How to cite this article:

Ya'u, B. I., Salleh, N., Nordin, A., Alwan A. A., Idris, N. B., & Abas, H. (2019). A systematic mapping study on cloud-based mobile application testing. *Journal of Information and Communication Technology, 18*(4), 485-527.

A SYSTEMATIC MAPPING STUDY ON CLOUD-BASED MOBILE APPLICATION TESTING

Badamasi Imam Ya'u, Norsaremah Salleh, Azlin Nordin,
Ali Amer Alwan, Norbik Bashah Idris & Hafiza Abas
Department of Computer Science,
International Islamic University, Malaysia.

badamasi.imam@live.iium.edu.my, norsaremah@iium.edu.my; azlinodin@iium.edu.my; aliamer@iium.edu.my; norbik@iium.edu.my; hafiza.kl@utm.my

ABSTRACT

Mobile applications and devices have played a significant role in boosting global businesses that encompass various domains such as health, education, banking, and transportation. These tools have become indispensable for everyday activities, and its applications have been developing rapidly with diverse features and platforms. However, this has created new problems and security challenges. To ensure the quality and security of these applications, a rigorous and systematic testing using cloud-based environment is required. By employing systematic mapping study (SMS) method, this paper will examine the empirical studies that address the issues on cloud-based mobile application testing. This paper presents a total of 23 primary studies that investigate cloud based mobile application testing and the effect of Testing as a Service (TaaS). The majority of these studies (56.5%) contribute to literature with a number of framework proposals. A large proportion of the studies (60.9%) analyzed Android applications, and usually supported a single type of mobile app testing. Other than that, the majority of the studies (52.2%) have failed to investigate the outcomes of TaaS, despite a plethora of services that offers TaaS. The SMS method conducted in this paper has identified gaps in literature, which are: 1) there is a lack of general and scalable approaches to support the diverse types of mobile app testing for applications using various

Received: 30 September 2018 **Accepted**: 7 April 2019 **Published:** 26 September 2019

platforms, and 2) the lack of evaluation methods such as case study to validate the proposed approaches.

Keywords: Cloud-based mobile application testing, systematic mapping, testing-as-a-service.

INTRODUCTION

Mobile devices have become popular and ubiquitous in everyday life, and their popularity have supported a global market apps ecosystem (Holl, Vieira, & Faria, 2016; Liang et al., 2014; Tao, Lin, & Lu, 2015). Mobile applications are being developed at an exponential pace, creating many applications for various domains such as social services, health, education, banking, and transportation (Meng, Jiang, & Xu, 2015; Villanes, Costa, & Dias-Neto, 2015; Zein, Salleh, & Grundy, 2016; Zhang, Gao, Cheng, & Uehara, 2015). In the current decade, the mobile apps market has created a new paradigm shift as software products are delivered to consumers quickly and at a nominal cost. This essentially expands business opportunities for small entrepreneurs to compete with prominent software developers (Mahmood, Esfahani, Kacem, Mirzaei, Malek, & Stavrou; 2012; Mahmood, Mirzaei & Malek, 2014). This has led to a massive development of new apps, mostly downloaded on platforms such as Android and iOS which have created new sets of difficulties and security challenges (Huang & Gong, 2012; Mahmood et al., 2012; Meng et al., 2015; Tao et al., 2015; Villanes et al., 2015). One of the key features of mobile application is regular updates (Al-Ahmad, Aljunid, & Sani, 2013), but the timing to update a mobile platform in a new device model has been irregular, which compromises the quality of the mobile application (Rojas, Meireles, & Dias-Neto, 2016). Therefore, mobile applications would require frequent testing activities (Baride & Dutta, 2011; Haller, 2013; Starov, Vilkomir, & Kharchenko, 2013; Zein et al., 2016; Zhang & Pi, 2015), as the lack of proper testing and validation exacerbates the security challenges by making mobile apps vulnerable to both internal and external attacks (Zhong & Xiao, 2014).

Although testing is necessary and crucial in any application development (Choudhary, Gorla, & Orso, 2015; Holzmann & Hutflesz, 2014), testing mobile apps is expensive due to the short time period to market the product (Starov et al., 2013), challenging (Chana & Rana, 2012; Haller, 2013; Villanes et al., 2015) and complicated due to the diverse underlining technology and methodology (Al-Ahmad et al., 2013; Baride & Dutta, 2011; Huang & Gong, 2012; Murugesan & Balasubramanian, 2014; Pardeshi, 2013; Tung, Lin, &

Shan, 2014; Zein et al., 2016). The large variations of mobile devices and their associated platforms and features have increased the complexities of mobile application, as well as the testing processes of these mobile apps (Holl et al., 2016; Rojas et al., 2016). Mobile app developers particularly those working on apps for the Android platform, which is the most widely used platform for developing mobile applications, would need to conduct compatibility testing across multiple devices prior to the release of a new application (Haller, 2013; Huang & Gong, 2012; Kuo, Liu, & Yu, 2015; Liu, Chen, & Chen, 2015).

In comparison with other software or system testing, mobile apps are developed with peculiarities which are distinguishable from traditional application systems (Kirubakaran & Karthikeyani, 2013; Muccini, Di Francesco, & Esposito, 2012). For instance, Speedy Writer system testing examines installation, performance, and printer compatibility, while DataRocket conducts system testing that examines the performance and network compatibility (Black, 2002).

Thus, to ensure an extensive array of mobile apps testing that deals with various problems, it is necessary to explore the testing strategies in cloud computing. Cloud computing is a well-consolidated paradigm that delivers convenient and on-demand services such as networks, servers, storage and applications (Chana & Rana, 2012; Coutinho, de Carvalho Sousa, Rego, Gomes, & de Souza, 2015); which offers a pay-as-you-go models to end users (Bai, Li, Chen, Tsai, & Gao, 2011). Therefore, it focuses on the management of these shared services (Rahimi, Ren, Liu, Vasilakos, & Venkatasubramanian, 2014). Cloud computing has been promising in the realm of computing, which indirectly impacts several in-demand research fields such as software testing (Baride & Dutta, 2011; Inçki, Ari, & Sözer, 2012).

Cloud-based testing, also referred to as Testing as a Service (TaaS), is a new service in cloud computing which is considered a popular scalable testing environment (Villanes et al., 2015). TaaS offers on-demand testing services and utility-based service models through cost effective ways (Al-Ahmad et al., 2013; Chana & Rana, 2012; Gao et al., 2012). TaaS utilizes cloud infrastructure based on 'pay-as-you-test' business model to improve resource sharing at an affordable cost (Malini, Venkatesh, Sundarakantham, & Mercyshalinie, 2014; Rojas et al., 2016; Villanes et al., 2015).

The key contribution of this paper is the presentation of a systematic mapping study (SMS) on cloud based mobile application testing. This paper is the first to employ the SMS method in literature that reports the current and future developments of cloud based mobile application testing. This paper explores the results on the dimensions of existing approaches to enable a clear understanding of the current contributions within the scope of the research, and a discussion on the weaknesses and areas for future research. Researchers and practitioners in the areas of mobile cloud computing for software and application testing will find the information in this SMS relevant for further study.

The remainder of the paper discusses the following main sections; The first section presents related work that discusses literature review on the scope of the study. The second section presents the research methodology that describes how the review was conducted. The third section details the data extraction that outlines the processes of data extraction. The fourth section presents the result and analysis reports on the finding from this SMS. The fifth section presents a discussion on the details of the different aspects of the SMS. In the final section, a conclusion to summarizes the paper is presented.

RELATED WORK

A survey was conducted to investigate and report on mobile cloud technology infrastructure, various service frameworks, and existing service models (Mohiuddin, Islam, Alam, & Ali, 2012). The study suggested that the mobile cloud access is a potential service format that expands the new business service domain. Bhattacharya and De (2016) presented a survey on the adaptation techniques used for computation offloading. Response time in cloud based mobile applications were investigated in a study by (Dey, Liu, Wang, & Lu, 2013).

Holl et al. (2016) investigated the methods that improved the effectiveness of testing processes in mobile app development. An automated software testing as a cloud-based service on TaaS was reported in Candea, Bucur and Zamfir (2010). In a survey conducted by Choudhary et al. (2015), the authors investigated and compared the existing test input generation tools for Android. A survey of cloud computing and testing tools for cloud-based software systems were reported in (Bai et al., 2011). Another study presented the review on mobile testing, cloud testing and mobile cloud application testing (Al-Ahmad et al., 2013). In Rahimi et al. (2014), the study presented the results of a survey on the current state of mobile cloud computing (MCC) with revised solutions. An overview on the scope and parameters of testing mobile application was presented in Muccini et al. (2012). Other than that, a survey on the elasticity management in the TaaS service model was reported in (Munoz-Escoı & Bernabeu-Aubán, 2015).

Despite the vast number of surveys and review papers that are available in literature, there are insufficient studies and systematic surveys that are conducted in areas of mobile application testing, especially studies that are based on systematic literature review (SLR) or SMS techniques. Zein et al. (2016) conducted a SMS study on mobile application testing techniques and their challenges. The study analyzed five mobile application testings: usability testing, test automation, context-awareness, security testing and general testing category. However, the study disregarded cloud-based mobile application testing or testing as a service, and was not emphasized nor highlighted in future research area.

In another SMS study by Silva et al. (2016), the findings about benchmark applications used in MCC research was presented. The authors highlighted that there are few studies on MCC, and in turn utilized real applications in controlled experiments to evaluate their proposal on MCC. Furthermore, few studies have provided substantial evidence on the application details to enable other researchers to replicate and enhance the experiments. In a survey by Coutinho et al. (2015), the authors performed a systematic review on cloud computing elasticity, addressing the different aspects of elasticity such as definition, metrics, tool support, evaluation and identification of existing solutions. Despite reporting current issues on cloud computing elasticity, the issue of mobile testing or testing as a service using cloud computing was not carried out by the authors.

