Information and Software Technology 110 (2019) 56–77

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Investigation on test effort estimation of mobile applications: Systematic literature review and survey

Anureet Kaur[a,*], Kulwant Kaur[b]

[a] I.K. Gujral Punjab Technical University, Kapurthala, India
[b] School of IT, Apeejay Institute of Management Technical Campus Jalandhar, India

ARTICLE INFO

Keywords:
Mobile applications
Test effort estimation
Software engineering
Systematic literature review (SLR)
Survey

ABSTRACT

Context: In the last few years, the exigency of mobile devices has proliferated to prodigious heights. The process of developing the mobile software/application proceeds amidst testing phase to verify the correctness of the mobile app. The estimation of testing plays a vital role in the effective completion of testing.
Objective: To identify how estimation of test effort for mobile applications is distinct from other software via published literature and from mobile software organizations. Second is to recognize different issues in adapting traditional test estimation methods to the mobile domain and if suggestions from survey results could be helpful in providing an improved test estimation model for mobile applications.
Method: A systematic literature review is conducted followed by a survey through an online questionnaire filled from experienced mobile application developers and testers.
Results: The results from SLR cover identification of mobile app specific characteristics and reports test effort estimation techniques in the mobile domain. Findings from survey corroborate that a) Function Point/Test Point Analysis is highly adapted traditional test estimation technique to mobile domain; b) Challenges like uncertain requirements, no tool support for test estimation, complexity in testing, client miscommunication etc. are reported; c)Suggestions to improve test estimation process include proper test planning, adoption of agile methodology, healthier communication among client, developer, and tester etc.; d) On the basis of responses, Analytical Hierarchical Process (AHP) identifies "Diverse Devices and OS" along with "Type of App" as highly influential mobile app characteristic on the test estimation process.
Conclusion: Results conclude that the importance of identified mobile app characteristics from SLR cannot be ignored in the estimation process of mobile software testing. There might be a possibility to improve existing test estimation techniques for mobile apps by giving weight to mobile app specific characteristics and by considering suggestions from experienced developers and testers.

1. Introduction

Mobile applications are those software applications which are intended to run on the smartphone, tablet, laptop computers, and mobile devices and/or for taking input contextual information [1]. To ensure that mobile applications conform to their specifications, a usual act to improve quality is performed, called software testing [2]. If effort, time and cost required to test the software are known in advance, then one can efficiently utilize testing resources to ensure a timely and within budget completion of projects.

For developing a mobile application, traditional software development estimation methods are applied ignoring characteristics specific to mobile devices such as memory capacity, processing power, graphic interface, connectivity factor, bandwidth factor, lower battery factor, input interface factor [3]. In literature, there are techniques for estimating the effort required for software development adapted to the mobile domain; such as are FPA (Function Point Analysis), UCP (Use Case Points Analysis), SLOC (Source Lines of Codes Analysis), COCOMO (Construction Cost Model), etc. [4].

The focus of this study is on estimating the testing effort for software to be developed for mobile applications, which is currently being estimated with traditional methods of estimation used for desktop/laptop applications. According to [2], testing part of the development phase goes through an additional life cycle, so separate estimation for this phase of development needs attention. Effort estimation is one of the perilous probes in the entire Software Testing Life Cycle (STLC). STLC process progresses through different stages i.e. Requirement Analysis, Test Planning, Test Case Development, Environment Setup, Test Execu-

* Correspondence to: 60, Kabir Park, opp. Guru Nanak Dev University, Amritsar, Punjab, India.
E-mail addresses: anumahal@gmail.com (A. Kaur), kulwantkaur@apjimtc.org (K. Kaur).

https://doi.org/10.1016/j.infsof.2019.02.003
Received 15 May 2018; Received in revised form 19 February 2019; Accepted 21 February 2019
Available online 26 February 2019

Table 1
RQs and research method adopted for answering the RQs.

No.	RQs	Research method
RQ1.	What are different mobile application characteristics?	Systematic Literature Review (SLR)
RQ2.	What are different mobile applications test effort estimation techniques?	Systematic Literature Review (SLR)
RQ3.	What are different issues and challenges faced by testers for estimating the testing effort of mobile applications?	Survey
RQ4.	What are the recommended practices to overcome the challenges of estimating the testing effort of mobile applications?	Survey
RQ5.	How do mobile specific characteristics affect the testing effort estimation process?	Survey

tion, and Test Cycle Closure. There are many existing testing effort estimation techniques used for conventional software based on judgment and rules of thumb, techniques based on analogy and work breakdown, techniques based on factors and weights, techniques based on size, fuzzy and other models [5,6,15,16,7–14].

Some authors [17,18] have proposed test effort estimation models for mobile apps and [19] has adapted traditional testing effort estimation techniques to mobile software. But none have considered characteristics/factors specific to mobile applications. The research gap existing in this area needs to be addressed. The first objective of this research is to identify mobile app characteristics and its influence on testing estimation process of mobile apps. Another objective is to study the current state of the practice about testing estimation in the mobile software industry, identifying how adaption of traditional testing estimation methods to the mobile domain is affecting the testing process and real challenges confronted while estimating the mobile application testing process.

