database. At the beginning of the research, it was determined by the research questions and the results of the first stages, that the research would be a systematic mapping of literature due to the need to classify software tests in DevOps contexts. The research can be generalized to all DevOps contexts because it collects the information without considering specific regions, places or periods. In addition, it considers primary studies from both industry and academia.

5. Conclusions

This research presents a Systematic Mapping Study (SMS) on software testing in the DevOps context. The SMS is based on the proposal of [36]. In the selection process, 3,312 studies were obtained and at the end of the process, 299 were selected as primary studies. Based on the data obtained from the primary studies, it was possible to answer the 8 research questions raised.

The interest of research on software testing in the DevOps context is current and continuously growing since 2011. It is also appreciated that it is a global interest, in particular, considering that there are 16 countries from 3 regions (America, Europe and Asia) who have published 239 (80%) of the studies. In accordance with the origin and empirical nature of DevOps, the majority of primary studies, which mean 235 (78.6%) are of the type of case studies and experiments. Likewise, 213 of these studies have been carried out in industry contexts (71.2%) and 185 in commercial applications (61.8%). In addition, 216 (72.2%) primary studies have proposed tools that support test automation. The results also indicate that software testing is considered an important factor in DevOps issues, but what levels of testing are being used are not specified. But, in those that do specify, unit and integration tests are the most studied, and to a lesser extent, user, load and stress and security tests.

In relation to technology, such as programming language and test support tools, it can be noted that these issues are not explicitly reported in primary studies. In the cases that do report, it is pointed out that Java is the most reported language with 90 (30%) both in academic and industrial environments; and in the case of test development tools, 25 papers, that is means, more than 8.3% have been reported to Junit. Other reported programming languages are: Python, Js and PHP respectively. Furthermore, it has to be mentioned that Java is the most reported language in primary studies over time, with an average of 13 papers per year.

The most studied types of applications are those of the Web type with 216 (72.2%), based on both SOA and MVC. One of the most reported tools is Jenkins for both continuous integration, continuous deployment and continuous delivery. In addition, tools such as: Travis, Docker, GitLab, Github and AWS are also reported, showing that the studies carried out are applied to current market tools.

The results of this research show research opportunities in software testing for the DevOps contexts. Likewise, it is clear that training in automated software testing skills could help small companies to compete in the world market with quality.

References / Список литературы

[1] Samarawickrama S.S., Perera I. Continuous scrum: A framework to enhance scrum with DevOps. In Proc. of the 17th International Conference on Advances in ICT for Emerging Regions, 2017, pp. 19-25.

[2] Nicolau de França B.B., Jeronimo H., Travassos G.H. Characterizing DevOps by hearing multiple voices. In Proc. of the XXX Brazilian Symposium on Software Engineering, 2016, pp. 53-62.

[3] Elliot S. DevOps and the Cost of Downtime: Fortune 1000 Best Practice Metrics Quantified. IDC, 2015, 13 p.

[4] Ebert C., Gallardo G. et al. «DevOps», IEEE Software, vol. 33, issue 3, 2016, pp. 94-100.

[5] Riungu-Kalliosaari L, Mäkinen S. et al. DevOps Adoption Benefits and Challenges in Practice: A Case Study. Lecture Notes in Computer Science, vol. 10027, 2016, pp. 590-597.

[6] Stillwell M., Coutinho J.G.F. A DevOps approach to integration of software components in an EU research project. In Proc. of the 1st International Workshop on Quality-Aware DevOps, 2015, pp. 1-6.

[7] Céspedes D., Angeleri P. et al. Software Product Quality in DevOps Contexts: A Systematic Literature Review. Advances in Intelligent Systems and Computing, vol. 1071, Springer, 2020, pp. 51-64.

[8] Perera P., Silva R., Perera I. Improve software quality through practicing DevOps. In Proc. of the 17th International Conference on Advances in ICT for Emerging Regions, 2017, pp. 13-18.

[9] Elberzhager F., Arif T. et al. From Agile Development to DevOps: Going Towards Faster Releases at High Quality – Experiences from an Industrial Context. Lecture Notes in Business Information Processing, vol. 269, 2017, pp. 33-44.

[10] Jones S., Noppen J., Lettice F. Management challenges for devops adoption within UK SMEs. In Proc. of the 2nd International Workshop on Quality-Aware DevOps, 2016, pp. 7-11.

[11] Soni M. End to End Automation on Cloud with Build Pipeline: The Case for DevOps in Insurance Industry, Continuous Integration, Continuous Testing, and Continuous Delivery. In Proc. of the IEEE International Conference on Cloud Computing in Emerging Markets, 2015, pp. 85-89.

[12] Senapathi M., Buchan J., Osman H. DevOps capabilities, practices, and challenges: Insights from a case study. In Proc. of the International Conference on Evaluation and Assessment in Software Engineering, 2018, pp. 57-67.

[13] Chen L. Continuous Delivery: Overcoming adoption challenges. Journal of Systems and Software, vol. 128, 2017, pp. 72-86.

[14] Valdivia J.A., Lora-González A. et al. Patterns Related to Microservice Architecture: a Multivocal Literature Review. Programming and Computer Software, vol. 46, issue 8, 2020, pp. 594-608 / Вальдивия Х.А., Лора-Гонсалес А и др. Паттерны микросервисной архитектуры: многопрофильный обзор литературы. Труды ИСП РАН, том 33, вып. 1, 2021 г., стр. 81-96. DOI: 10.15514/ISPRAS-2021-33(1)-6.

[15] Debois P. Agile Infrastructure & Operations. In Proc. of the Agile 2008 Conference, 2008, pp. 202-207.

[16] Virmani M. Understanding DevOps & bridging the gap from continuous integration to continuous delivery. In Proc. of the 5th International Conference on Innovative Computing Technology, 2015, pp. 78-82.

[17] Mullaguru S.N. Changing Scenario of Testing Paradigms using DevOps--A Comparative Study with Classical Models. Global Journal of Computer Science and Technology, vol. 15, issue 2, 2015, pp. 23-27.

[18] Chernonozhkin S.K. Automated Test Generation and Static Analysis. Programming and Computer Software, vol. 27, issue 2, 2001, pp. 86-94 / Черноножкин С.К. Задача автоматического построения тестов и статистический анализ. Программирование, том 27, вып. 2, 2001 г., стр. 47-59.

[19] Kuliamin V.V., Petrenko A.K. et al. The UniTesK Approach to Designing Test Suites. Programming and Computer Software, vol. 29, issue 6, 2003, pp. 310-322 / Кулямин В.В., Петренко А.К. и др. Подход UniTesK к разработке тестов. Программирование, том 29, вып. 6, 2003 г., стр. 25-43.

[20] Jabbari R., Ali N., Petersen K. What is DevOps?: A Systematic Mapping Study on Definitions and Practices. In Proce. of the Scientific Workshop of XP2016, 2016, article no. 12, 11 p.

[21] Ghantous G.B., Gill A. DevOps: Concepts, Practices, Tools, Benefits and Challenges. In Proc. of the 21st Pacific Asia Conference on Information Systems (PACIS), 2017, article no. 96, 13 p.