Another SLR study by Chana and Rana (2012), which involved a detail analysis using SMS, suggested an empirical evaluation of cloud based testing techniques. The authors classified their work into four categories which are; Cloud based Testing, Automated Test Case generation, Testing Frameworks and Cloud Application Development. However, the SLR study did not include cloud-based testing techniques in the context of mobile apps. The study by Endo et al. (2016) reported SLR on high availability (HA) solutions for cloud computing and its challenges. The authors concluded that the complexity of cloud infrastructure has become a challenging issue to cloud providers and hence, suggested the need for the implementation of Testing-as-a-Service. Unfortunately, this SLR study did not investigate the issue of mobile app testing in the cloud.

RESEARCH METHODOLOGY

This section presents a review on the methodology used in this SMS. Systematic Mapping Study (SMS) is a reviewing method that provides a systematic structure of the research evidence following a specific classification scheme (Petersen, Feldt, Mujtaba, & Mattsson, 2008). The essential processes of the SMS includes; the definition of research questions, conduction of search, screening of papers, keywording of abstracts, data extraction, and mapping of the study (see Figure 1). The mapping study presented in this paper are referred to the guidelines presented in Petersen, Vakkalanka, and Kuzniarz (2015).

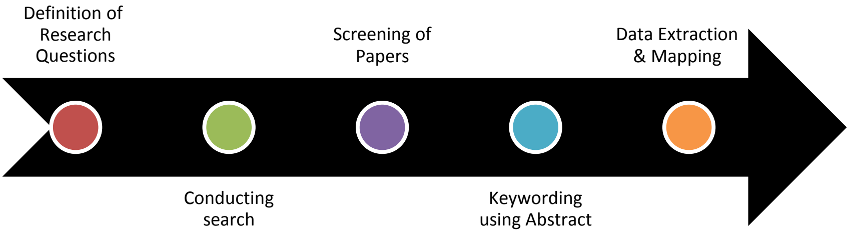
Definition of Research Questions
Conducting search
Screening of Papers
Keywording using Abstract
Data Extraction & Mapping

Figure 1. Systematic mapping process.

Definitions of Research Questions

The main objective of this study is to investigate, identify and classify all related evidence extracted from primary studies reported on cloud-based mobile application testing and Testing as a Service (TaaS). Therefore, the aim of this SMS is to answer the following questions:

RQ1: What are the types of cloud-based mobile app testing that are proposed in the studies?
RQ2: What research approaches do these studies apply, and what contribution facet(s) do they provide (e.g. framework, method, tool)?
RQ3: How mobile app TaaS affects the process of mobile app testing?

Sources of Evidence

Our pilot and primary search process involves the use of six online databases commonly used in published systematic reviews and mapping studies (e.g. (Salleh, Mendes, & Grundy, 2011; Silva et al., 2016; Zein et al., 2016)) which are; IEEEXplore, Scopus, ScienceDirect, ACM Digital Library, Springer and Web of Science.

Identifying the Relevant Literature

In this SMS, the search strategy is divided into three stages: pilot, primary and the snowballing search. The pilot search entails the use of different search strings on online databases to determine the search string that would yield the best results. The primary search focuses on searching relevant literature from online resource database, by using the search string identified from the pilot search. Finally, the snowballing search is carried out (one iterative searching from reference list of selected studies) to complement the search process.

Pilot Search Strategy

The aim of the pilot search is to find the most appropriate search strings that can return the best results from the list of selected online databases. This strategy is implemented to avoid the large number of false positives from the studies conducted in various disciplines. Furthermore, the pilot search overcomes the limitation of some search engines in accommodating relatively long search strings. During the pilot search, the six online databases mentioned earlier were used. Due to the limitations of some search engines such as Scopus and IEEE to input lengthy search strings, five different search strings were applied to each database (see Table 1).

Table 1

Search strings used during pilot search

String ID	String
S1	(mobile applications OR mobile apps) AND (type of testing OR testing strategy OR testing) AND (cloud OR cloud computing OR service)
S2	(mobile applications OR mobile apps) AND (cloud based OR cloud computing OR service) AND (testing OR strategy)
S3	(mobile apps OR mobile applications) AND (testing OR strategy)
S4	(mobile applications OR mobile apps) AND (cloud OR cloud based) AND (testing OR strategy)
S5	S5. (mobile applications OR mobile apps) AND (cloud OR cloud based) AND (testing as a service OR testing OR strategy)

Primary Search Strategy

By implementing the pilot search using different search strings, the most efficient search string that yielded the best result was selected. In reference to the study done by (Keele, 2007), the search string consists of synonyms and alternative keywords based on the research question. The search terms were combined to retrieve more relevant studies using 'OR' and 'AND' Boolean operators. Based on the results of the pilot search, the S5 search string was selected as it contained the common search terms, and was subsequently applied to all selected databases for searching the relevant primary studies.

Snowballing Search

During the snowballing search, another round of search is performed, whereby the reference lists of each selected paper from the primary search are thoroughly checked (one iteration of backward snowballing). Studies obtained from the backward snowballing search went through the same screening process based on the predefined inclusion and exclusion criteria. The studies that fulfilled the selection criteria were accepted for inclusion.

Inclusion and Exclusion Criteria

The selection of studies require a clear definition of the inclusion and exclusion criteria. Inclusion criterion is aimed at selecting all papers that report evidences of cloud-based mobile application testing or Testing as a Service (TaaS). Therefore, the selected primary studies are based on the comparison of titles and keywords, abstracts, and the full text (i.e. introduction, methodology, results and conclusion). Studies that fall in any of the following categories are excluded:

1. Studies that do not focus on cloud-based mobile application testing.
2. Short conference papers (less than four pages), posters, lecture notes, proposals, abstracts, opinion papers and work in progress reports.
3. Review papers and studies such as surveys, SLR, and SMS related to the topic.
4. Papers that were not written in English language.

Keywording of Abstracts

Keywording of abstract is a process that helps reduce the amount of time required to build a complete classification scheme for existing studies (Petersen et al., 2015). Hence, a thematic analysis approach to identify, analyze, and report themes that underlie the studies was adopted (Braun & Clarke, 2006). Keywording of abstracts are comprised of two phases. In the first phase, the abstracts of the selected studies were read, whereby the concepts, main contributions and area of focus that reflected the research questions were identified. The second phase involves the combination of a set of keywords that are gathered from different studies based on the thematic analysis approach, to gain insights on the nature and contribution of the research.

Due to the diverse mobile applications platforms, tools and approaches, a large number of concepts and techniques were discovered in the selected studies that addressed cloud based mobile application testing in the first phase of keywording. These concepts included black box testing, white box testing, grey box testing, test automation, manual testing, contextual fuzzing, emulation testing, model-driven approach, segmented evolutionary testing, and security validation. Hence, during the second phase of keywording, the concepts derived from the first phase were grouped together by considering the area of focus for each study. Thereafter, the relevant concepts such as "security testing", "compatibility testing", "functional testing" and "GUI testing" were chosen as the main categories for the classification scheme that would best represent the included studies.

Data Extraction and Mapping of Studies

Data extraction is the process of recording important information from the selected primary studies, usually in separate forms (Keele, 2007). EndNote citation management tool was used to record paper citations throughout this study. The data extraction form was designed to contain the following information: Study ID, Date of the Extraction, Paper Title, Author(s), Publication Type/ Year, Source, Study Context, App Type, Concept(s) addressed, Area of focus, Output, Limitation of the Approach, and Answers to the Research Questions.

DATA EXTRACTION RESULT

This section presents the results from the pilot search, primary search and snowballing search phases. This section also presents a classification scheme and answers to the research questions proposed.

Result from the Pilot Search Phase

In the pilot search phase, the search strings presented in Table 1 were processed through the six online databases. The results are shown in Table 2.

Table 2

Results of the pilot search

Resource	No.	Db Name	S1	S2	S3	S4	S5
Online Databases	1	IEEE	48	16	8	71	417
	2	Scopus	3	1	12	31	9
	3	Sci. Direct	3	57	82	52	126
	4	ACM	200	200	200	200	200
	5	Springer	8	138	198	200	200
	6	Web of Sci.	175	157	5	115	249
		Total	437	569	505	669	1201

Result from the Primary Search Phase

Based on the results, S5 search string generated the largest number of studies, including both relevant and irrelevant studies based on area of study, discipline and context. To achieve the publication target for this study, the studies were screened and selected according to titles and keywords (95 studies selected), followed by further screening and selection based on the abstracts. Finally, full text screening was carried out, resulting in the selection of 18 studies (see Figure 2).

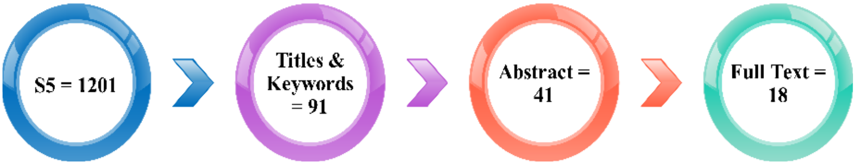

Figure 2. Primary search phase activities.

Table 3

Results of primary studies based on selection criteria

Database	Selection based on Titles and Keywords	Selection based on Abstract	Selection based on Full Text
IEEE	23	12	5
Scopus	4	2	2
Sci. Direct	4	0	0
ACM	49	23	11
Springer	6	1	0
Web of Sci.	5	3	0
Total	91	41	18

Result from the Snowballing Phase

The purpose of the snowballing search phase is to retrieve relevant primary studies that might have been missed during the primary search phase. Hence, a searching technique known as, backward snowballing, was applied to examine the reference list of the selected primary studies (Wohlin, 2014). After the evaluation of the studies based on the same screening criteria as in the primary phase, the search led to the identification of five (5) additional relevant studies. Therefore, a sum of 23 studies have been included in this SMS. Figure 3 illustrates the processes in the secondary search phase.