The paper is divided into eight sections. Section 2 summarizes the research method followed in the paper. Section 3 presents the process of conducting a Systematic Literature Review (SLR). Section 4 gives detail about the research approach used for conducting a survey along with major findings and challenges during the testing estimation process of mobile applications. Section 5 discusses the threats to validity in conducting SLR and survey. Section 6 presents related work followed by Section 7 with a conclusion and future work.

2. Research method

The research method aims at analyzing the available literature on characteristics specific to mobile apps, mobile app test effort estimation techniques and at involving developers and testers in the field for analyzing the impact of identified characteristics on estimation process. To achieve this aim, a systematic literature review and the industrial survey are used as a research method to answer the research questions (RQs) as shown in Table 1. After an initial analysis of existing literature, a questionnaire is developed to get better insights into current practices in industries on mobile app development and testing estimation.

The first goal of this study is to gain the real knowledge and status about the different papers that present how mobile applications testing is different from traditional (desktop/laptop) software testing estimation (RQ1). The second goal is to study existing literature on estimating the effort involved in mobile application testing (RQ2). For this purpose, a Systematic Literature Review (SLR) is conducted following the guidelines provided [20]. After completion of systematic literature review, an online survey is conducted to answer RQ3, RQ4, and RQ5. The overview of the research method can be depicted in Fig. 1.

3. Conduct of systematic literature review (SLR)

The guidelines provided by Kitchenham and Charters (2007) are followed for conducting Systematic Literature Review (SLR). SLR is a research manner for carrying out a literature review in an orderly way of charting definite phases. SLR method drives through various phases for performing literature review including specifying research questions, identification of search string and data sources, selecting studies, quality assessment, and data extraction and finally reporting the review. This literature review comprises studies reporting on characteristics of mobile applications and existing approaches on mobile application test effort estimation. SLR is conducted on 6 pertinent data sources using well-defined search criteria and retrieved mix of relevant and irrelevant publications. After performing a set of selection criteria and quality assessment, only 49 studies could be shortlisted for further analyses.

3.1. Research questions

The motive of this SLR is to identify test effort estimation in mobile applications. To get an answer from existing literature, SLR-RQ1 is formed. The second motive is to identify existing literature on test effort estimation techniques for mobile apps. SLR-RQ2 yearns to find relevant answers for this motive.

SLR-RQ1 What are different mobile application characteristics?

SLR-RQ2 What are different mobile applications test effort estimation techniques?

3.2. Search string and data sources

The guidelines provided by Kitchenham and Charters (2007) are followed to define the search string by analyzing the main keywords in RQs, synonyms of the keywords and on any other spellings of the words. The search string for RQ1 and RQ2 are shown in Table 2.

The digital databases that were used to search the keywords are SpringerLink, IEEE Xplore, ACM Digital Library, Elsevier Science Direct, Research Gate, and CiteSeer.

3.3. Study selection process

The next step is to apply the search string for SLR-RQ1 and SLR-RQ2 on all the selected electronic data sources to find the entailed studies. The results from data sources are monitored to include search string in title and abstracts. The search string is again refined each time to check the outcome and analyzed for better results. The results are restricted to peer-reviewed conference papers, books and journal papers. The duplicate titles and abstracts are removed. Then, a technique called snowball tracking is used for studying all the references of primary studies to exploit further studies and increase the chances of inclusion of important papers in the systematic literature review. This leaves us with relevant 452 studies. The resultant studies further undergo a selection process of Inclusion/Exclusion and Quality Assessment.

3.3.1. Inclusion/Exclusion criteria

The results acquired through the various studies generated with the search string defined previously in the electronic databases were analyzed according to the Inclusion/Exclusion criteria. Table 3 enlists the inclusion and exclusion criteria. The studies are restricted to include only published in the English language. The evaluation of the papers is done by reading the title and abstract first and checked if it is related to the issues addressed in RQs. Then the decision is made for its acceptance for reading the whole paper or is rejected therein. Also, only one copy of same study is included which is repeated in multiple sources and rest are excluded. The inclusion and exclusion criteria ended with 49 appropriate papers out of 452.