[22] Lwakatare L.E., Kuvaja P., Oivo M. Dimensions of DevOps. Lecture Notes in Business Information Processing, vol. 212, 2015, pp. 212–217.

[23] Katal A., Bajoria V., Dahiya S. DevOps: Bridging the gap between development and operations. In Proc. of the 3rd International Conference on Computing Methodologies and Communication, 2019, pp. 1-7.

[24] Kamuto M.B., Langerman J.J. Factors inhibiting the adoption of DevOps in large organisations: South African context. In Proc. of the 2nd IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology, 2017, pp. 48-51.

[25] Zimmerer P. Strategy for Continuous Testing in iDevOps. In Proc. of the IEEE/ACM 40th International Conference on Software Engineering, 2018, pp. 532-533.

[26] Fowler M. Continuous Integration. 2006. Available at: https://www.martinfowler.com/articles/continuousIntegration.html, accessed 28-nov-2020.

[27] Parnin C., Helms E. et al. The Top 10 Adages in Continuous Deployment. IEEE Software, vol. 34, issue 3, 2017, pp. 86-95.

[28] Fowler M. Continuous Delivery. 30-may-2013. Available at:: https://martinfowler.com/bliki/ContinuousDelivery.html, accessed 28-nov-2020.

[29] Fitzgerald B., Stol K.J. Continuous software engineering and beyond: Trends and challenge. In Proc. of the 1st International Workshop on Rapid Continuous Software Engineering, 2014, pp. 1-9.

[30] ISO/IEC/IEEE, «ISO/IEC/IEEE 24765:2017 Systems and software engineering – Vocabulary. Geneva, 2017.

[31] Guide to the Software Engineering Body of Knowledge (SWEBOK), Version 3.0. IEEE Computer Society, 2014, 339 p.

[32] Gupta R.K., Manikreddy P., Gv A. Challenges in adapting agile testing in a legacy product. In Proc. of the 11th IEEE International Conference on Global Software Engineering, 2016, pp. 104-108.

[33] Jeeva Padmini K.V., Kankanamge P.S. et al. Challenges faced by agile testers: A case study. In Proc. of the Moratuwa Engineering Research Conference, 2018, pp. 431-436.

[34] Coutinho J.C.S., Andrade W.L., Machado P.D.L. Requirements engineering and software testing in agile methodologies: A systematic mapping. In Proc. of the XXXIII Brazilian Symposium on Software Engineering, 2019, pp. 322-331.

[35] Cruzes D.S., Moe N.B., Dyba T. Communication between developers and testers in distributed continuous agile testing. In Proc. of the 11th IEEE International Conference on Global Software Engineering, 2016, pp. 59-68.

[36] Petersen K., Vakkalanka S., Kuzniarz L. Guidelines for conducting systematic mapping studies in software engineering: An update. Information and Software Technology, vol. 64, 2015, pp. 1-18.

[37] Kuhrmann M., Diebold P., Münch J. Software process improvement: A systematic mapping study on the state of the art. PeerJ Computer Science, issue 5, 2016, article no. 62, 38 p.

[38] Pinto G., Castor F. et al. Work practices and challenges in continuous integration: A survey with Travis CI users. Software: Practice and Experience, vol. 48, issue 12, 2018, pp. 2223-2236.

[39] Shahin M., Ali Babar M., Zhu L. Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices. IEEE Access, vol. 5, 2017, pp. 3909-3943.

[40] Alnafessah A., Gias A.U. et al. Quality-Aware DevOps Research: Where Do We Stand. IEEE Access, vol. 9, 2021, pp. 44476-44489.

[41] Mascheroni M.A., Irrazábal E. Continuous Testing and Solutions for Testing Problems in Continuous Delivery: A Systematic Literature Review. Computación y Sistemas, vol. 22, issue 3, 2018, pp. 1009-1038.

[42] Shahin M., Babar M.A. et al. Beyond Continuous Delivery: An Empirical Investigation of Continuous Deployment Challenges. In Proc. of the International Symposium on Empirical Software Engineering and Measurement, 2017, pp. 111-120.

[43] Laukkanen E., Itkonen J., Lassenius C. Problems, causes and solutions when adopting continuous delivery – A systematic literature review. Information and Software Technology, vol. 82, 2017, pp. 55-79.

[44] Sane P. A Brief Survey of Current Software Engineering Practices in Continuous Integration and Automated Accessibility Testing. In Proc. of the International Conference on Wireless Communications, Signal Processing and Networking, 2021, pp. 130-134.

[45] Rajapakse R.N., Zahedi M. et al. Challenges and solutions when adopting DevSecOps: A systematic review. Information and Software Technology, vol. 141, 2021, article no. 106700, 27 p.

[46] Daoudagh S., Lonetti F., Marchetti E. Continuous Development and Testing of Access and Usage Control. In Proc. of the European Symposium on Software Engineering, 2020, pp. 51-59.

[47] Ampatzoglou A., Bibi S. et al. Identifying, categorizing and mitigating threats to validity in software engineering secondary studies», Information and Software Technology, vol. 106, 2019, pp. 201-230.

Appendix A. List of Primary Studies

Table A. Primary Studies

ID	Authors	Year	Title
S01	K. Priyadarsini and E. Fantin Irudaya Raj and A. Yasmine Begum and V. Shanmugasundaram	2020	Comparing DevOps procedures from the context of a systems engineer
S02	Casola V., De Benedictis A., Rak M., Salzillo G.	2020	A cloud secdevops methodology: From design to testing
S03	Fehlmann T., Kranich E.	2020	A Framework for Automated Testing
S04	Amaral C.J., Kampik T., Cranefield S.	2020	A framework for collaborative and interactive agent-oriented developer operations
S05	C. Klammer; J. Gmeiner	2020	A Lightweight Customized Build Chain Visualization Approach Applied in Industry
S06	Casola V., De Benedictis A., Rak M., Villano U.	2020	A methodology for automated penetration testing of cloud applications
S07	R. Guntha; S. N. Rao; H. Muccini; M. Vinodini Ramesh	2020	A Novel Paradigm for Rapid Yet Robust Continuous Delivery of Software for Disaster Management Scenarios
S08	Hsu W., Lin J.-S., Chen Y.-C., Wang C.-Y., Huang C.-T.	2020	An Automatic Software Quality and Function Assurance Case Study for Agile
S09	Cai Y.X., Shang Y.F., Tan Y.X., Tang Z.W., Zhao B.	2020	An Effective Solution for Application Orchestration
S10	R. W. Macarthy; J. M. Bass	2020	An Empirical Taxonomy of DevOps in Practice
S11	A. Kanchana; C. Murthy B.N.	2020	Automated Development and Testing of ECUs in Automotive Industry with Jenkins
S12	Avritzer A.	2020	Automated scalability assessment in devops environments
S13	Rakshith M.N., Shivaprasad N.	2020	Build Optimization Using Jenkins
S14	Karlaš B., Interlandi M., Renggli C., Wu W., Zhang C., Mukunthu Iyappan Babu D., Edwards J., Lauren C., Xu A., Weimer M.	2020	Building Continuous Integration Services for Machine Learning
S15	G. Ambrosino; G. B. Fioccola; R. Canonico; G. Ventre	2020	Container Mapping and its Impact on Performance in Containerized Cloud Environments
S16	S. H. Reiterer; S. Balci; D. Fu; M. Benedikt; A. Soppa; H. Szczerbicka	2020	Continuous Integration for Vehicle Simulations
S17	L. Gota; D. Gota; L. Miclea	2020	Continuous Integration in Automation Testing
S18	Gorsky S.A.	2020	Continuous integration, delivery, and deployment for scientific workflows in Orlando Tools
S19	T. Rangnau; R. v. Buijtenen; F. Fransen; F. Turkmen	2020	Continuous Security Testing: A Case Study on Integrating Dynamic Security Testing Tools in CI/CD Pipelines
S20	M. Johnson; D. Cummings; B. Leinwand; C. Elsberry	2020	Continuous Testing and Deployment for Urban Air Mobility
S21	Angara J., Prasad S.	2020	Continuous testing real-time health analytics dashboard
S22	Doležel M.	2020	Defining testops: Collaborative behaviors and technology-driven workflows seen as enablers of effective software testing in devops
S23	Török M., Pataki N.	2020	DevOps dashboard with heatmap
S24	Yang D., Wang D., Yang D., Dong Q., Wang Y., Zhou H., Daocheng H.	2020	DevOps in practice for education management information system at ECNU
S25	Laaber C., Würsten S., Gall H.C., Leitner P.	2020	Dynamically reconfiguring software microbenchmarks: Reducing execution time without sacrificing result quality
S26	Al-Sabbagh K.W., Staron M., Ochodek M., Meding W.	2020	Early prediction of test case verdict with bag-of-words vs. word embeddings
S27	Couto L.D., Tran-Jørgensen P.W.V., Nilsson R.S., Larsen P.G.	2020	Enabling continuous integration in a formal methods setting