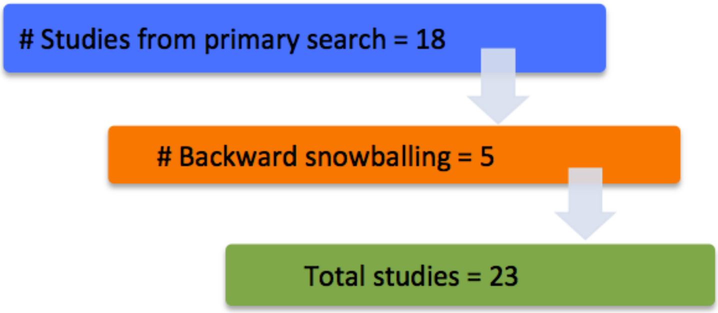
# Studies from primary search = 18
# Backward snowballing = 5
Total studies = 23

Figure 3. Flow of the study selection process.

Publication over the Years

The literature search included studies that have been published up until December 2016. Cloud-based mobile application testing has been a new and recent area of research, with the first publication appearing in 2010. Publications peaked in 2014 and 2015 with six (6) and seven (7) studies respectively. The year 2015 has shown the highest number of publications that are related to cloud-based mobile application testing (see Figure 4). The reasonable explanation for the peak values in 2014 and 2015 may be due to the progressive pace of research in the area of mobile app testing.

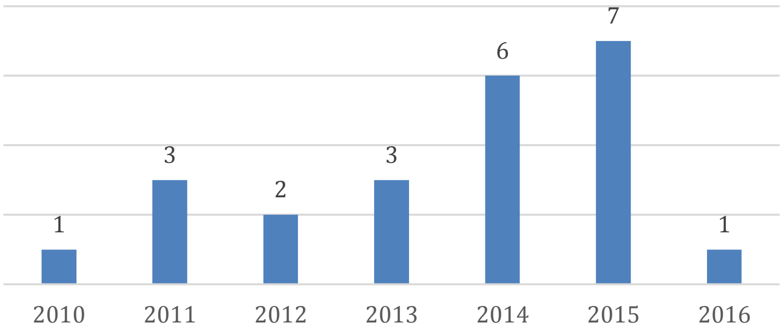

Figure 4. Publications over the years.

Publication Types

This paper classifies the studies based on the three types of publication, which are journal, conference and workshop papers. The majority of the studies are conferences with 19 studies, representing 82.6% of all studies; followed by journals with 3 studies, that represents 13.0%; and workshop with 1 study, representing 4.3%. Furthermore, the included studies are designated with the naming convention; CMT#, whereby CMT stands for cloud-based mobile testing ,and # is the sequential number (1, 2, 3…) of the particular study (see the full list of included studies in Appendix A).

Mobile Apps Platforms

In this SMS, four categories of mobile apps operating systems or platforms are identified, which are *Android*, *Windows*, *General* and *Others*. The *General* category consists of studies on proposed solutions or approaches that are focused on more than one platform (i.e. their solution(s) can be applied to any mobile application platform). The *Others* category are for studies that do not specify any mobile platform, and those that comprises of mobile app platforms that are not included in the first three categories (*Android, Windows and General)* such as Web OS. The majority of the studies (60.9%) evaluated their proposed approaches using the Android platform. This may be due to the high popularity and flexibility of the Android platform (Villanes et al., 2015). As a whole, the *General* category represents 26.1%; *Windows*, 4.3% and *Others*, 8.7%. Figure 5 shows the statistics of the mobile app platforms used in the included studies. The list of studies based on the type of mobile application platforms used are presented in Table 4.

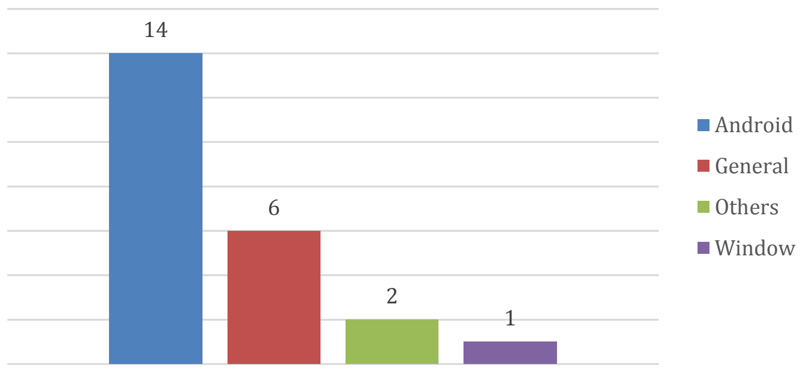

Figure 5. Mobile application platforms.

Table 4

List of studies based on mobile app platforms

Platform	Studies	#Studies	Percentage
Window	[CMT4]	1	4.3%
Others	[CMT14, CMT21]	2	8.7%
General	[CMT1, CMT8, CMT13, CMT15, CMT16, CMT19]	6	26.1%
Android	[CMT2, CMT3, CMT5, CMT6, CMT7, CMT9, CMT10, CMT11, CMT12, CMT17, CMT18, CMT20, CMT22, CMT23]	14	60.9%

Classification Scheme

The classification scheme consists of four classes which are; 1) Type of mobile apps testing, 2) contribution facet, 3) research approaches, and 4) Testing as a Service (TaaS). The first class, type of mobile apps testing answers the research question RQ1. During keywording of abstract, the security, compatibility, functionality, GUI, and automated testing were the main concepts centred around mobile application testing in the included studies. Other mobile application testing topics that fell outside of these categories were classified as general testing.

The second and the third classes of the classification scheme, which are the contribution and research approach facets respectively, answers the research question, RQ2. The contribution facets suggested in the research study conducted by (Shahrokni & Feldt, 2013) was used to address RQ2. In this SMS, the existing category of the contribution facets were applied and tabulated in Table 5.

Table 5

Contribution facets

Category	Description
Framework	A well-structured and detailed method, with wide scope and purpose that focuses on a number of research questions or areas
Model	Provides an abstraction view of a topic and problems, rather than a tangible and specific approach for solving specific problems
Tool	Provides means of evaluating a concept using specific tools
Evaluation	A technique used to empirically measure the proposed solution(s)
Metric	Provides guidelines for measuring particular phenomena
Method	Focuses on a more specific goal with a narrow research question or purpose

With respect to research approaches, the classification of research approaches suggested in (Wieringa, Maiden, Mead, & Rolland, 2006) was applied.

These included validation research, evaluation research, solution proposal, philosophical paper, opinion paper and experience paper. Table 6 lists a brief description for each approaches.

Table 6

Research approaches

Category	Description
Validation	Techniques investigated are novel and have not yet been implemented in practice. For example, techniques used in this category are experiments; i.e., work done in the laboratory.
Evaluation	Techniques are implemented in practice and an evaluation of the technique is conducted. The process of implementation is described, along with the consequences of the implementation with regards to benefits and drawbacks. It also includes the identification of problems in industry.
Solution Proposal	A solution for a problem is proposed and the solution may be either novel or a significant extension of an existing technique. The potential benefits and the applicability of the solution is described using a small example.

The fourth class in the scheme answers the research question, RQ3. Testing as a service is divided into two categories: TaaS and Non-TaaS. These categories group the included studies, and focuses on the testing as a service category to potentially develop and enhance mobile application testing.

RESULTS AND ANALYSIS

The aim of this SMS is to identify and systematically classify the evidences in the literature on cloud-based mobile application testing, and investigate the effects of TaaS. Hence, the answers to the research questions are structured according to the approach dimensions of the included studies.

RQ1: What are the types of cloud-based mobile app testing that are proposed in the studies?

Mobile app testing can be done through various approaches, techniques, and strategies. However, the type of mobile testing employed by each included study was specifically concentrated on mobile app testing that addresses the context of cloud computing. In this study, a number of cloud-based mobile app testing types were discovered from the included primary studies. Based on the dimensions of the types of testing used in the selected studies, the types of testing using a thematic analysis were carried out (Braun & Clarke, 2006) and a classification scheme (Petersen et al., 2008) was established. The types of testing were simplified into the following categories: *security*, *compatibility*, *functionality*, *GUI, automated* and *General* category. The 'general' category represents types of mobile apps testing that are not classified with the high-level contributing categories mentioned earlier. The classification method used in this SMS and the resulting categories are depicted in Figure 6.

The results show that automated testing (6 studies, 26%) is the most frequent type of cloud-based mobile app testing that is widely studied, followed by security testing. The distribution of the selected studies according to the type of testing that are employed is summarized in

Table 7, and the details of the summary are described in the following sub-sections.

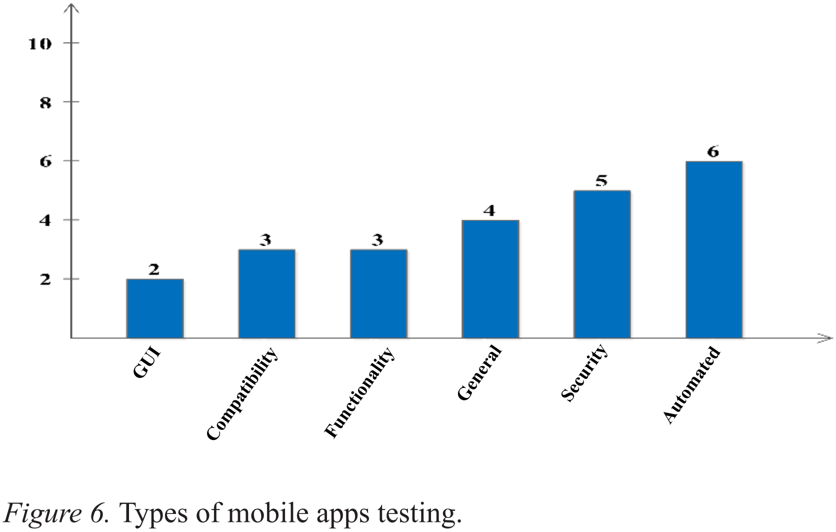

Figure 6. Types of mobile apps testing.