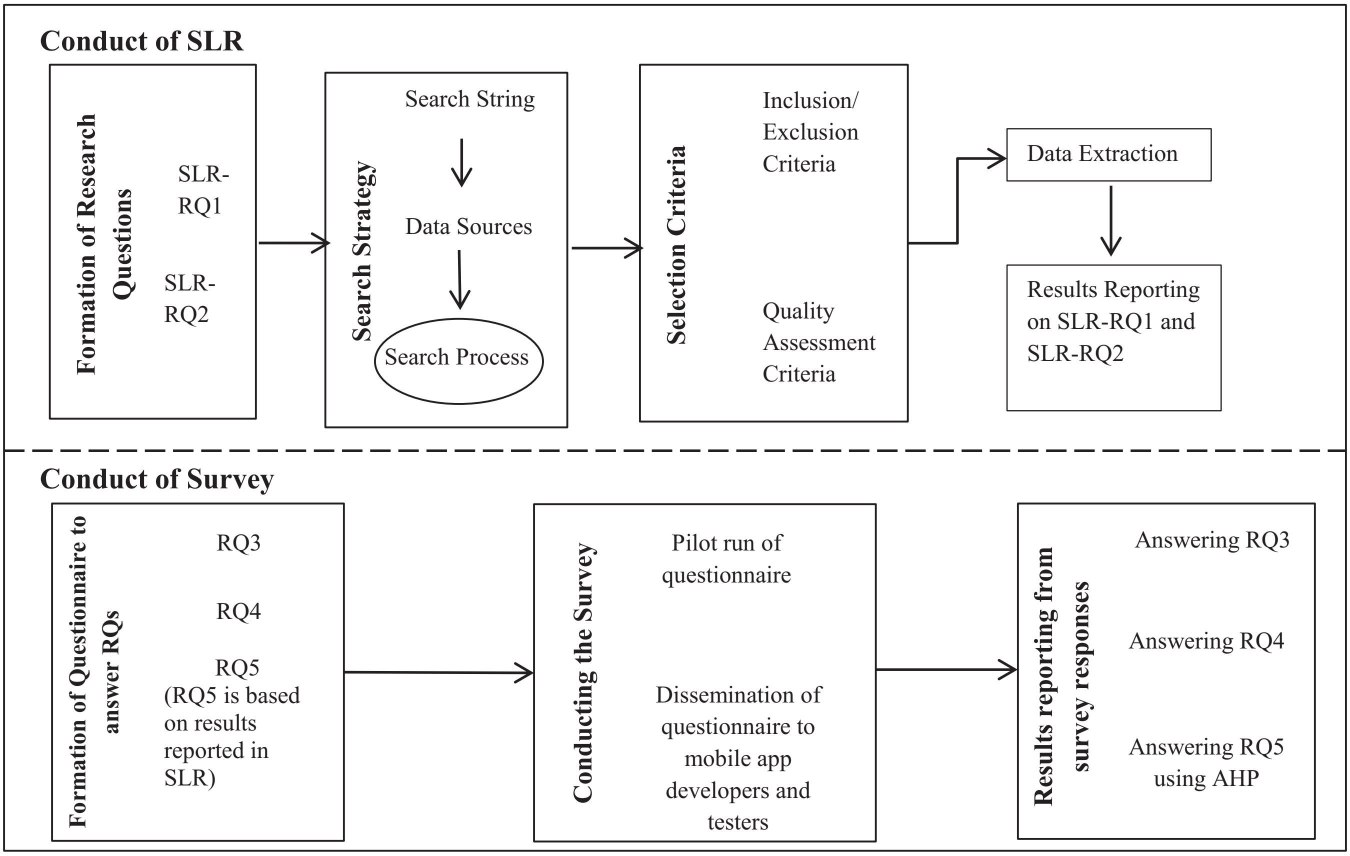

Fig. 1. Overview of research method.

Table 2
Search string for SLR-RQ1 and SLR-RQ2.

No.	Search string
SLR-RQ1.	(“Mobile Application” OR “Mobile software” OR “Mobile App” OR “Mobile project”) AND (“Characteristics” OR “Features” OR “Attribute” OR “Factors”)
SLR-RQ2.	(“Mobile Application” OR “Mobile software” OR “Mobile App” OR “Mobile project”) AND (“Test*” OR “verification” OR “validation”) AND (“Effort” OR “cost” OR “resource” OR “size” OR “metric”) AND (“estimate*” OR “predict*” OR “assessment” OR “forecast*” OR “calculate*” OR “sizing” OR “ measure*”) AND (“Process” OR “techniques” OR “models” OR “ approaches”)

Table 3
Inclusion and exclusion criteria.

Inclusion criteria	Exclusion criteria
Studies corresponding to mobile application features or characteristics	The studies not published in English
Studies focusing on test effort estimation of mobile applications	Studies corresponding to mobile devices and not mobile applications
Studies answering at least one research question	Only one copy of same study is included which is repeated in multiple sources and rest are excluded

3.3.2. *Quality assessment*

After applying the inclusion and exclusion criteria, quality assessment for each study is ensured. The quality assessment is achieved by allotting scores to selected studies. A set of five assessment questions are prepared that can be answered either with score 1(Yes), 0.5(Partial) or 0(No). The questions are listed below:

Q1. Are the research motives clearly stated?
Q2. Are the mobile app characteristics well defined?
Q3. Is the test estimation context adequately portrayed?
Q4. Are test estimation techniques for mobile apps well defined?
Q5. Is the test estimation accuracy measured and described?