S28	Karakasis V., Manitaras T., Rusu V.H., Sarmiento-Pérez R., Bignamini C., Kraushaar M., Jocksch A., Omlin S., Peretti-Pezzi G., Augusto J.P.S.C., Friesen B., He Y., Gerhardt L., Cook B., You Z.-Q., Khuvis S., Tomko K.	2020	Enabling Continuous Testing of HPC Systems Using ReFrame
S29	Vassallo C., Proksch S., Zemp T., Gall H.C.	2020	Every build you break: developer-oriented assistance for build failure resolution
S30	Luzar A., Stanovnik S., Cankar M.	2020	Examination and comparison of tosca orchestration tools
S31	Meinicke J., Wong C.-P., Vasilescu B., Kästner C.	2020	Exploring differences and commonalities between feature flags and configuration options
S32	Demeyer S., Parsai A., Vercammen S., van Bladel B., Abdi M.	2020	Formal Verification of Developer Tests: A Research Agenda Inspired by Mutation Testing
S33	M. Mazkatli; D. Monschein; J. Grohmann; A. Koziolek	2020	Incremental Calibration of Architectural Performance Models with Parametric Dependencies
S34	Shin J.-S., Kim J.	2020	K-one playground: Reconfigurable clusters for a cloud-native testbed
S35	P. Batra; A. Jatain	2020	Measurement Based Performance Evaluation of DevOps
S36	Eismann S., Bezemer C.-P., Shang W., Okanović D., Van Hoorn A.	2020	Microservices: A performance tester's dream or nightmare?
S37	van den Heuvel W.-J., Tamburri D.A.	2020	Model-driven ml-ops for intelligent enterprise applications: vision, approaches and challenges
S38	Shahin M., Babar M.A.	2020	On the role of software architecture in DevOps transformation: An industrial case study
S39	Mirhosseini S., Parnin C.	2020	Opunit: Sanity Checks for Computing Environments
S40	Gias A.U., Van Hoorn A., Zhu L., Casale G., Düllmann T.F., Wurster M.	2020	Performance engineering for microservices and serverless applications: The RADON approach
S41	J. Chen	2020	Performance Regression Detection in DevOps
S42	Raj P., Sinha P.	2020	Project management in era of agile and devops methodolgies
S43	Cheriyan A., Gondkar R.R., Babu S.S.	2020	Quality Assurance Practices and Techniques Used by QA Professional in Continuous Delivery
S44	M. Huang; W. Fan; W. Huang; Y. Cheng; H. Xiao	2020	Research on Building Exploitable Vulnerability Database for Cloud-Native App
S45	C. Fayollas; H. Bonnin; O. Flebus	2020	SafeOps: A Concept of Continuous Safety
S46	Vishnu Vardhan Reddy B.S., Swamy B.K., Sai S.P.S., Kiran K.V.D.	2020	Securing web application by using qualitative research methods for detection of vulnerabilities in any application of DevSecOps
S47	Petrovic N., Tosic M.	2020	SMADA-Fog: Semantic model driven approach to deployment and adaptivity in fog computing
S48	Orviz Fernández P., David M., Duma D.C., Ronchieri E., Gomes J., Salomoni D.	2020	Software Quality Assurance in INDIGO-DataCloud Project: a Converging Evolution of Software Engineering Practices to Support European Research e-Infrastructures
S49	Wang Y., Mäntylä M.V., Demeyer S., Wiklund K., Eldh S., Kairi T.	2020	Software test automation maturity: A survey of the state of the practice
S50	E. Bernard; F. Ambert; B. Legeard	2020	Supporting efficient test automation using lightweight MBT
S51	R. Li; X. Liu; X. Zheng; C. Zhang; H. Liu	2020	TDD4Fog: A Test-Driven Software Development Platform for Fog Computing Systems
S52	Y. Wang; M. Pyhäjärvi; M. V. Mäntylä	2020	Test Automation Process Improvement in a DevOps Team: Experience Report
S53	Hasan M.M., Bhuiyan F.A., Rahman A.	2020	Testing practices for infrastructure as code
S54	Marlowe T.J., Kirova V., Chang G.	2020	The state of agile: Changes in the world of change
S55	Klemets J., Storholmen T.C.B.	2020	Towards Super User-Centred Continuous Delivery: A Case Study
S56	Ding Z., Chen J., Shang W.	2020	Towards the use of the readily available tests from the release pipeline as performance tests. Are we there yet