Table 7

Distribution of selected studies according to type of mobile app testing

No.	Type of Testing	# Study	Study(s)
1	Security	5	CMT2, CMT5, CMT7, CMT15, CMT21
2	Compatibility	3	CMT9, CMT13, CMT19
3	Functionality	3	CMT3, CMT12, CMT18
4	GUI	2	CMT4, CMT22
5	Automated	6	CMT1, CMT6, CMT10, CMT14, CMT17, CMT23
6	General	4	CMT8, CMT11, CMT16, CMT20

Automated Testing

Automated testing is defined as the use of a special software to control the execution of tests on another software or application (De Oliveira & Duarte, 2013). This process is used to compare the actual test results with the predicted results, thus improving testing efficiency (Tao et al., 2015). Based on the SMS in this paper, six (6) studies were focused on automated testing. Table 8 records the classification of automated testing studies according to testing technique and research focus. Model-driven refers to a software development methodology that creates conceptual domain models for abstract representation of the domain knowledge (Ridene & Barbier, 2011). Evolutionary testing approach is a search-based testing method that allows individuals to correspond to a particular test case with a large sample size of participants which evolves to maximize code coverage (Mahmood et al., 2014).

There is only one study [CMT1] that proposes a model-driven approach for automating mobile application checking. This approach employed the Domain Specific Modeling Language to enable the testing of remote mobile devices, which have variable features in the context of software product line engineering. A Mobile Application Testing Language (MATeL) was proposed to allow the descriptions of test scenarios, for which the expression of commonality and variability between mobile phones can be conducted. However, the approach fails to demonstrate how different types of mobile apps testing are executed automatically.

Table 8

Classification of automated testing

Technique	Study Focus	Platform	Study
Model-driven (Domain-specific modeling language)	Mobile application checking/ variability management	General	CMT1
Program analysis/ evolutionary algorithm	Generation of test case for code coverage	Android	CMT6
Automated testing	Testing and analysis	Others	CMT10
Scheduling and dispatching algorithm	Utilization of computing resources	Others	CMT14
Soft-link automated testing	Remote mobile testing	Android	CMT17
AM-TaaS	Emulation of mobile devices	Android	CMT23

Furthermore, there is only one study, [CMT6], that introduces a white box testing approach called EvoDroid, an evolutionary system-testing framework. The framework focuses on automated testing of Android apps. This approach supports automatic generation of abstract model from the app's behavior to enable automated testing. [CMT6] study further describes the testing techniques such as program analysis, evolutionary testing, and system testing for mobile Android apps, in relation to the genetic makeup of an individual. Although the study presented a novel framework, there were a few limitations, which are: 1) confined to Android apps, 2) limitation of search-based algorithm, and 3) constraints in generating apps models which uses third party libraries.

A framework for an Android testing toolkit (ATT) was designed and implemented in [CMT10] based on master-slave model. In this approach, the coordinator runs on a master JVM that interacts with multiple slave nodes. ATT consists of integrated tools and offers a set of application program interfaces (APIs), which includes: user interface (UI), sensory and system event generation; system control, profiling, monitoring and system state query; and program instrumentation and resigning. However, the approach implemented was limited to Android apps and failed to include other types of apps testing. In [CMT14], an automated testing platform TaaS was proposed to overcome resource utilization problems in computing. The proposed platform used techniques in cloud computing to build elastic resource infrastructures that provided different kinds of testing services to testers. Although the approach focused on unit testing, the authors presented an excellent TaaS solution that improves the use of computing resources. In [CMT17], a soft-link based system called remote mobile test system (RMTS) was presented to improve the efficiency of mobile apps testing and ensure mobile application quality and reliability. The aim was to minimize the cost of purchasing mobile devices for testing new applications. The approach presented a cost-effective mobile apps testing strategy, however, lacks detailed description on the idiosyncrasies of mobile apps testing and the outcome of the testing.

In another research [CMT23], a framework known as automated mobile TaaS (AM-TaaS) was introduced to offer software test services due to the high popularity of cloud services and the wide range of usage of mobile devices with different platforms. The framework is based on cloud resources mobile emulation testing, aided with AQuA's guidelines (Villanes et al., 2015). It emulates a selected mobile device based on the APK file and test cases, and subsequently performs the selected test cases automatically for the uploaded mobile app. The authors presented compelling evidence on the application of the proposed framework. However, the approach was only limited to Android applications.

Security Testing

Security testing or *negative testing* (e.g. fuzz testing) is a method that feeds malformed and unexpected input data into a program to detect security vulnerabilities in an application (Mahmood et al., 2012). This method was implemented to examine the application data and network security (Class, 2015). In this category, five cloud-based mobile security-testing approaches were reviewed.

In [CMT2], a white box approach to automate Android mobile application security in a cloud was presented. The paper proposed a framework that suggested the use of call graph and architectural models to automatically generate test cases that are intended to overcome mobile app security vulnerabilities. Similarly, [CMT5] presented a hybrid framework based on white box, black box and environment forensic approaches to address the security assessment challenges for Android applications by applying static, dynamic and application footprint analysis. The authors in [CMT2] and

[CMT5] presented convincing empirical result and methodology, however, both approaches were only limited to Android applications. In [CMT7], a cloud platform based automated security testing system was implemented to resolve the challenges of traditional testing methods. Although the approach was centered on security testing of Android mobile apps, this approach was coupled with other testing techniques such as functional testing and analysis of mobile apps. Nevertheless, the approach required more reliable testing tools, and should have expanded the study to other operating systems (OS) to validate the security of the mobile apps. In [CMT15], a sketch on the high-level design of *AppInspector*, an automated security validation system for analyzing mobile applications was presented. The system generated reports on potential security threats and private violations that ensures rigorous test coverage and scalability in the large numbers of mobile apps. [CMT21] proposed and implemented a framework for a web security to leverage testing as a service in cloud environment. From the studies of [CMT15] and [CMT21], the approaches discussed had a few drawbacks, which are; the approach in [CMT15] required full implementation and should have undergone thorough validation of the proposed design, while the approach in [CMT21] failed to report the effects of applying the approach to any other mobile app platform. Security testing is further classified based on the testing technique, focus of the technique and the mobile app platform used in the approaches, as shown in Table 9.

Table 9

Classification of security testing

Technique	Focus	Platform	Study
Program Analysis	Automation of mobile security	Android	CMT2
Environmental Forensic/ Data Fusion	Security Access		CMT5
Functional test	Mobile Internet		CMT7
Security Validation	Analysis and generation of security report	General	CMT15
Security TaaS	Web vulnerability	Others	CMT21

Compatibility Testing

Compatibility testing is a mobile application testing that ensures a particular mobile app runs effectively on all relevant devices and on different versions of operating systems that are available in the market (Haller, 2013).

In this SMS, three (3) studies were identified that addressed the issue of compatibility testing [CMT9, CMT13, CMT19]. A strategy for compatibility testing was proposed in [CMT9] as a solution to overcome the challenges faced in mobile applications, due the complexity of mobile app and devices. The authors proposed a tree model that comprises the K-Means statistical algorithms to generate an optimized compatibility test sequence for mobile app. However, the proposed approach was only limited to Android mobile app. In addition, the authors failed to describe any tool support for facilitating the compatibility testing process of the mobile apps. In the study [CMT13], a mobile cloud testing framework known as AM-TaaS, was proposed to improve the quality of mobile applications. This framework was based on automated testing that offers mobile cloud testing services. The framework used emulated devices to reduce the cost and complexity of mobile testing. The approach is suitable as the proposed framework is holistic in its application on any type of mobile app platforms. Based on the study [CMT19], a mobile testing that incorporates Swisscom IT Service's mobile testing framework was presented. The study employed a user-centric testing approach and addressed the need for automated testing based on two reasons; agility and compatibility testing. Although the approach detailed the essential steps to implement automated testing, the authors however, did not present convincing results from their evaluation. Table 10 shows the classification of compatibility testing according to the technique, focus of the technique and the platform used in the included studies.

Table 10

Classification of compatibility testing

Technique	Focus	Platform	Study
Clustering Algorithm	Cost effective testing	Android	CMT9
	Functional testing		CMT13
Automated Testing	Software quality management	General	CMT19

Functionality Testing

Functionality testing is the mobile app testing that determines the effectiveness of every function of the mobile application (Zhang & Pi, 2015). In other words, functional test enables the software tester to examine the behaviour of an application that follows the expected manner in certain tasks (Kaur & Kaur, 2016)(Class, 2015).

In an attempt to improve TaaS in mobile development, a universal TaaS platform known as cloud testing of mobile systems (CTOMS) was proposed in [CMT3]. The approach presented an integrated solution for mobile TaaS development, and studied the functionality for testing mobile applications using coverage, connection, functional and UI testing techniques. The authors described the features and architecture of CTOMS comprehensively, however, neglected to present the explanation on the feasibility of study results.

A functional test method based on TaaS platform for mobile app was proposed in [CMT12]. In this study, a test script was generated automatically based on the functional traversal. The tests were then conducted on diverse cloud-based devices, whereby test errors were discovered. Although the approach presented algorithms and empirical results, the authors limited the study on mobile Android apps and narrowed the methodology to only functional test. In [CMT18], CTOMS was extended and presented to enable tests to be analyzed on a variety of remote mobile devices through multidirectional testing. This framework was improved to support more types of testing such as security and performance testing. Although CTOMS was improved, further evaluations are required in the appropriate case study. Table 11 shows the three studies (all tested in Android platform) that focuses on the different functionality testing.

Table 11

Functionality testing techniques

Technique	Focus	Study
TaaS	Android mobile development	CMT3
Functional traversal method using TaaS	General mobile application testing	CMT12
TaaS	Heterogeneous network system	CMT18

GUI Testing

GUI testing examines different devices to provide an adequate rendering of data, and ensures that the native application is correctly displayed on different devices (Kirubakaran & Karthikeyani, 2013). In this SMS, two studies [CMT4, CMT22] were identified that specifically addresses GUI testing (see Table 12).

In the study [CMT4], a cloud service framework known as *Caiipa*, that utilizes the contextual fuzzing approach was proposed. The purpose of this approach is to test the sustainability of mobile apps over the expanded mobile context space. Although Caiipa is a cloud-based service, the framework does not investigate TaaS in leveraging cloud-based testing services. The approach was evaluated using mobile Windows platform. The framework was restricted to one platform rather than encompassing other mobile app platforms. In the study [CMT22], an A/B testing of native mobile application was presented. The approach offered a flexible multivariate testing that compares the higher number of variants, through the combination of variations for different sections of the user interface. An aspect-oriented programming was used in this approach to relieve the burden of the application programmer to write boiler-plate codes related to user interface applications. The study presented a strong testing approach, but needed to broaden its implementations to other mobile platforms such as iOS and Windows.