The authors have performed the quality assessment of all the selected primary studies. Hereafter, 49 papers are designated to report two RQs. The scores of chosen studies are portrayed in Appendix A. 46 studies are devoted for answering SLR-RQ1 and only 3 studies to address SLR-RQ2. Table 4 shows initial results from primary and secondary search process and final selected studies after conducting Inclusion/Exclusion and quality assessment process.

3.4. *Data extraction*

The data extraction phase involves the extraction of data from the final selected studies that address the peculiarities of RQs. The following information is gathered in a data extraction form:-

- Title of selected studies
- Author

Table 4
Data sources results from search string.

Digital databases	Initial results after conducting the primary and secondary search process		Final studies after selection process	
	SLR-RQ1 Search string	SLR-RQ2 Search string	SLR-RQ1 Search string	SLR- RQ2 Search string
SpringerLink	46	20	07	01
ACM	51	31	09	None
IEEE	84	46	16	01
Science Direct	89	24	06	None
Research Gate	21	None	07	None
CiteSeer	32	08	01	01
Sub-Total	323	129	46	03
Total (before and after the final selection of studies for SLR-RQ1 and RQ2)	**452**		**49**	

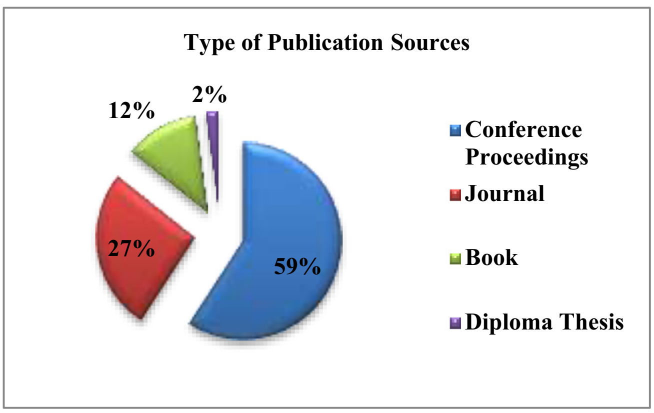

Fig. 2. Publication sources for selected studies.

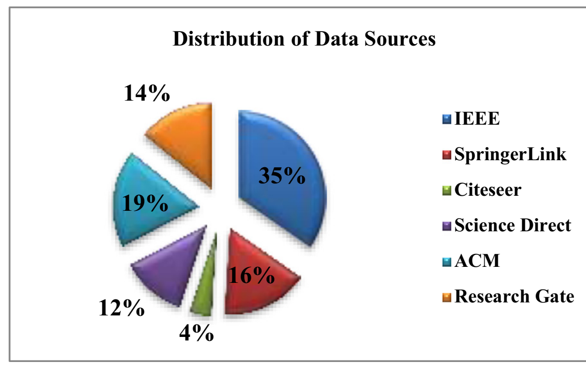

Fig. 3. Distribution of selected studies from data sources.

- Year
- Type of electronic data source
- Source of publication
- Study type
- Main findings in the study (characteristics/method/tool support)
- RQ addressed

The data extraction for the finally chosen studies are done in an MS Excel sheet and added in Appendix B in a table.

3.5. Results and discussion

3.5.1. Results overview

This section defines the results stated from the systematic literature review questions. Fig. 2 shows the distribution of the chosen studies based on the type of published sources. 29(59%) of chosen studies are from conference proceedings, 13(27%) from peer-reviewed journals, 06(12%) from books and 01 (2%) diploma thesis is included. The distribution of selected studies from different data sources is shown in Fig. 3. Out of the 49 studies, 17(35%) came from IEEE Xplore, ACM Digital Library 09(19%), 08 studies (16%) came from SpringerLink, 07(14%) studies from Research Gate, 06(12%) from Elsevier Science Direct and 02(4%) from CiteSeer. Majority of the papers are from the year 2015, 2012 and 2014. The distribution of selected studies according to the published year can be seen in Fig. 4.

3.5.2. Result reporting on SLR-RQ1-What are different mobile application characteristics?

The results from the Systematic Literature Review (SLR) recognized 14 characteristics in the majority of the chosen studies after passing all the selection criteria from primary studies. Forty-six studies are committed to answering SLR-RQ1 out of the total forty-nine selected studies.

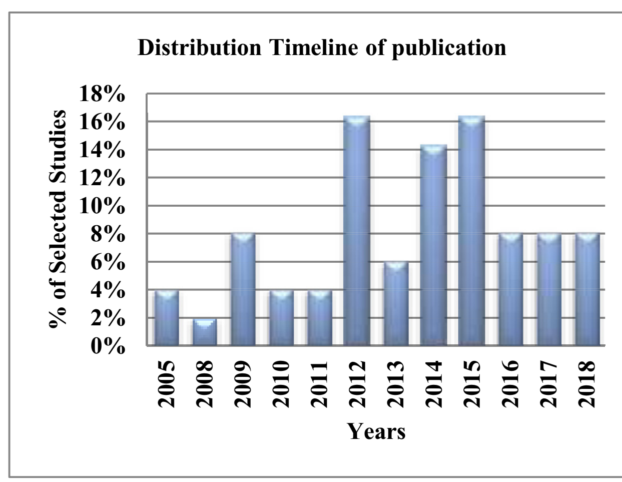

Fig. 4. Distribution of selected studies (Year-Wise).