S57	Leotta M., Cerioli M., Olianas D., Ricca F.	2020	Two experiments for evaluating the impact of Hamcrest and AssertJ on assertion development
S58	K. Gallaba; S. McIntosh	2020	Use and Misuse of Continuous Integration Features: An Empirical Study of Projects That (Mis)Use Travis CI
S59	Y. Zhou; Y. Su; T. Chen; Z. Huang; H. C. Gall; S. Panichella	2020	User Review-Based Change File
S60	Yu L., Alégroth E., Chatzipetrou P., Gorschek T.	2020	Utilising CI environment for efficient and effective testing of NFRs
S61	Van Rossem S., Tavernier W., Colle D., Pickavet M., Demeester P.	2020	VNF Performance modelling: From stand-alone to chained topologies
S62	Bertolino, Antonia and Angelis, Guglielmo De and Guerriero, Antonio and Miranda, Breno and Pietrantuono, Roberto and Russo, Stefano	2019	DevOpRET: Continuous reliability testing in DevOps
S63	Jacobsen, Douglas M. and Kleinman, Randy and Longley, Harold	2019	Managing a Cray supercomputer as a git branch
S64	B. Meyers; K. Gadeyne; B. Oakes; M. Bernaerts; H. Vangheluwe; J. Denil	2019	A Model-Driven Engineering Framework to Support the Functional Safety Process
S65	F. Zampetti; G. Bavota; G. Canfora; M. D. Penta	2019	A Study on the Interplay between Pull Request Review and Continuous Integration Builds
S66	D. Chhillar; K. Sharma	2019	ACT Testbot and 4S Quality Metrics in XAAS Framework
S67	M. K. A. Abbass; R. I. E. Osman; A. M. H. Mohammed; M. W. A. Alshaikh	2019	Adopting Continuous Integeration and Continuous Delivery for Small Teams
S68	M. Guerriero; M. Garriga; D. A. Tamburri; F. Palomba	2019	Adoption, Support, and Challenges of Infrastructure-as-Code: Insights from Industry
S69	T. Durieux; R. Abreu; M. Monperrus; T. F. Bissyandé; L. Cruz	2019	An Analysis of 35+ Million Jobs of Travis CI
S70	T. Vasile; S. Cane; C. Bertram; F. Jakob	2019	Applying Security Concepts to Continuous Integration for the Purpose of Testing Embedded Systems
S71	C. Vassallo; S. Proksch; H. C. Gall; M. Di Penta	2019	Automated Reporting of Anti-Patterns and Decay in Continuous Integration
S72	A. Janes; B. Russo	2019	Automatic Performance Monitoring and Regression Testing During the Transition from Monolith to Microservices
S73	Krym T., Poniszewska-Marańda A., Markl E., Dupas R.	2019	Automatic Process of Continuous Integration of Web Application
S74	Najafi A., Rigby P.C., Shang W.	2019	Bisecting commits and modeling commit risk during testing
S75	D. A. Tomassi; N. Dmeiri; Y. Wang; A. Bhowmick; Y. Liu; P. T. Devanbu; B. Vasilescu; C. Rubio-González	2019	BugSwarm: Mining and Continuously Growing a Dataset of Reproducible Failures and Fixes
S76	Satyal S., Weber I., Paik H.-Y., Di Ciccio C., Mendling J.	2019	Business process improvement with the AB-BPM methodology
S77	R. K. Gupta; M. Venkatachalapathy; F. K. Jeberla	2019	Challenges in Adopting Continuous Delivery and DevOps in a Globally Distributed Product Team: A Case Study of a Healthcare Organization
S78	Judvaitis J., Nesenbergs K., Balass R., Greitans M.	2019	Challenges of DevOps ready IoT testbed
S79	Nogueira A.F., Sergeant E., Ribeiro J.C.B., Zenha-Rela M.A., Craske A.	2019	Collecting data from continuous practices: An infrastructure to support team development
S80	C. Singh; N. S. Gaba; M. Kaur; B. Kaur	2019	Comparison of Different CI/CD Tools Integrated with Cloud Platform
S81	I. M. A. Jawarneh; P. Bellavista; F. Bosi; L. Foschini; G. Martuscelli; R. Montanari; A. Palopoli	2019	Container Orchestration Engines: A Thorough Functional and Performance Comparison

S82	M. Grambow; F. Lehmann; D. Bermbach	2019	Continuous Benchmarking: Using System Benchmarking in Build Pipelines
S83	Glein R., Perloff A., Ulmer K.	2019	Continuous integration of FPGA designs for CMS
S84	W. Felidré; L. Furtado; D. A. da Costa; B. Cartaxo; G. Pinto	2019	Continuous Integration Theater
S85	Johanssen, JO; Kleebaum, A; Paech, B; Bruegge, B	2019	Continuous software engineering and its support by usage and decision knowledge: An interview study with practitioners
S86	L. G. Guşeilă; D. Bratu; S. Moraru	2019	Continuous Testing in the Development of IoT Applications
S87	Lescisin M., Mahmoud Q.H., Cioraca A.	2019	Design and implementation of SFCI: A tool for security focused continuous integration
S88	O. Veres; N. Kunanets; V. Pasichnyk; N. Veretennikova; R. Korz; A. Leheza	2019	Development and Operations - the Modern Paradigm of the Work of IT Project Teams
S89	R. A. K. Jennings; G. Gannod	2019	DevOps - Preparing Students for Professional Practice
S90	C. Heistand; J. Thomas; N. Tzeng; A. R. Badger; L. M. Rodriguez; A. Dalton; J. Pai; A. Bodzas; D. Thompson	2019	DevOps for Spacecraft Flight Software
S91	L. Georgeta Guşeilă; D. -V. Bratu; S. -A. Moraru	2019	DevOps Transformation for Multi-Cloud IoT Applications
S92	P. Agrawal; N. Rawat	2019	Devops, A New Approach To Cloud Development & Testing
S93	Embury S.M., Page C.	2019	Effect of continuous integration on build health in undergraduate team projects
S94	C. Vassallo	2019	Enabling Continuous Improvement of a Continuous Integration Process
S95	K. Baral; R. Mohod; J. Flamm; S. Goldrich; P. Ammann	2019	Evaluating a Test Automation Decision Support Tool
S96	H. Huijgens; E. Greuter; J. Brons; E. A. van Doorn; I. Papadopoulos; F.	2019	Factors Affecting Cloud Infra-Service Development Lead Times: A Case Study at ING
	Morales Martinez; M. Aniche; O. Visser; A. van Deursen		
S97	T. Suk; J. Hwang; M. F. Bulut; Z. Zeng	2019	Failure-Aware Application Placement Modeling and Optimization in High Turnover DevOps Environment
S98	Bezemer C.-P., Eismann S., Ferme V., Grohmann J., Heinrich R., Jamshidi P., Shang W., Van Hoorn A., Villavicencio M., Walter J., Willnecker F.	2019	How is performance addressed in DevOps? A survey on industrial practices
S99	B. Chen	2019	Improving the Software Logging Practices in DevOps
S100	S. Carturan; D. Goya	2019	Major Challenges of Systems-of-Systems with Cloud and DevOps ‚Äì A Financial Experience Report
S101	J. A. Shah; D. Dubaria	2019	NetDevOps: A New Era Towards Networking DevOps
S102	J. Haavisto; M. Arif; L. Lov√©n; T. Lepp√§nen; J. Riekki	2019	Open-source RANs in Practice: an Over-The-Air Deployment for 5G MEC
S103	Keahey K., Anderson J., Ruth P., Colleran J., Hammock C., Stubbs J., Zhen Z.	2019	Operational lessons from chameleon
S104	K. Hakimzadeh; J. Dowling	2019	Ops-Scale: Scalable and Elastic Cloud Operations by a Functional Abstraction and Feedback Loops
S105	E. Salinas	2019	Pat Helland on Failure and Resilience in Distributed Systems
S106	Al-Sabbagh K.W., Staron M., Hebig R., Meding W.	2019	Predicting test case verdicts using textual analysis of committed code churns
S107	A. Nuriddinov; W. Tavernier; D. Colle; M. Pickavet; M. Peustery; S. Schneidery	2019	Reproducible Functional Tests for Multi-scale Network Services
S108	Wiedemann A., Forsgren N., Wiesche M., Gewald H., Krcmar H.	2019	Research for practice: The Devops phenomenon