Table 12

Approaches in GUI testing

Technique	Focus	Platform	Study
Contextual fuzzing	Performance evaluation	Window	CMT4
Multivariate testing	Mobile application testing	Android	CMT22

General

This category consists of mobile app testing approaches that are not included in the high-level contributing categories mentioned earlier. These include; general application testing which was employed in [CMT8], [CMT11] and [CMT16] studies, and load, performance and functional testing in the [CMT20] study. In the [CMT8] study, an approach was proposed to set up a cloud-based infrastructure as a service for mobile TaaS. The proposed mobile infrastructure environment supported the provisioning, managing and billing service of mobile cloud services across different clouds. However, the results from the study only included the use of Android mobile platform, and disregarded any discussion to accommodate other platforms. Other than that, a resource adaptation strategy was proposed in [CMT11]. This approach was coupled with the application testing to improve the utilization of resources on cloud-based testing platform. The experiment conducted in the study showed promising results in improving the resource utilization of a cloud testing platform. However, to conclude the reliability of the approach, additional experiments are required to test the diverse mobile apps devices and platforms. A model for a cloud-based software testing was introduced in [CMT16]. The model provided solutions for mobile app testing challenges by using emulators. Unfortunately, the authors failed to present detailed experimental results to validate the proposed model. In [CMT20], mobile testing as a service framework in a cloud environment, MTAAS was presented. The framework allowed the testing of many mobile applications in different portable devices and mobile platforms. The main advantage of this approach is that it allows the implementation of the framework to conduct three types of mobile app testing, which are load, performance and functional testing. Nevertheless, these types of mobile apps testing are not particularly developed for mobile Android platform. Table 13 shows the included studies according to the different testing techniques such as TaaS, resource adaptation strategy and automated testing.

Table 13

Classification of general testing

Technique	Focus	Platform	Study
TaaS	Testing infrastructure	General	CMT8
	Testing on smart devices	Android	CMT20
Resource adaptation strategy	Resource utilization		CMT11
Automated testing	Utilization of cloud resources	General	CMT16

RQ2: What research approaches do these studies apply, and what contribution facet(s) do they provide (e.g. framework, method, tool)?

The question is intended to address the type of research approaches and the contributions reported in literature that specifically investigates cloud-based mobile app testing. To answer this question, the studies were evaluated according to the research approaches and empirical contributions in the literature to solve peculiar cloud-based mobile apps testing problems. Three (3) research approaches applicable to the included studies were identified. Figure 7 presents the distribution of the included studies according to the three types of research approach. A total of ten (10) studies were conducted on validation research, eight (8) on evaluation research, and five (5) on solution proposal. None of the studies implemented philosophical, opinion or experience based research approaches. The research approaches and list of studies are presented in Table 14.

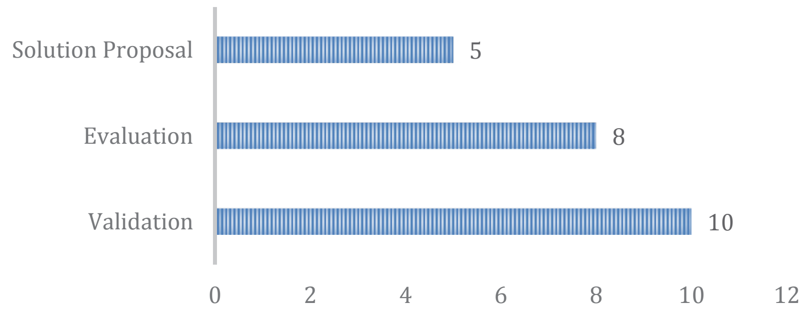

Figure 7. Research type used by included studies.

Table 14

Research approaches

Research Approach	Study(s)	Number of Study
Validation Research	10	CMT3, CMT6, CMT7, CMT8, CMT9, CMT13, CMT14, CMT15, CMT19, CMT22

(continued)

Research Approach	Study(s)	Number of Study
Evaluation Research	8	CMT1, CMT4, CMT10, CMT11, CMT12, CMT16, CMT17, CMT23
Solution Proposal	5	CMT2, CMT5, CMT18, CMT20, CMT21

Figure 8 illustrates the contribution facets of the included studies, which comprises four (4) categories: *framework*, *method*, *tool* and *model*. These categories are inspired from (Shahrokni & Feldt, 2013). Although some studies contribute to more than one facet, for example, framework and supported tool, the studies are essentially grouped according to the predominant contribution facet. For example, studies that identified tool as its major contribution, were subsequently categorized as *tool*. From Figure 7, the *framework* category has the largest number of studies (14 studies) that represents 60.1%, followed by *method* and *tool* categories (4 studies respectively), at 17.4%. The category with the least number of studies is *model* (only one study), at 4.3%. Table 15 shows the list of included studies and description for each contribution facet.

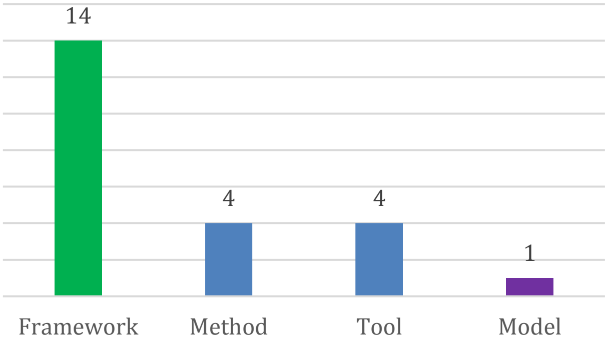

Figure 8. Contribution facets.

Table 15

Contribution facets for each study

Contribution Facet	Number of Study	Study	Name/ Description
Framework	14	CMT1	MaTel- Mobile Application Testing Language Framework.
		CMT2	Android Framework for Automated Security Testing.
		CMT4	Caiipa- A testing framework using contextual fuzzing.
		CMT5	Hybrid Security Assessment Framework.
		CMT6	EvoDroid- Evolutionary computing framework.
		CMT10	ATT- Android Testing Toolkit Framework.
		CMT11	A resource adaptation dynamic strategy for enabling cloud testing platform (CTP) to add or remove Virtual Machines, based on workload of CTP and number of devices.
		CMT13	AM-TaaS- A mobile testing framework that facilitates the test environment setup and configuration that covers a range of mobile devices and platforms.
		CMT18	CTOMS-Cloud Testing of Mobile System framework.
		CMT19	Swisscom mobile testing framework.
		CMT20	MTAAS- Mobile testing as a service framework.
		CMT21	Framework for Security Testing as a Service (TaaS).
		CMT22	A/B and Multivariate Testing Framework.
		CMT23	AM-TaaS- Automated mobile testing as a service framework.
Method	4	CMT8	MIaaS- A method for setting up mobile infrastructure as a service
		CMT9	Systematic and cost-effective mobile compatibility testing method based on tree model using K-Means Algorithm.
		CMT12	Functional test method based on testing as a service platform.
		CMT14	Scheduling and dispatching methods for improving the utilization of computing resources based on TaaS.

(continued)

Contribution Facet	Number of Study	Study	Name/ Description
Tool	4	CMT3	CTOMS- A tool for cloud testing of mobile.
		CMT7	Metasploit- Permeable Automated Testing Tool.
		CMT15	AppInspector- Automated Security Testing and Validation System.
		CMT17	RMTS- Remote Mobile Test System.
Model	1	CMT16	A cloud-based model for setting up a mobile environment, actual device and platform on the cloud resources, based on the given system configuration.

Based on the categories of contribution facets and the type of mobile app testing, the studies were mapped using a bubble chart. The y-axis represents the type of mobile app testing and the x-axis represents the contribution facet of the studies (see Figure 9). From Figure 9, the major contribution facet, *framework*, contains studies that span all types of mobile apps testing. However, the *model* contribution facet has only one type of mobile app testing, which is General. The contribution facets for GUI testing was limited to only framework-based. The other types of mobile apps testing that have lower contributions include, compatibility and functionality testing.

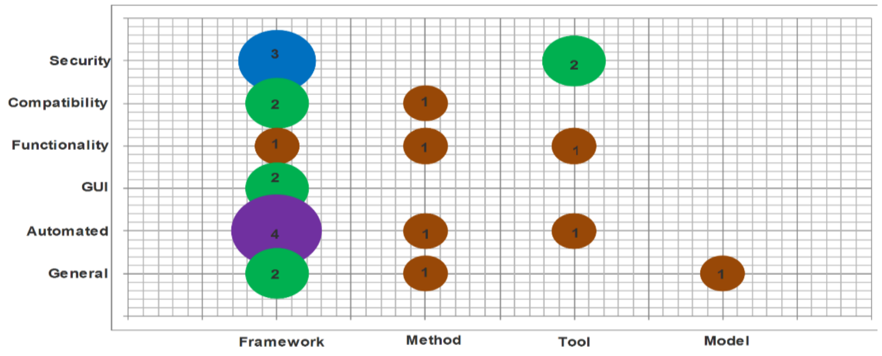

Figure 9. Types of Mobile Apps Testing vs Contribution Facet.

RQ3: How mobile app testing as a service (TaaS) affects the process of mobile app testing?

TaaS is a cloud based testing service (Tao et al., 2015) that is popular for its scalable testing environment (Villanes et al., 2015) that offers cost-effective and on-demand testing services and utility-based service models (Al-Ahmad et al., 2013; Chana & Rana, 2012; Gao et al., 2012; Tung et al., 2014; Villanes et al., 2015). TaaS has been applied in various software platforms to enhance Regression Test, Web Security, Unit Testing and Load Web Testing (Villanes et al., 2015). Furthermore, mobile TaaS has utilized cloud infrastructure based on 'pay-as-you-test' business model to achieve resource sharing at an affordable cost (Rojas et al., 2016; Villanes et al., 2015). Hence, mobile TaaS is capable of testing different mobile apps that are installed in different devices and uses different operating systems (Malini et al., 2014). This research question was to identify the influence of TaaS on mobile app testing process. From the list of the included studies, 11 studies (47.8%) suggested that TaaS is the key to successful mobile app testing. Table 16 lists the positive outcomes of TaaS on the different perspectives that were considered in the included studies.