Fig. 5 depicts a number of studies featuring these characteristics. Table 5 lists the characteristic along with a description of each characteristic

3.5.3. Result reporting on SLR-RQ2-What are different mobile applications test effort estimation techniques?

The studies identified for answering SLR-RQ2 have adapted version of models/techniques used for test estimation in other software(desktop/laptop) that are applied to the mobile domain. Though research conducted on testing in mobile applications is available, very little literature is published on the estimation of the testing in a mobile application. The insufficient work done in mobile testing estimation space became the deriving factor of this research work. The comparison of resultant studies for answering SLR-RQ2 is presented in Table 6.

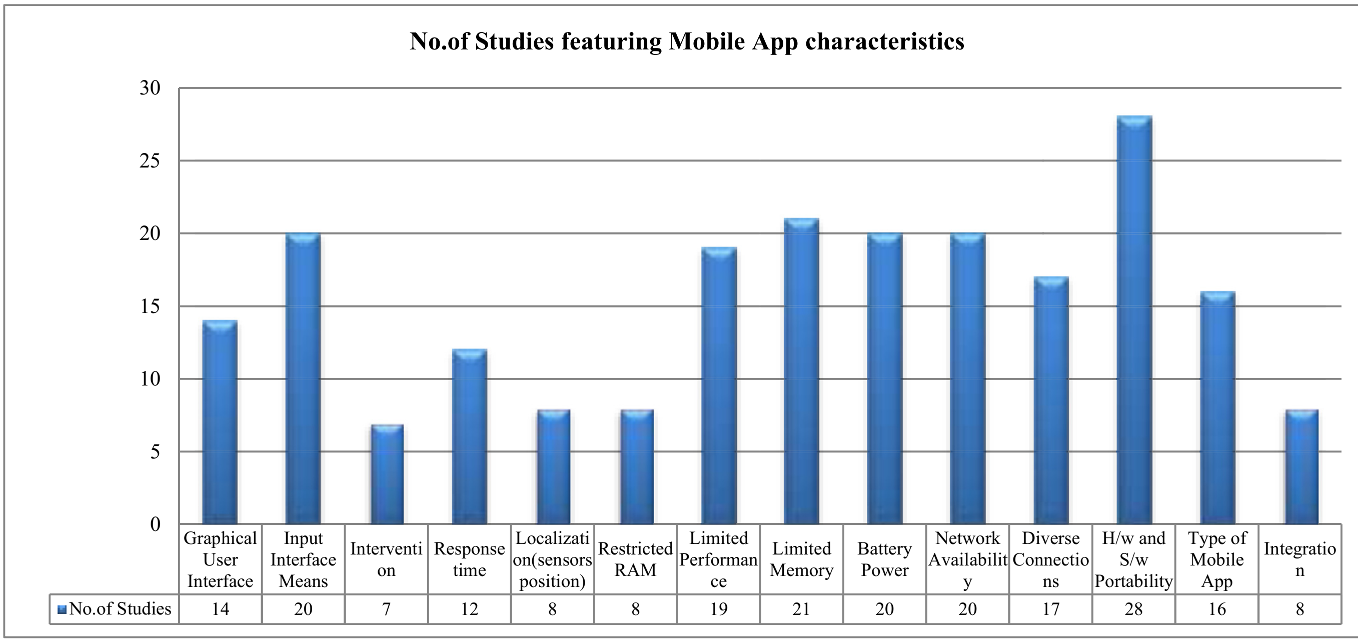

Fig. 5. Mobile app characteristics identified in number of studies.

Table 5
Identified mobile apps characteristics from literature review.