S109	P. K. Sidhu; G. Mussbacher; S. McIntosh	2019	Reuse (or Lack Thereof) in Travis CI Specifications: An Empirical Study of CI Phases and Commands
S110	Mäkinen S., Puonti M., Lehtonen T., Mikkonen T., Kilamo T., Männistö T.	2019	Revisiting continuous deployment maturity: A two-year perspective
S111	Siewruk G., Mazurczyk W., Karpiński A.	2019	Security assurance in Devops methodologies and related environments
S112	Vera-Rivera F.H., Vera-Rivera J.L., Gaona-Cuevas C.M.	2019	Sinplafut: A microservices - Based application for soccer training
S113	S. M. Naik; M. Fernandes; G. Pushpakumar; R. Pathak	2019	Smart Grid Communication Protocol Test Automation along with Protection Test Automation
S114	Risdianto A.C., Usman M., Kim J.W.	2019	SmartX box: Virtualized hyper-converged resources for building an affordable playground
S115	K. Czarnecki	2019	Software Engineering for Automated Vehicles: Addressing the Needs of Cars That Run on Software and Data
S116	Keskin Kaynak İ., Çilden E., Aydin S.	2019	Software Quality Improvement Practices in Continuous Integration
S117	Cunningham S., Gambo J., Lawless A., Moore D., Yilmaz M., Clarke P.M., O'Connor R.V.	2019	Software Testing: A Changing Career
S118	Rahman A., Williams L.	2019	Source code properties of defective infrastructure as code scripts
S119	Kapoor S., Sagar K., Reddy B.V.R.	2019	Speedroid: A novel automation testing tool for mobile apps
S120	Arulkumar V., Lathamanju R.	2019	Start to Finish Automation Achieve on Cloud with Build Channel: By DevOps Method
S121	Figalist I., Biesdorf A., Brand C., Feld S., Kiermeier M.	2019	Supporting the DevOps Feedback Loop using Unsupervised Machine Learning
S122	G. Lim; M. Ham; J. Moon; W. Song; S. Woo; S. Oh	2019	TAOS-CI: Lightweight Modular Continuous Integration System for Edge Computing
S123	Cruzes D.S., Melsnes K., Marczak S.	2019	Testing in a DevOps Era: Perceptions of Testers in Norwegian Organisations
S124	D. Martin; S. Panichella	2019	The Cloudification Perspectives of Search-Based Software Testing
S125	Fazayeli H., Syed-Mohamad S.M., Md Akhir N.S.	2019	Towards auto-labelling issue reports for pull-based software development using text mining approach
S126	Meixner K., Winkler D., Biffl S.	2019	Towards combined process & tool variability management in software testing
S127	R. Pietrantuono; A. Bertolino; G. De Angelis; B. Miranda; S. Russo	2019	Towards Continuous Software Reliability Testing in DevOps
S128	F. Giorgi; F. Paulisch	2019	Transition Towards Continuous Delivery in the Healthcare Domain
S129	C. Paule; T. F. Düllmann; A. Van Hoorn	2019	Vulnerabilities in Continuous Delivery Pipelines? A Case Study
S130	M. Chwalisz; K. Geissdoerfer; A. Wolisz	2019	Walker: DevOps Inspired Workflow for Experimentation
S131	B. Benni; M. Blay-Fornarino; S. Mosser; F. Précisio; G. Jungbluth	2019	When DevOps Meets Meta-Learning: A Portfolio to Rule them all
S132	Daoudagh, Said and Lonetti, Francesca and Marchetti, Eda	2019	An automated framework for continuous development and testing of access control systems
S133	Luz, Welder Pinheiro and Pinto, Gustavo and Bonifácio, Rodrigo	2018	Building a Collaborative Culture: A Grounded Theory of Well Succeeded Devops Adoption in Practice
S134	Osses, Felipe and Márquez, Gastón and Astudillo, Hernán	2018	Exploration of Academic and Industrial Evidence about Architectural Tactics and Patterns in Microservices
S135	Schulz, Henning and Angerstein, Tobias and van Hoorn, André	2018	Towards Automating Representative Load Testing in Continuous Software Engineering
S136	K. Kuusinen; V. Balakumar; S. C. Jepsen; S. H. Larsen; T. A. Lemqvist; A. Muric; A. Ø. Nielsen; O. Vestergaard	2018	A Large Agile Organization on Its Journey Towards DevOps

S137	Sandobalin J.	2018	A Model-Driven Approach to Continuous Delivery of Cloud Resources
S138	Sandobalin J., Insfran E., Abrahao S.	2018	A smart provisioning approach to cloud infrastructure
S139	Baudry B., Harrand N., Schulte E., Timperley C., Tan S.H., Selakovic M., Ugherughe E.	2018	A spoonful of DevOps helps the GI go down
S140	J. Shah; D. Dubaria; J. Widhalm	2018	A Survey of DevOps tools for Networking
S141	H. Li; T. Chen; A. E. Hassan; M. Nasser; P. Flora	2018	Adopting Autonomic Computing Capabilities in Existing Large-Scale Systems
S142	Zykov S.V.	2018	Agile services
S143	Akman S., Aksuyek E.B., Kaynak O.	2018	ALM Tool Infrastructure with a Focus on DevOps Culture
S144	Wiedemann A., Wiesche M.	2018	Are you ready for Devops? Required skill set for Devops teams
S145	I. Rubasinghe; D. Meedeniya; I. Perera	2018	Automated Inter-artefact Traceability Establishment for DevOps Practice
S146	R. V. Rosa; C. E. Rothenberg	2018	Automated VNF Testing with Gym: A Benchmarking Use Case
S147	V. Debroy; L. Brimble; M. Yost; A. Erry	2018	Automating Web Application Testing from the Ground Up: Experiences and Lessons Learned in an Industrial Setting
S148	M. J. Kargar; A. Hanifizade	2018	Automation of regression test in microservice architecture
S149	V. Mohan; L. ben Othmane; A. Kres	2018	BP: Security Concerns and Best Practices for Automation of Software Deployment Processes: An Industrial Case Study
S150	A. Rahman; L. Williams	2018	Characterizing Defective Configuration Scripts Used for Continuous Deployment
S151	Rahman A., Agrawal A., Krishna R., Sobran A.	2018	Characterizing the influence of continuous integration: Empirical results from 250+ open source and proprietary projects
S152	A. Agarwal; S. Gupta; T. Choudhury	2018	Continuous and Integrated Software Development using DevOps
S153	X. Bai; M. Li; D. Pei; S. Li; D. Ye	2018	Continuous Delivery of Personalized Assessment and Feedback in Agile Software Engineering Projects
S154	S. A. I. B. S. Arachchi; I. Perera	2018	Continuous Integration and Continuous Delivery Pipeline Automation for Agile Software Project Management
S155	L. Williams	2018	Continuously Integrating Security
S156	Alshahwan N., Gao X., Harman M., Jia Y., Mao K., Mols A., Tei T., Zorin I.	2018	Deploying search based software engineering with sapienz at facebook
S157	Marijan D., Sen S.	2018	Devops enhancement with continuous test optimization
S158	D. Marijan; M. Liaaen; S. Sen	2018	DevOps Improvements for Reduced Cycle Times with Integrated Test Optimizations for Continuous Integration
S159	Angara J., Gutta S., Prasad S.	2018	DevOps with continuous testing architecture and its metrics model
S160	Park S., Huh J.-H.	2018	Effect of cooperation on manufacturing IT project development an for successful industry 4.0 Project: Safety management for security
S161	Mårtensson T., Ståhl D., Bosch J.	2018	Enable more frequent integration of software in industry projects
S162	Casale G., Li C.	2018	Enhancing Big Data Application Design with the DICE Framework
S163	T. F. Düllmann; C. Paule; A. van Hoorn	2018	Exploiting DevOps Practices for Dependable and Secure Continuous Delivery Pipelines
S164	Loseva E., Obeid A., Richter H., Backes R., Eichhorn D.	2018	FIXIT - A semi-automatic software deployment tool for arbitrary targets
S165	Jiang H., Chen X., He T., Chen Z., Li X.	2018	Fuzzy clustering of crowdsourced test reports for apps
S166	D. Widder; B. Vasilescu; M. Hilton; C. Kästner	2018	I'm Leaving You, Travis: A Continuous Integration Breakup Story
S167	Fernandes, TCM; Costa, I; Salvetti, N; de Magalhaes, FLF; Fernandes, AA	2018	Influence of DevOps practices in IT management processes according to the COBIT 5 model