Table 16

Description of the Influence of TaaS in the Primary Studies

Studies	Influence of TaaS
CMT3	Integration of different functionality for testing mobile apps
CMT7	TaaS facilitates the simulation of real network environment of mobile and tests the diversity of mobile device terminals
CMT8	Provision of cloud infrastructure and support tools for testing diverse mobile apps at an affordable cost
CMT9	TaaS enhances compatibility testing at anytime and anywhere, cost reduction, elasticity in test automation, and demand on service for mobile compatibility testing

(continued)

Studies	Influence of TaaS
CMT12	TaaS provides an integrated test environment for Android apps
CMT13	Facilitation of setups and configuration for test environment, Coverage of mobile devices and platforms
CMT14	Building of elastic resource infrastructures and provision of various testing services to users at low-cost
CMT18	TaaS enables the testing of variety of remote mobile devices
CMT20	TaaS helps in providing more realistic results through the testing of real devices
CMT21, CMT23	TaaS provides testing capabilities and on-demand testing to customers

The remaining 12 studies [CMT1, CMT2, CMT4, CMT5, CMT6, CMT10, CMT11, CMT15, CMT16, CMT17, CMT19 and CMT22] representing 52.2% of the studies proposed different approaches, other than TaaS. The main approaches that were discussed and elaborated in the studies are as follows: [CMT1]; Domain-specific modelling language, [CMT2]; program analysis; [CMT4]; contextual fuzzing, [CMT5]; information fusion, [CMT6]; evolutionary testing and program analysis, [CMT10]; scalable testing and analysis, [CMT11]; resource adaptation strategy, [CMT15]; security validation and analysis, [CMT16]; adaptable and on-demand network access, [CMT17]; remote mobile test system, [CMT19]; user-centric testing, and [CMT22]; A/B and multivariate testing. In addition to the distribution of studies that included TaaS, contribution facets of the studies mentioned above was plotted against the TaaS category. Figure 10 shows a bubble chart for the contribution facets (Y-axis) against TaaS category (X-axis). There is a high number of studies that implemented approaches with TaaS that involves the *Framework*, *Method* and *Tool* contribution facets. However, there is no study that included TaaS in the *Model* contribution facet. Furthermore, the *Framework* contribution facet contained the highest number of studies (14 studies), despite the majority of the studies (9 studies) do not employed TaaS.

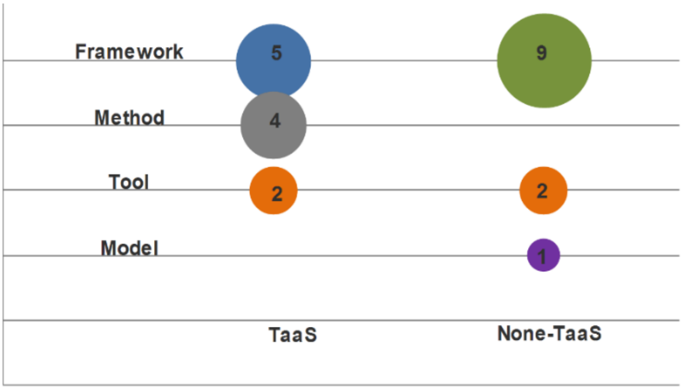

Figure 10. Contribution facet against testing as a service category.

DISCUSSION

In this section, a discussion is presented on the research implication from the results of the mapping study, and reviews the limitations of this paper.

Contribution facets of included studies

Based on the selection criteria, 23 primary studies were included in this SMS, whereby 18 studies were identified from the primary search phase and 5 studies from the secondary search phase. The majority of the studies were conference publications (82.6%), followed by journals (13%). Furthermore, the range of publications have been between the year 2010 and 2016 that evidently suggest that cloud based mobile app testing is a new area of research, compared to research conducted on traditional mobile testing approaches. To answer the proposed research questions, the included studies were categorized according to the types of mobile app testing, research approaches, the contribution facets (i.e. framework, method, tool, and model) and the effects of TaaS. The majority of the contributions (55.6%) were in the form of frameworks with some tool supports. In addition, there is a lack of contributions with regards to metric (for example, performance) and evaluation research (for example, experiment or case study). Moreover, it is paramount to provide further convincing results that relates to feasibility, efficiency and validity aspects. Two studies [CMT8, CMT9] representing 8.7%, reported case studies to evaluate their proposals (methods/ algorithm).

We recommend that future research on cloud-based mobile app testing should focus on research contribution facet such as metric, and perform empirical evaluation studies to improve and test the significance and validity of the testing approaches. In addition, researchers and practitioners should expand the techniques to accommodate other mobile platforms other than Android applications.

Cloud-based mobile application testing approaches

Six (6) categories of approaches in mobile app testing were identified, which comprises *security*, *compatibility*, *functionality*, *GUI*, *automated* and *general* category. The results suggest that there has been a lack of research publications that combines the approaches with multiple testing techniques. In other words, approaches that can be used to perform any type of mobile app testing in cloud for different mobile apps are not conducted or evaluated. In addition, all the included studies involved cloud-based approaches, and were expected to exploit cloud infrastructure and services through test automation. The main goal of software testing is test automation, which aims to significantly reduce cost, minimize human error and simplify regression testing (Yu et al., 2010). Although testing using emulators does have its limitations (Ridene & Barbier, 2011), the use of emulators and real devices when testing mobile application environments and complexity are critical to achieve successful mobile application testing (Baride & Dutta, 2011). However, only six studies [CMT1, CMT6, CMT10, CMT14, CMT17, CMT23], representing 26.1%, explicitly focused on test automation. From these findings, there are further research opportunities on cloud based mobile testing particularly on the scalability of frameworks to accommodate various types of testing that are applicable to different mobile devices and platforms. Further consideration on test automation would also complement the contributions of existing approaches, where testing techniques are considerably more or less manual.

Moreover, there is a lack of research on other aspects of cloud-based mobile application testing such as performance, regression, load, usability, interrupt, memory leakage, installation, operational, laboratory, certification, location and outdated software testing. Hence, we recommend researchers and practitioners in cloud-based mobile app testing to consider these types of mobile app testing in future research to leverage the benefits of cloud-based mobile app testing. For example, mobile certification testing helps testers to discover how mobile devices are certified according to compliance guidelines set by different mobile platforms (Candea et al., 2010). The certification of compliance ensures the standard and quality of mobile devices, and its compatibility with platforms. Location testing assists in tackling network connectivity fluctuations against various geographical locations (Wang, Green, & Malkawi, 2002). Another important type of mobile application testing is installation. Mobile applications should be installed smoothly without any error. Installation testing is conducted to verify proper installations, updates and uninstallation of mobile applications by users (Myers, Sandler, & Badgett, 2011).

Testing as a service

Mobile applications has become ubiquitous (Holl et al., 2016) and their exponential growth in recent years has evolved the concepts of mobile TaaS (Villanes et al., 2015). The aim of TaaS is to provide the capability to test different mobile apps, installed in different devices by using different operating systems at lower cost (Malini et al., 2014). Although, a number of studies (47.8%) in this SMS addressed and reported various benefits on TaaS, the majority of the studies (52.2%) presented approaches that did not involve the services of TaaS. This suggests that further research is needed on the applicability of mobile TaaS. Thus, it is important for researchers and testers to discover scalable approaches to integrate mobile TaaS to exploit the benefits of cloud environment.

Finally, no study was conducted to examine the reuse of requirements (RR) in developing mobile application testing tool or system. RR is a process of storing which represents a set of requirements that are used in the implementation of future developments of similar and variant applications across diverse products and projects (Benfield, 2015). RR improves the productivity and quality of the application (Ya'u, Nordin, & Salleh, 2016, 2017). Therefore, researchers should practise the reuse of the requirements for developing mobile app testing tools to exploit its potential benefits, while improving productivity and quality of cloud based mobile app testing products.

Threats to validity

It is important to review the limitations that might affect the validity of this SMS. For example, this paper limited the primary search to electronic databases and therefore, may potentially overlooked relevant conference and journal papers that were not published online. Furthermore, mobile and smart phone applications and cloud based mobile application testing are deemed to be relatively new research fields (Zein et al., 2016). Hence, to avoid missing any publications, the search period was extended to December 2016 to cover all relevant papers since the inception of the online database used in this SMS. Furthermore, the online search was coupled with backward snowballing to obtain all possible relevant papers. Other than that, the criteria for inclusion and exclusion used in this SMS would require further review to identify its effects on the validity of this paper. Guidelines from existing research by (Petersen et al., 2015; Salleh et al., 2011; Zein et al., 2016) was implemented in conducting selection, extraction and mapping exercises to enhance the quality of the SMS. There are possibilities for personal or author bias, since the data extraction activity was performed independently by the researchers. This could potentially influence the accuracy of the extracted data. Therefore, a discussion was conducted to resolved any conflicts or disagreement in a joint meeting to ensure the data was obtained objectively.