Category	Sub-category	References	Description of mobile app characteristic
Usability	Graphical user interface	[21,22,31–34,23–30]	Mobile devices have a small screen. Keeping the constraint in mind the app should be developed and tested well to check if it operates differently on varied screen size and orientation.
	Input interface means	[2,21,40–49,22,28,32,35–39]	The input to a mobile device can be through voice, touch, keypad, stylus, etc. The mobile app should be tested against varied means of input interfaces.
	Intervention	[25,31,34,39,50–52]	Interruptions such as receiving a message, battery low, in between calls occur when the mobile app is in use. The app should be tested for its behavior during these interruptions.
	Response time	[23,25,56,57,32,34,37,38,47,53–55]	The mobile app should take minimum time to respond for any means of input. The time it takes for opening the app should be tested to increase its usability.
	Localization (sensors position)	[2,21,25,29,32,34,39,45]	Mobile apps can have sensors that respond when the mobile device moves such features make good user experience. It should be tested for such sensors appropriately.
Efficiency	Restricted RAM	[2,25,34,35,37,53,57,58]	Apps should be programmed and tested so that they exhaust less amount of RAM when they run on the mobile device. Large size mobile apps tend to run slow and further influence user experience
	Limited performance	[2,25,48,50,53,54,56–60,30,34,35,37,41,44,45,47]	With limited capacity of processors; the mobile apps should be tested for its performance in conjunction with limited memory.
	Limited memory	[2,23,47–50,53–58,30,60,34,35,37,38,42,43,45]	The internal memory of the mobile device is limited. The mobile app consumes a memory space when it is installed on the device. The developers should use such programming practices that allow development of small size apps. The testers should check how the app performs when the memory of the device reaches maximum memory limit
	Battery power	[2,25,45,49,50,53,54,56–58,60,61,30,32,34,35,37,41–43]	Mobile devices have very limited battery life. The mobile apps should be developed in a way so they should consume less battery power. The app should be tested in a scenario when the battery is too low that how it behaves in this instance and should retain data integrity when the battery dies.
Network	Network availability	[21,25,49,54–57,59,60,62–64,26,29,32,34,36,38,41,43]	Network availability varies, so apps should be developed and tested keeping this constraint in mind. It should be tested how it behaves when the user moves to the remote area when networks are not in range.
	Diverse connections	[21,25,55–57,59,62,63,65,31,36,38,43,49–51,54]	App should be tested on different connections such as Wi-Fi, Bluetooth, 3 G, 4 G, NFC, etc., so that it should be keeping up with all connections.
Diverse Devices	H/w and S/w portability	[21,23,35,38,40–45,49,51,24,52,55–58,62,64,65,25,27–29,31–33]	The app should be tested on maximum possible mobile devices so that it should not behave differently on varied devices. The apps are developed to run on multiple OS such as iOS, Windows, Android, Symbian etc. and should be properly tested for functionality on the varied OS.
	Type of mobile app	[23,25,50,53–56,65,27,34,37,38,40,43,45,49]	Consideration on whether the app will run on the device only or on the web also, or is it a native or hybrid app? The development and testing of native, web and hybrid mobile application are different. So each one should be tested thoroughly depending on the type of app.
Integration with other apps		[32,34,45,50,52,54,56,65]	There are some apps that run in integration with other apps. Testing should be done to check if mobile app integrates well with other apps on the user's device or not

Table 6
Comparison of identified mobile test effort estimation techniques.

Method	Input to model	Estimation attributes	Effort measured in	Accuracy measure	Achieved accuracy	Tool support
Architecture-based model [18]	Functional and quality attributes as model based specifications and formal specifications	Time and cost	Not defined	Not defined	N.A.	No
Based on the size and execution complexity of the test cases [66]	Test specifications	Size, execution complexity of tests, team experience, software stability, test environment conditions, time pressure	Person-Hour	MRE, MMRE, Pred(x)	MMRE = 58.20% and Pred (0.25) = 51.97%	Yes
Modified use case point model [19]	Different user scenarios	Size, efficiency and risk factor	Person-Hour	Comparison of real efforts and estimated efforts	2% deviation from actual test effort	No

An estimation model for the test execution effort based on the test specifications was proposed by [66]. For that, the authors define and validate a measure of size and execution complexity of test cases. This measure was obtained from test specifications written in the controlled natural language. Then they evaluated the model through an empirical study on the mobile application domain, whose results suggested an accuracy improvement when compared with estimation based only on historical test productivity.

A framework proposed by [18] for estimation of testing and reliability in mobile applications is based on the architecture model. The team conducted a case study with two organizations to find the pros and cons of conventional testing estimation techniques. The findings from the survey helped them in proposing the new framework. But the applicability of the proposed method is not further explored in their paper.

A use case point estimation method for test estimation in an agile approach to software development is proposed in [19]. The authors have added a new layer of the efficiency of the testing team and risk factor in existing use case model. A case study was performed on four real projects, out of which two were mobile applications. The results of the proposed model presented better estimates for test effort as compared to unmodified use case point method.

3.5.4. Discussion

The results from SLR for answering research question SLR-RQ1 reported fourteen characteristics of mobile apps. Testing of the mobile app on “Different mobile OS and devices” as of concerning H/w and S/w portability is identified in most studies. The second most reported characteristic is “different mobile connections” and “Limited memory”. Rest of the characteristics is discussed in Table 5.

As for estimation on testing of the mobile app, these identified characteristics may or may not affect the test estimation process. The impact of each characteristic while performing test estimation can range from being negligible to highly significant. A survey on investigating the impact of mobile app characteristics, accumulated from mobile app developers and testers can be beneficial to accomplish this task. The findings for SLR-RQ1 clearly state how these mobile app characteristics are different from traditional software. But test estimation techniques reported in SLR-RQ2 does not consider these important characteristics undertaking the estimation process.