S168	Soenen T., van Rossem S., Tavernier W., Vicens F., Valocchi D., Trakadas P., Karkazis P., Xilouris G., Eardley P., Kolometsos S., Kourtis M.-A., Guija D., Siddiqui S., Hasselmeyer P., Bonnet J., Lopez D.	2018	Insights from SONATA: Implementing and integrating a microservice-based NFV service platform with a DevOps methodology
S169	S. van Rossem; W. Tavernier; D. Colle; M. Pickavet; P. Demeester	2018	Introducing Development Features for Virtualized Network Services
S170	Asha N., Mani P.	2018	Knowledge-based acceptance test driven agile approach for quality software development
S171	F. L. Eickhoff; M. L. McGrath; C. Mayer; A. Bieswanger; P. A. Wojciak	2018	Large-scale application of IBM Design Thinking and Agile development for IBM z14
S172	Staron M., Meding W., Söder O., Bäck M.	2018	Measurement and Impact Factors of Speed of Reviews and Integration in Continuous Software Engineering
S173	L. Chen	2018	Microservices: Architecting for Continuous Delivery and DevOps
S174	H. Alipour; Y. Liu	2018	Model Driven Deployment of Auto-Scaling Services on Multiple Clouds
S175	M. Wurster; U. Breitenbücher; O. Kopp; F. Leymann	2018	Modeling and Automated Execution of Application Deployment Tests
S176	D'Ambrogio A., Falcone A., Garro A., Giglio A.	2018	On the importance of simulation in enabling continuous delivery and evaluating deployment pipeline performance
S177	Zhang Y., Vasilescu B., Wang H., Filkov V.	2018	One size does not fit all: An empirical study of containerized continuous deployment workflows
S178	A. Cheriyan; R. R. Gondkar; T. Gopal; S. B. S.	2018	Quality Assurance Practices in Continuous Delivery - an implementation in Big Data Domain
S179	R. Mijumbi; K. Okumoto; A. Asthana; J. Meekel	2018	Recent Advances in Software Reliability Assurance
S180	Kerzazi N., EL Asri I.	2018	Release engineering: From structural to functional view
S181	G. Marquez; F. Osses; H. Astudillo	2018	Review of Architectural Patterns and Tactics for Microservices in Academic and Industrial Literature
S182	Satyal S., Weber I., Paik H.-Y., Di Ciccio C., Mendling J.	2018	Shadow Testing for Business Process Improvement
S183	Limoncelli T.A.	2018	SQL is no excuse to avoid DevOps
S184	P. Zimmerer	2018	Strategy for Continuous Testing in iDevOps
S185	K. K. Luhana; C. Schindler; W. Slany	2018	Streamlining mobile app deployment with Jenkins and Fastlane in the case of Catrobat's pocket code
S186	X. Bai; D. Pei; M. Li; S. Li	2018	The DevOps Lab Platform for Managing Diversified Projects in Educating Agile Software Engineering
S187	Guamán D., Pérez J., Díaz J.	2018	Towards a (semi)-automatic reference process to support the reverse engineering and reconstruction of software architectures
S188	K. Martin; U. Ömer; M. Florian	2018	Towards a Continuous Feedback Loop for Service-Oriented Environments
S189	Steffens A., Lichter H., Moscher M.	2018	Towards data-driven continuous compliance testing
S190	F. Klinaku; V. Ferme	2018	Towards Generating Elastic Microservices: A Declarative Specification for Consistent Elasticity Configurations
S191	M. Peuster; H. Karl	2018	Understand Your Chains and Keep Your Deadlines: Introducing Timeconstrained Profiling for NFV
S192	Kim C., Kim S., Kim J.	2018	Understanding automated continuous integration for containerized smart energy IoT-cloud service
S193	B. Snyder; B. Curtis	2018	Using Analytics to Guide Improvement during an Agile,ÄìDevOps Transformation
S194	Schermann G., Cito J., Leitner P., Zdun U., Gall H.C.	2018	We're doing it live: A multi-method empirical study on continuous experimentation
S195	Pinto G., Castor F., Bonifacio R., Rebouças M.	2018	Work practices and challenges in continuous integration: A survey with Travis CI users