CONCLUSION

This study presented a systematic mapping study on cloud based mobile application testing. Out of a total of 95 primary studies, the selection and screening of the studies were conducted based on titles and keywords that were chosen according to the inclusion and exclusion criteria. 23 primary studies were finally included. The 23 studies were identified through the primary search phase (18 studies) by using online databases, and secondary search phase (5 studies) by using backward snowballing. The findings of this SMS suggests that research on cloud based mobile application testing is relatively new as the oldest primary study that was included in this review was published in 2010. Furthermore, the majority of the included studies were published in conference proceedings that represents 82.6% out of the total included studies. This indicates that the area of research is still new, thus providing abundant opportunities for current and future research. The results suggests that the majority of the studies (60.9%) were contributed through frameworks, followed by method and tool that represents 17.4% each. Moreover, there are insufficient research that contributes to literature, particularly for models and empirical evaluation (experimental or cases studies) that are employed to evaluate the proposed approaches. The majority of the studies (60.9%) were focused on Android mobile apps and neglected the use of other renowned mobile platforms such as IOS, Blackberry, Symbian and Web OS. For instance, only six studies (26.1%) proposed a general approach that accommodated any mobile app platform. This indicates that there has been a lack of general approaches in literature on the study of the scalability for testing mobile apps from different OS or platforms. The types of mobile app testing were classified into six categories, which are security, compatibility, functionality, GUI, Automated and General. The results show that there is a lack of approaches that offer solution of mobile app testing which is not tied to any specific mobile platform. Based on these findings, the implications from this research which included areas of weaknesses and suggestions on the need to complement the existing studies with scalable cloud-based approaches based on TaaS were reviewed. These approaches would have the capacity to support as many types of testing that can be run on all existing mobile app platforms. The findings from this SMS could benefit researchers and practitioners, especially mobile application testers to identify and enhance the advancement of cloud based mobile application testing and pave the way for future research.

ACKNOWLEDGEMENT

This research was funded by the Ministry of Higher Education Malaysia under FRGS research grant (FRGS14-125-0366).

REFERENCES

Al-Ahmad, A. S., Aljunid, S. A., & Sani, A. S. A. (2013, 23-24 Dec. 2013). Mobile cloud computing testing review. Paper presented at the *Advanced Computer Science Applications and Technologies* (*ACSAT*), 2013 International Conference on.

Bai, X., Li, M., Chen, B., Tsai, W.-T., & Gao, J. (2011). Cloud testing tools. Paper presented at the *Service Oriented System Engineering* (*SOSE*), 2011 IEEE 6th International Symposium on.

Baride, S., & Dutta, K. (2011). A cloud based software testing paradigm for mobile applications. *SIGSOFT Softw. Eng. Notes, 36*(3), 1-4. doi:10.1145/1968587.1968601

Benfield, J. (2015). How reuse can help you define world-class compliance requirements. Retrieved from http://www.blueprintsys.com/how-reuse-can-help-you-define-world-class-compliance-requirements/

Bhattacharya, A., & De, P. (2016). A survey of adaptation techniques in computation offloading. *Journal of Network and Computer Applications*.

Black, R. (2002). *Managing the testing process*: John Wiley & Sons.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology, 3*(2), 77-101.

Candea, G., Bucur, S., & Zamfir, C. (2010). Automated software testing as a service. Paper presented at the *Proceedings of the 1st ACM Symposium on Cloud Computing*, Indianapolis, Indiana, USA.

Chana, I., & Rana, A. (2012). Empirical evaluation of cloud-based testing techniques: A systematic review. *ACM SIGSOFT Software Engineering Notes, 37*(3), 1-9.

Choudhary, S. R., Gorla, A., & Orso, A. (2015, 9-13 Nov. 2015). *Automated Test Input Generation for Android: Are We There Yet? (E).* Paper presented at the Automated Software Engineering (ASE), 2015 30th IEEE/ACM International Conference on.

Class, S. T. (2015). Types of Mobile App Testing. Retrieved from http://www.softwaretestingclass.com/types-of-mobile-app-testing-tutorial-4/

Coutinho, E. F., de Carvalho Sousa, F. R., Rego, P. A. L., Gomes, D. G., & de Souza, J. N. (2015). Elasticity in cloud computing: A survey. *Annals of telecommunications - annales des télécommunications, 70*(7), 289-309. doi:10.1007/s12243-014-0450-7

De Oliveira, G. S., & Duarte, A. (2013). *A framework for automated software testing on the cloud.* Paper presented at the 2013 International Conference on Parallel and Distributed Computing, Applications and Technologies.

Dey, S., Liu, Y., Wang, S., & Lu, Y. (2013). *Addressing response time of cloud-based mobile applications*. Paper presented at the Proceedings of the first international workshop on Mobile cloud computing &#38; networking, Bangalore, India.

Endo, P. T., Rodrigues, M., Gonçalves, G. E., Kelner, J., Sadok, D. H., & Curescu, C. (2016). High availability in clouds: Systematic review and research challenges. *Journal of Cloud Computing, 5*(1), 16.

Gao, J., Manjula, K., Roopa, P., Sumalatha, E., Bai, X., Tsai, W. T., & Uehara, T. (2012, 3-6 Dec. 2012). *A cloud-based TaaS infrastructure with tools for SaaS validation, performance and scalability evaluation.* Paper presented at the 4th IEEE International Conference on Cloud Computing Technology and Science Proceedings.

Gilbert, P., Chun, B.-G., Cox, L. P., & Jung, J. (2011). *Vision: Automated security validation of mobile apps at app markets*. Paper presented at the Proceedings of the second international workshop on Mobile cloud computing and services, Bethesda, Maryland, USA.

Haller, K. (2013). Mobile Testing. *SIGSOFT Softw. Eng. Notes, 38*(6), 1-8. doi:10.1145/2532780.2532813

Holl, K., Vieira, V., & Faria, I. (2016). An Approach for Evaluating and Improving the Test Processes of Mobile Application Developments. *Procedia Computer Science, 94*, 33-40. doi:http://dx.doi.org/10.1016/j.procs.2016.08.009

Holzmann, C., & Hutflesz, P. (2014). *Multivariate Testing of Native Mobile Applications*. Paper presented at the Proceedings of the 12th International Conference on Advances in Mobile Computing and Multimedia, Kaohsiung, Taiwan.

Huang, J. F., & Gong, Y. Z. (2012, 3-6 Dec. 2012). *Remote mobile test system: A mobile phone cloud for application testing*. Paper presented at the Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on.

Inçki, K., Ari, I., & Sözer, H. (2012, 20-22 June 2012). *A Survey of Software Testing in the Cloud*. Paper presented at the 2012 IEEE Sixth International Conference on Software Security and Reliability Companion.

Kaur, K., & Kaur, A. (2016, 16-18 March 2016). *Cloud era in mobile application testing*. Paper presented at the 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom).

Keele, S. (2007). Guidelines for performing systematic literature reviews in software engineering *Technical report, Ver. 2.3 EBSE Technical Report. EBSE*.

Kirubakaran, B., & Karthikeyani, V. (2013, 21-22 Feb. 2013). *Mobile Application Testing- Challenges and Solution Approach through Automation*. Paper presented at the Pattern Recognition, Informatics and Mobile Engineering (PRIME), 2013 International Conference on.

Kuo, J. Y., Liu, C. H., & Yu, W. T. (2015, June 27 2015-July 2 2015). *The Study of Cloud-Based Testing Platform for Android*. Paper presented at the 2015 IEEE International Conference on Mobile Services.

Liang, C.-J. M., Lane, N. D., Brouwers, N., Zhang, L., Karlsson, B. F., Liu, H., . . . Zhao, F. (2014). *Caiipa: Automated large-scale mobile app testing through contextual fuzzing*. Paper presented at the Proceedings of the 20th annual international conference on Mobile computing and networking, Maui, Hawaii, USA.

Liu, C. H., Chen, S. L., & Chen, W. K. (2015, June 27 2015-July 2 2015). *Improving Resource Utilization of a Cloud-Based Testing Platform for Android Applications*. Paper presented at the 2015 IEEE International Conference on Mobile Services.

Mahmood, R., Esfahani, N., Kacem, T., Mirzaei, N., Malek, S., & Stavrou, A. (2012). *A whitebox approach for automated security testing of Android applications on the cloud*. Paper presented at the 7th International Workshop on Automation of Software Test (AST), 2012.

Mahmood, R., Mirzaei, N., & Malek, S. (2014). *Evodroid: Segmented evolutionary testing of android apps*. Paper presented at the Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering.

Malini, A., Venkatesh, N., Sundarakantham, K., & Mercyshalinie, S. (2014, 11-13 Dec. 2014). *Mobile application testing on smart devices using MTAAS framework in cloud*. Paper presented at the International Conference on Computing and Communication Technologies.

Meng, Z., Jiang, Y., & Xu, C. (2015). *Facilitating reusable and scalable automated testing and analysis for android apps*. Paper presented at the Proceedings of the 7th Asia-Pacific Symposium on Internetware, Wuhan, China.

Mohiuddin, K., Islam, A., Alam, A., & Ali, A. (2012). *24X7X365: Mobile cloud access*. Paper presented at the Proceedings of the CUBE International Information Technology Conference, Pune, India.

Muccini, H., Di Francesco, A., & Esposito, P. (2012). *Software testing of mobile applications: Challenges and future research directions.* Paper presented at the Automation of Software Test (AST), 2012 7th International Workshop on.

Munoz-Escoı, F. D., & Bernabeu-Aubán, J. M. (2015). *A Survey on Elasticity Management in the PaaS Service Model*. Retrieved from

Murugesan, L., & Balasubramanian, P. (2014, 4-6 June 2014). *Cloud based mobile application testing.* Paper presented at the 2014 IEEE/ACIS 13th International Conference on Computer and Information Science (ICIS).

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

Pardeshi, S. N. (2013). Study on testing as a service on cloud. *International Journal of Advanced Computer Research, 3*(1).

Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008). *Systematic mapping studies in software engineering.* Paper presented at the 12th International Conference on Evaluation and Assessment in Software Engineering.

Petersen, K., Vakkalanka, S., & Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology, 64*, 1-18.

Rahimi, M. R., Ren, J., Liu, C. H., Vasilakos, A. V., & Venkatasubramanian, N. (2014). Mobile cloud computing: A survey, state of art and future directions. *Mobile Networks and Applications, 19*(2), 133-143. doi:10.1007/s11036-013-0477-4

Ridene, Y., & Barbier, F. (2011). *A model-driven approach for automating mobile applications testing*. Paper presented at the Proceedings of the 5th European Conference on Software Architecture: Companion Volume, Essen, Germany.