SLR-RQ2 results in reporting only few test estimation techniques of mobile apps. The identified studies present techniques/models for test effort estimation in the mobile domain. None of the studies consider mobile app characteristics for estimation. This can serve as a future direction when investigated in the mobile domain for proposing a standardized model and validate the estimation results of the model on mobile apps. For measuring the accuracy of identified estimation models in mobile app domain, statistical measure MMRE and Pred(x) are used. Also, the estimated results are compared with actual test effort to measure the accuracy of the model. The tool support for test effort estimation is reported in only one study. Two identified studies out of three states the measurement of test effort in terms of Person-Hour.

4. Conduct of survey

Section 3 provided the necessary background mainly 14 mobile app characteristics and test effort estimation techniques which are required for conducting the survey. The goal of the survey is to identify the problems encountered by testers while incorporating prevalent test effort estimation techniques for mobile apps and their suggestions for improvising the estimation process. Also, the identified mobile app characteristics in Section 3 are examined from the software industry perspective. To achieve this goal following RQs are formed:

RQ3. What are different issues and challenges faced by testers for estimating the testing effort of mobile applications?
RQ4. What are the recommended practices to overcome the challenges of estimating the testing effort of mobile applications?
RQ5. How do mobile specific characteristics affect the testing effort estimation process?

4.1. Designing and conducting the survey

The design of a questionnaire for conducting the online survey was based on existing literature and consulting experts in mobile application development and testing area. The questionnaire was first given a pilot run before it was sent to the participants. The questionnaire was forwarded as a Google form link “https://goo.gl/forms/jP1PlkEs2MR9OH9x2” via email to more than 300 developers and testers for volunteer participation in the survey. For selecting the emails; social sites like LinkedIn, Facebook, etc. are explored for those working in the software industry with experience. This task seems to be time-consuming as it was hard to convince friends on social sites and also to get e-mail IDs of their friends working in the mobile software domain. Secondly, respondents considered filling the questionnaire as a “waste of time” task in their hectic schedule. So repeatedly reminders were sent through emails and frequent calls to friends were made to get the work done. The survey questions were categorized into four parts. Questions related to the profile of respondents and their organization, Mobile Application Testing Estimation Questions, Mobile Application Development & Testing and finally Mobile Application Characteristics influence on test estimation. The questionnaire is added in Appendix C. A total of 87+ responses were received from mobile application development and testing domain.

Table 7
Numerical scale [67].

Definition	Intensity of importance
Extremely important	98
Very Strongly more important	76
Strongly more important	54
Moderately more important	32
Equally important	1

Table 8
RI of judgment matrix from 2 to 15 given by [72].

Matrix degree	RI	Matrix degree	RI
2	0.00	9	1.48
3	0.58	10	1.49
4	0.90	11	1.51
5	1.12	12	1.48
6	1.24	13	1.56
7	1.32	14	1.58
8	1.41	15	1.59

4.2. Introduction to AHP

In this paper, the Analytic Hierarchy Process (AHP) is used for deriving weights for each identified mobile app characteristic based on the influence rating collected from various respondents. AHP was introduced by [67] that helps in the decision-making process by deriving weighting scales using the pairwise comparisons and relies on expert judgments. The choice of AHP over other Multiple-Criteria Decision Analysis (MCDM) methods [68] is due to its simplicity, transparency, and capacity to integrate a large amount of heterogeneous data. There are many applications of AHP for deriving weights [69–71]. AHP includes consistency checking of the decision maker's evaluations which help in reducing the bias in the decision making process as proposed and validated by various authors [67,72–75]. The method followed in AHP is shown in the following steps:-

(1) Formation of hierarchy model: – A decision-making problem is broken down into a series of factors (Criteria) that builds the hierarchy model. At the top of the hierarchy is objective and on the bottom is a decision to be taken (i.e. alternatives).
(2) Formation of judgment matrix: – Elements in Judgment matrix are a relative priority of each factor with respect to the other using a numerical scale developed by [67] as shown in Table 7.
(3) Checking for consistency: – There may be inconsistency in the judgment matrix. Consistency ratio (CR) is used to check every judgment matrix for consistency. The consistency ratio (CR) should not exceed 0.1, else the matrix is not considered to have acceptable consistency [72]. To calculate the Consistency Ratio (CR) formula is:

$$CR = CI/RI \tag{1}$$

where Consistency Index (CI) is calculated as:

$$CI = (\lambda \max - n)/n - 1 \tag{2}$$

In the above Eq. (2), λ_{max} or principle eigenvalue of a matrix is calculated as follows; first, a consistency matrix is calculated by multiplying pair-wise matrix by weight matrix. The weighted sum matrix is then divided with criterion weight. Finally, λ_{max} is calculated by averaging the value of consistency matrix [72,73]. 'n' is the number of compared factors. Random index (RI) is dependent on the matrix degree. Table 8 shows RI for matrix degree <=15.

The introduction to AHP was necessary to provide background knowledge for analyzing RQ5 results.