S196	Fabian Fagerholm and Alejandro {Sanchez Guinea} and Hanna Mäenpää and Jürgen Münch	2017	The RIGHT model for Continuous Experimentation
S197	Ferme, Vincenzo and Pautasso, Cesare	2017	Towards Holistic Continuous Software Performance Assessment
S198	B. P. Eddy; N. Wilde; N. A. Cooper; B. Mishra; V. S. Gamboa; K. M. Shah; A. M. Deleon; N. A. Shields	2017	A Pilot Study on Introducing Continuous Integration and Delivery into Undergraduate Software Engineering Courses
S199	C. Vassallo; G. Schermann; F. Zampetti; D. Romano; P. Leitner; A. Zaidman; M. Di Penta; S. Panichella	2017	A Tale of CI Build Failures: An Open Source and a Financial Organization Perspective
S200	A. J. Younge; K. Pedretti; R. E. Grant; R. Brightwell	2017	A Tale of Two Systems: Using Containers to Deploy HPC Applications on Supercomputers and Clouds
S201	S. Wongkampoo; S. Kiattisin	2017	Atom-Task Precondition Technique to Optimize Large Scale GUI Testing Time based on Parallel Scheduling Algorithm
S202	Wu C.-F.E., Burugula R.S., Yu H., Dubey N., Jann J., Nguyen M.	2017	Automation of cloud node installation for testing and scalable provisioning
S203	M. Shahin; M. A. Babar; M. Zahedi; L. Zhu	2017	Beyond Continuous Delivery: An Empirical Investigation of Continuous Deployment Challenges
S204	T. T. Brooks	2017	Big Data Complex Event Processing for Internet of Things Provenance: Benefits for Audit, Forensics, and Safety
S205	Ståhl D., Bosch J.	2017	Cinders: The continuous integration and delivery architecture framework
S206	Wettinger J., Breitenbücher U., Falkenthal M., Leymann F.	2017	Collaborative gathering and continuous delivery of DevOps solutions through repositories
S207	C. H. Kao	2017	Continuous evaluation for application development on cloud computing environments
S208	D. Stahl; T. Martensson; J. Bosch	2017	Continuous practices and devops: beyond the buzz, what does it all mean?
S209	Fitzgerald B., Stol K.-J.	2017	Continuous software engineering: A roadmap and agenda
S210	Metzger S., Durden D., Sturtevant C., Luo H., Pingintha-Durden N., Sachs T., Serafimovich A., Hartmann J., Li J., Xu K., Desai A.R.	2017	Eddy4R 0.2.0: A DevOps model for community-extensible processing and analysis of eddy-covariance data based on R, Git, Docker, and HDF5
S211	J. A. Kupsch; B. P. Miller; V. Basupalli; J. Burger	2017	From continuous integration to continuous assurance
S212	P. Perera; R. Silva; I. Perera	2017	Improve software quality through practicing DevOps
S213	S. Vost; S. Wagner	2017	Keeping Continuous Deliveries Safe
S214	Zimmermann, O	2017	Microservices tenets: Agile approach to service development and deployment
S215	Chung S.	2017	Object-oriented programming with DevOps
S216	Heinrich R., Van Hoorn A., Knoche H., Li F., Lwakatare L.E., Pahl C., Schulte S., Wettinger J.	2017	Performance engineering for microservices: Research challenges & directions
S217	Haili W., Renbin G., Congbin W., Lei G.	2017	Research and application of development model of information service for IOT of oil and gas production based on cloud architecture
S218	Z. Farahmandpour; S. Versteeg; J. Han; A. Kameswaran	2017	Service Virtualisation of Internet-of-Things Devices: Techniques and Challenges
S219	Bucena I., Kirikova M.	2017	Simplifying the devops adoption process
S220	A. van Deursen	2017	Software engineering without borders
S221	D. Spinellis	2017	State-of-the-Art Software Testing
S222	Martensson T., Stahl D., Bosch J.	2017	The EMFIS model - Enable more frequent integration of software
S223	Y. Zhao; A. Serebrenik; Y. Zhou; V. Filkov; B. Vasilescu	2017	The impact of continuous integration on other software development practices: A large-scale empirical study

S224	C. Parnin; E. Helms; C. Atlee; H. Boughton; M. Ghattas; A. Glover; J. Holman; J. Micco; B. Murphy; T. Savor; M. Stumm; S. Whitaker; L. Williams	2017	The Top 10 Adages in Continuous Deployment
S225	S. Palihawadana; C. H. Wijeweera; M. G. T. N. Sanjitha; V. K. Liyanage; I. Perera; D. A. Meedeniya	2017	Tool support for traceability management of software artefacts with DevOps practices
S226	E. Laukkanen; M. Paasivaara; J. Itkonen; C. Lassenius; T. Arvonen	2017	Towards Continuous Delivery by Reducing the Feature Freeze Period: A Case Study
S227	D. Ameller; C. Farr√©; X. Franch; D. Valerio; A. Cassarino	2017	Towards continuous software release planning
S228	C. Duffau; B. Grabiec; M. BlayFornarino	2017	Towards Embedded System Agile Development Challenging Verification, Validation and Accreditation: Application in a Healthcare Company
S229	Nidagundi P., Novickis L.	2017	Towards utilization of lean canvas in the devops software
S230	Hilton M., Nelson N., Tunnell T., Marinov D., Dig D.	2017	Trade-offs in continuous integration: Assurance, security, and flexibility
S231	Morris D., Voutsinas S., Hambly N.C., Mann R.G.	2017	Use of Docker for deployment and testing of astronomy software
S232	M. Zhao; F. Le Gall; P. Cousin; R. Vilalta; R. Mu√±oz; S. Castro; M. Peuster; S. Schneider; M. Siapera; E. Kapassa; D. Kyriazis; P. Hasselmeyer; G. Xilouris; C. Tranoris; S. Denazis; J. Martrat	2017	Verification and validation framework for 5G network services and apps
S233	Ur Rahman, Akond Ashfaque and Williams, Laurie	2016	Security Practices in DevOps
S234	F. Calefato; F. Lanubile	2016	A Hub-and-Spoke Model for Tool Integration in Distributed Development
S235	Di Nitto E., Jamshidi P., Guerriero M., Spais I., Tamburri D.A.	2016	A software architecture framework for quality-aware devops
S236	Hanappi O., Hummer W., Dustdar S.	2016	Asserting reliable convergence for configuration management scripts
S237	J. Bae; C. Kim; J. Kim	2016	Automated deployment of SmartX IoT-cloud services based on continuous integration
S238	Makki M., Van D., Joosen L.W.	2016	Automated workflow regression testing for multi-tenant SaaS: Integrated support in self-service configuration dashboard
S239	Schermann G., Schöni D., Leitner P., Gall H.C.	2016	Bifrost: Supporting continuous deployment with automated enactment of multi-phase live testing strategies
S240	Risdianto A.C., Shin J., Kim J.	2016	Building and operating distributed SDN-cloud testbed with hyper-convergent smartx boxes
S241	D. Liu; H. Zhu; C. Xu; I. Bayley; D. Lightfoot; M. Green; P. Marshall	2016	CIDE: An Integrated Development Environment for Microservices
S242	C. Vassallo; F. Zampetti; D. Romano; M. Beller; A. Panichella; M. Di Penta; A. Zaidman	2016	Continuous Delivery Practices in a Large Financial Organization
S243	T. Savor; M. Douglas; M. Gentili; L. Williams; K. Beck; M. Stumm	2016	Continuous Deployment at Facebook and OANDA
S244	Rossi C., Shibley E., Su S., Beck K., Savor T., Stumm M.	2016	Continuous deployment of mobile software at facebook (showcase)
S245	C. Pang; A. Hindle	2016	Continuous Maintenance
S246	M. Staples; L. Zhu; J. Grundy	2016	Continuous Validation for Data Analytics Systems
S247	Hadar E., Hadar I.	2016	CURA: Complex-system Unified reference architecture position paper: A practitioner view
S248	Riungu-Kalliosaari L., Mäkinen S., Lwakatare L.E., Tiihonen J., Männistö T.	2016	DevOps adoption benefits and challenges in practice: A case study