Rojas, I. K. V., Meireles, S., & Dias-Neto, A. C. (2016). *Cloud-based mobile app testing framework: Architecture, implementation and execution*. Paper presented at the Proceedings of the 1st Brazilian Symposium on Systematic and Automated Software Testing, Maringa, Parana, Brazil.

Salleh, N., Mendes, E., & Grundy, J. (2011). Empirical studies of pair programming for CS/SE teaching in higher education: A systematic literature review. *Software Engineering, IEEE Transactions on, 37*(4), 509-525.

Shahrokni, A., & Feldt, R. (2013). A systematic review of software robustness. *Information and Software Technology, 55*(1), 1-17.

Silva, F. A., Zaicaner, G., Quesado, E., Dornelas, M., Silva, B., & Maciel, P. (2016). Benchmark applications used in mobile cloud computing research: A systematic mapping study. *The Journal of Supercomputing, 72*(4), 1431-1452. doi:10.1007/s11227-016-1674-2

Starov, O., & Vilkomir, S. (2013). *Integrated TaaS platform for mobile development: Architecture solutions*. Paper presented at the Proceedings of the 8th International Workshop on Automation of Software Test, San Francisco, California.

Starov, O., Vilkomir, S., & Kharchenko, V. (2013). *Cloud testing for mobile software systems.* Paper presented at the 8th International Joint Conference on Software Technologies, in ICSOFT

Tao, C., Gao, J., & Li, B. (2015, Oct. 30 2015-Nov. 1 2015). *Cloud-Based Infrastructure for Mobile Testing as a Service.* Paper presented at the 2015 Third International Conference on Advanced Cloud and Big Data.

Tao, D., Lin, Z., & Lu, C. (2015). Cloud platform based automated security testing system for mobile internet. *Tsinghua Science and Technology, 20*(6), 537-544.

Tung, Y. H., Lin, C. C., & Shan, H. L. (2014, 7-11 April 2014). *Test as a service: A Framework for web security TaaS service in cloud environment.* Paper presented at the 2014 IEEE 8th International Symposium on Service Oriented System Engineering.

Villanes, I. K., Costa, E. A. B., & Dias-Neto, A. C. (2015, June 27 2015-July 2 2015). *Automated Mobile Testing as a Service (AM-TaaS).* Paper presented at the 2015 IEEE World Congress on Services.

Wang, S. S., Green, M., & Malkawi, M. (2002, 2002). *Mobile positioning technologies and location services.* Paper presented at the Proceedings RAWCON 2002. 2002 IEEE Radio and Wireless Conference (Cat. No.02EX573).

Wieringa, R., Maiden, N., Mead, N., & Rolland, C. (2006). Requirements engineering paper classification and evaluation criteria: A proposal and a discussion. *Requirements Engineering, 11*(1), 102-107.

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

Ya'u, B. I., Nordin, A., & Salleh, N. (2016). *Software requirements patterns and meta model: A Strategy for enhancing Requirements Reuse (RR).* Paper presented at the Information and Communication Technology for The Muslim World (ICT4M), 2016 6th International Conference on.

Ya'u, B. I., Nordin, A., & Salleh, N. (2017). Investigation of Requirements Reuse (RR) challenges and existing RR approaches. *Advanced Science Letters, 23*(5), 4101-4105. doi:10.1166/asl.2017.8232

Yu, L., Tsai, W. T., Chen, X., Liu, L., Zhao, Y., Tang, L., & Zhao, W. (2010). *Testing as a service over cloud.* Paper presented at the 2010 Fifth IEEE International Symposium on Service Oriented System Engineering.

Zein, S., Salleh, N., & Grundy, J. (2016). A systematic mapping study of mobile application testing techniques. *Journal of Systems and Software, 117*, 334-356.

Zhang, S., & Pi, B. (2015). *Mobile functional test on TaaS environment.* Paper presented at the Service-Oriented System Engineering (SOSE), 2015 IEEE Symposium on.

Zhang, T., Gao, J., Cheng, J., & Uehara, T. (2015). *Compatibility Testing Service for Mobile Applications.* Paper presented at the Service-Oriented System Engineering (SOSE), 2015 IEEE Symposium on.

Zhong, H., & Xiao, J. (2014). *Design for a cloud-based hybrid Android application security assessment framework.* Paper presented at the International Conference on Reliability, Maintainability and Safety (ICRMS).

Appendix A: Studies Included in the mapping study

[CMT1] Ridene, Y., & Barbier, F. (2011). *A model-driven approach for automating mobile applications testing*. Paper presented at the Proceedings of the 5th European Conference on Software Architecture: Companion Volume, Essen, Germany.

[CMT2] Mahmood, R., Esfahani, N., Kacem, T., Mirzaei, N., Malek, S., & Stavrou, A. (2012, 2-3 June 2012). *A whitebox approach for automated security testing of Android applications on the cloud.* Paper presented at the 7th International Workshop on Automation of Software Test (AST), 2012

[CMT3] Starov, O., & Vilkomir, S. (2013). Integrated TaaS platform for mobile development: architecture solutions. Paper presented at the *Proceedings of the 8th International Workshop on Automation of Software Test,* San Francisco, California.

[CMT4] Liang, C.-J. M., Lane, N. D., Brouwers, N., Zhang, L., Karlsson, B. F., Liu, H., . . . Zhao, F. (2014). Caiipa: automated large-scale mobile app testing through contextual fuzzing. Paper presented at the *Proceedings of the 20th annual international conference on Mobile computing and networking*, Maui, Hawaii, USA.

[CMT5] Zhong, H., & Xiao, J. (2014, 6-8 Aug. 2014). *Design for a cloud-based hybrid Android application security assessment framework.* Paper presented at the International Conference on Reliability, Maintainability and Safety (ICRMS).

[CMT6] Mahmood, R., Mirzaei, N., & Malek, S. (2014). Evodroid: Segmented evolutionary testing of android apps. Paper presented at *the Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering.*

[CMT7] Tao, D., Lin, Z., & Lu, C. (2015). Cloud platform based automated security testing system for mobile internet. *Tsinghua Science and Technology, 20*(6), 537-544.

[CMT8] Tao, C., Gao, J., & Li, B. (2015, Oct. 30 2015-Nov. 1 2015). *Cloud-Based infrastructure for mobile testing as a service.* Paper presented at the 2015 Third International Conference on Advanced Cloud and Big Data.

[CMT9] Zhang, T., Gao, J., Cheng, J., & Uehara, T. (2015). *Compatibility testing service for mobile applications.* Paper presented at the Service-Oriented System Engineering (SOSE), 2015 IEEE Symposium on.

[CMT10] Meng, Z., Jiang, Y., & Xu, C. (2015). Facilitating Reusable and Scalable Automated Testing and Analysis for Android Apps. Paper presented at the *Proceedings of the 7th Asia-Pacific Symposium on Internetware,* Wuhan, China.

[CMT11] Liu, C. H., Chen, S. L., & Chen, W. K. (2015, June 27 2015-July 2 2015). *Improving resource utilization of a cloud-based testing platform for android applications.* Paper presented at the 2015 IEEE International Conference on Mobile Services.

[CMT12] Zhang, S., & Pi, B. (2015, March 30 2015-April 3 2015). *Mobile functional test on TaaS environment.* Paper presented at the Service-Oriented System Engineering (SOSE), 2015 IEEE Symposium on.

[CMT13] Rojas, I. K. V., Meireles, S., & Dias-Neto, A. C. (2016). Cloud-Based mobile app testing framework: Architecture, implementation and execution. Paper presented at the *Proceedings of the 1st Brazilian Symposium on Systematic and Automated Software Testing,* Maringa, Parana, Brazil.

[CMT14] Yu, L., Tsai, W. T., Chen, X., Liu, L., Zhao, Y., Tang, L., & Zhao, W. (2010, 4-5 June 2010). *Testing as a service over cloud*. Paper presented at the 2010 Fifth IEEE International Symposium on Service Oriented System Engineering.

[CMT15] Gilbert, P., Chun, B.-G., Cox, L. P., & Jung, J. (2011). Vision: Automated security validation of mobile apps at app markets. Paper presented at the *Proceedings of the second international workshop on Mobile cloud computing and services,* Bethesda, Maryland, USA.

[CMT16] Baride, S., & Dutta, K. (2011). A cloud based software testing paradigm for mobile applications. *SIGSOFT Softw. Eng. Notes, 36*(3), 1-4. doi:10.1145/1968587.1968601

[CMT17] Huang, J. f., & Gong, Y. z. (2012, 3-6 Dec. 2012). *Remote mobile test system: A mobile phone cloud for application testing.* Paper presented at the Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on.

[CMT18] Starov, O., Vilkomir, S., & Kharchenko, V. (2013). *Cloud testing for mobile software systems.* Paper presented at the 8th International Joint Conference on Software Technologies, in ICSOFT K. Haller, Mobile testing, SIGSOFT Softw. Eng. Notes, 38(6) (2013) 1-8.

[CMT19] Haller, K. (2013). Mobile Testing. *SIGSOFT Softw. Eng. Notes, 38*(6), 1-8. doi:10.1145/2532780.2532813

[CMT20] Malini, A., Venkatesh, N., Sundarakantham, K., & Mercyshalinie, S. (2014, 11-13 Dec. 2014). *Mobile application testing on smart devices using MTAAS framework in cloud.* Paper presented at the International Conference on Computing and Communication Technologies.

[CMT21] Tung, Y. H., Lin, C. C., & Shan, H. L. (2014, 7-11 April 2014). *Test as a service: A framework for web security TaaS service in cloud environment.* Paper presented at the 2014 IEEE 8th International Symposium on Service Oriented System Engineering.

[CMT22] Holzmann, C., & Hutflesz, P. (2014). Multivariate Testing of Native Mobile Applications. Paper presented at the *Proceedings of the 12th International Conference on Advances in Mobile Computing and Multimedia*, Kaohsiung, Taiwan.

[CMT23] Villanes, I. K., Costa, E. A. B., & Dias-Neto, A. C. (2015, June 27 2015-July 2 2015). *Automated Mobile Testing as a Service (AM-TaaS).* Paper presented at the 2015 IEEE World Congress on Services.