S249	Colavita F.	2016	Devops movement of enterprise agile breakdown silos, create collaboration, increase quality, and application speed
S250	M. Callanan; A. Spillane	2016	DevOps: Making It Easy to Do the Right Thing
S251	Sheridan C., Whigham D., Artac M.	2016	DICE fault injection tool
S252	Amith Raj MP; A. Kumar; S. J. Pai; A. Gopal	2016	Enhancing security of Docker using Linux hardening techniques
S253	M. T. Rahman; L. Querel; P. C. Rigby; B. Adams	2016	Feature Toggles: Practitioner Practices and a Case Study
S254	Mäkinen S., Leppänen M., Kilamo T., Mattila A.-L., Laukkanen E., Pagels M., Männistö T.	2016	Improving the delivery cycle: A multiple-case study of the toolchains in Finnish software intensive enterprises
S255	Jones S., Noppen J., Lettice F.	2016	Management challenges for devops adoption within UK SMEs
S256	Artač M., Borovšak T., Di Nitto E., Guerriero M., Tamburri D.A.	2016	Model-Driven continuous deployment for quality devops
S257	B. Adams; S. McIntosh	2016	Modern Release Engineering in a Nutshell -- Why Researchers Should Care
S258	Kroß J., Willnecker F., Zwickl T., Krcmar H.	2016	PET: Continuous performance evaluation tool
S259	Ohtsuki M., Ohta K., Kakeshita T.	2016	Software engineer education support system ALECSS utilizing devOps tools
S260	A. A. U. Rahman; L. Williams	2016	Software Security in DevOps: Synthesizing Practitioners,Äô Perceptions and Practices
S261	Cito J., Mazlami G., Leitner P.	2016	TemPerf: Temporal correlation between performance metrics and source code
S262	Shahin M., Babar M.A., Zhu L.	2016	The Intersection of Continuous Deployment and Architecting Process: Practitioners' Perspectives
S263	R. Punjabi; R. Bajaj	2016	User stories to user reality: A DevOps approach for the cloud
S264	Gottesheim, Wolfgang	2015	Challenges, Benefits and Best Practices of Performance Focused DevOps
S265	Shtern, Mark and Simmons, Bradley and Smit, Michael and Lu, Hongbin and Litoiu, Marin	2015	Performance Management and Monitoring
S266	Stillwell M., Coutinho J.G.F.	2015	A DevOps approach to integration of software components in an EU research project
S267	E. Salant; P. Leitner; K. Wallbom; J. Ahtes	2015	A framework for a cost-efficient cloud ecosystem
S268	D. Bruneo; F. Longo; G. Merlino; N. Peditto; C. Romeo; F. Verboso; A. Puliafito	2015	A Modular Approach to Collaborative Development in an OpenStack Testbed
S269	Rajagopalan S., Jamjoom H.	2015	App–Bisect: Autonomous healing for microservice-based apps
S270	H. Chen; R. Kazman; S. Haziyev; V. Kropov; D. Chtchourov	2015	Architectural Support for DevOps in a Neo-Metropolis BDaaS Platform
S271	Scheuner J., Cito J., Leitner P., Gall H.	2015	Cloud workBench: Benchmarking IaaS providers based on infrastructure-ascode
S272	S. Gebert; C. Schwartz; T. Zinner; P. Tran-Gia	2015	Continuously delivering your network
S273	Lehtonen T., Suonsyrjä S., Kilamo T., Mikkonen T.	2015	Defining metrics for continuous delivery and deployment pipeline
S274	D. Bruneo; F. Longo; G. Merlino; N. Peditto; C. Romeo; F. Verboso; A. Puliafito	2015	Enabling Collaborative Development in an OpenStack Testbed: The CloudWave Use Case
S275	Wettinger J., Andrikopoulos V., Leymann F.	2015	Enabling devops collaboration and continuous delivery using diverse application environments
S276	M. Soni	2015	End to End Automation on Cloud with Build Pipeline: The Case for DevOps in Insurance Industry, Continuous Integration, Continuous Testing, and Continuous Delivery
S277	Segall I., Tzoref-Brill R.	2015	Feedback-driven combinatorial test design and execution
S278	Vasilescu B., Yu Y., Wang H., Devanbu P., Filkov V.	2015	Quality and productivity outcomes relating to continuous integration in GitHub

S279	M. de Bayser; L. G. Azevedo; R. Cerqueira	2015	ResearchOps: The case for DevOps in scientific applications
S280	E. Laukkanen; M. Paasivaara; T. Arvonen	2015	Stakeholder Perceptions of the Adoption of Continuous Integration-A Case Study
S281	A. A. U. Rahman; E. Helms; L. Williams; C. Parnin	2015	Synthesizing Continuous Deployment Practices Used in Software Development
S282	N. Rathod; A. Surve	2015	Test orchestration a framework for Continuous Integration and Continuous deployment
S283	A. Wahaballa; O. Wahballa; M. Abdellatief; H. Xiong; Z. Qin	2015	Toward unified DevOps model
S284	M. Virmani	2015	Understanding DevOps & bridging the gap from continuous integration to continuous delivery
S285	Chen J., Xu X., Osterweil L.J., Zhu L., Brun Y., Bass L., Xiao J., Li M., Wang Q.	2015	Using simulation to evaluate error detection strategies: A case study of cloudbased deployment processes
S286	J. Engblom	2015	Virtual to the (near) end: Using virtual platforms for continuous integration
S287	B. S. Farroha; D. L. Farroha	2014	A Framework for Managing Mission Needs, Compliance, and Trust in the DevOps Environment
S288	S. Harrer; C. R√∂ck; G. Wirtz	2014	Automated and Isolated Tests for Complex Middleware Products: The Case of BPEL Engines
S289	Fitzgerald B., Stol K.	2014	Continuous software engineering and beyond: Trends and challenges
S290	C. A. Cois; J. Yankel; A. Connell	2014	Modern DevOps: Optimizing software development through effective system interactions
S291	S. A. Wright; D. Druta	2014	Open source and standards: The role of open source in the dialogue between research and standardization
S292	S. W. Hussaini	2014	Strengthening harmonization of Development (Dev) and Operations (Ops) silos in IT environment through systems approach
S293	S. Bellomo; N. Ernst; R. Nord; R. Kazman	2014	Toward Design Decisions to Enable Deployability: Empirical Study of Three Projects Reaching for the Continuous Delivery Holy Grail
S294	Erculiani F., Abeni L., Palopoli L.	2014	UBuild: Automated testing and performance evaluation of embedded linux systems
S295	S. Neely; S. Stolt	2013	Continuous Delivery? Easy! Just Change Everything (Well, Maybe It Is Not That Easy)
S296	Schaefer A., Reichenbach M., Fey D.	2013	Continuous integration and automation for DevOps
S297	D. G. Feitelson; E. Frachtenberg; K. L. Beck	2013	Development and Deployment at Facebook
S298	S. Meyer; P. Healy; T. Lynn; J. Morrison	2013	Quality Assurance for Open Source Software Configuration Management

Information about authors / Информация об авторах

Brian PANDO, Computer and Systems Engineer. Research interests: Software Engineering, Web Development, Software Development.

Брайан ПАНДО, компьютерный и системный инженер. Научные интересы: программная инженерия, веб-разработка, разработка программного обеспечения.

Abraham DÁVILA is a Principal Professor of the Computer Engineering program and is a Doctoral Candidate in Software Engineering, in the field of process improvement. Field of scientific interests: Software engineering, Software quality process, Software quality product, Education in software engineering, Innovations based on software.

Авраам ДАВИЛА – профессор программы компьютерной инженерии и докторант в области программной инженерии. Область научных интересов: программная инженерия, процесс качества программного обеспечения, образование в области программной инженерии, инновации на основе программного обеспечения.