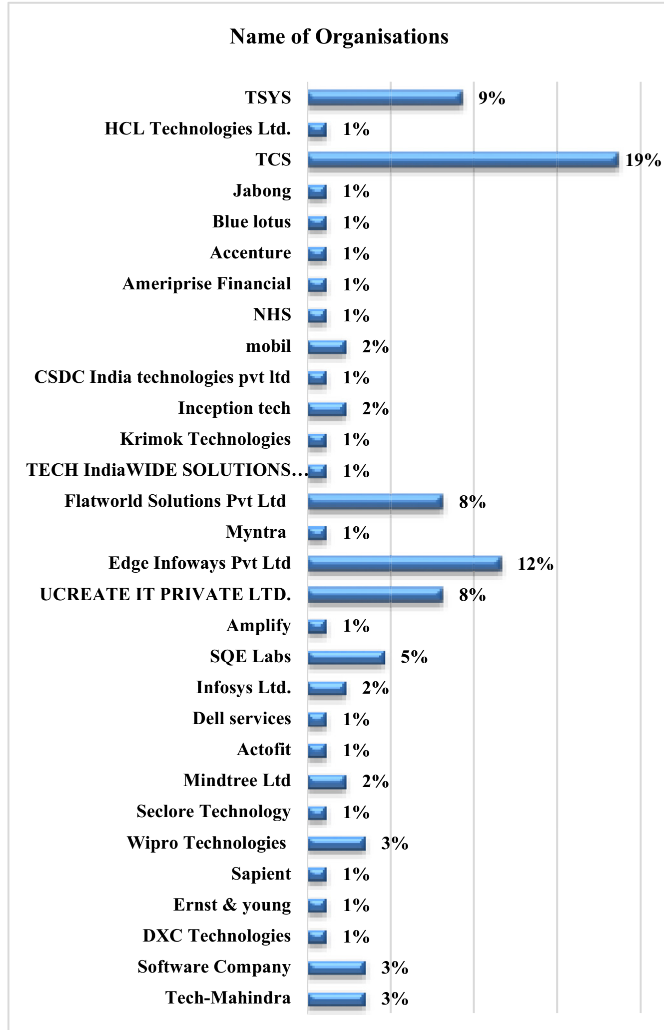

Fig. 6. Percentage of participants from different organizations.

4.3. Results from survey

4.3.1. Participants overview

Figs. 6–10 show the participants basic nature and their organization profile. All the values in figures are shown in percentage. Maximum Responses were received from TCS organization covering (19%) and (12%) participants from Edge Info ways Pvt. Ltd. It can be noted that (91%) participants are from India and (4%) is from Canada and rest are from the USA, UK, Russia, and Australia. The maximum response (37%) is from organizations with more than 5000 employees. As the participants are from the software engineering field and our target was to collect a questionnaire from only testers, but some organizations have no separate team for testers. Members from the development team are involved in the testing phase as well for mobile apps. The maximum response is from participants who are either mobile app developer (23%) or tester with (17%) participation. Rests of the other participants are with different roles related to testing only in their organizations. Out of all the participants, 62% exclusively deal with estimation of testing for mobile applications.

4.3.2. Responses regarding mobile application development and testing estimation

This subsection presents general information on mobile application development and testing estimation prevalent in software industries.

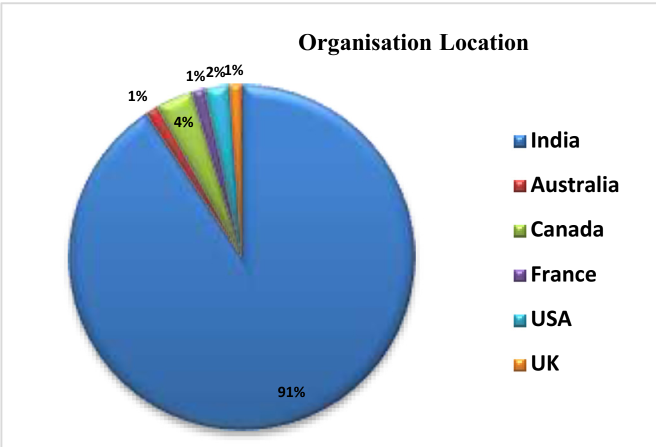

Fig. 7. Percentage of participants from different countries.

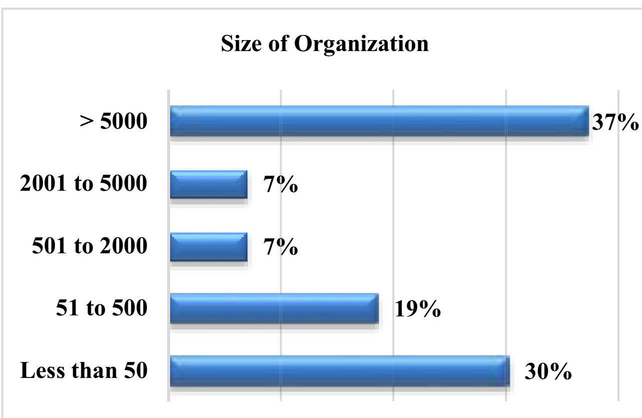

Fig. 8. Percentage of the participants in a survey by the size of the organization.

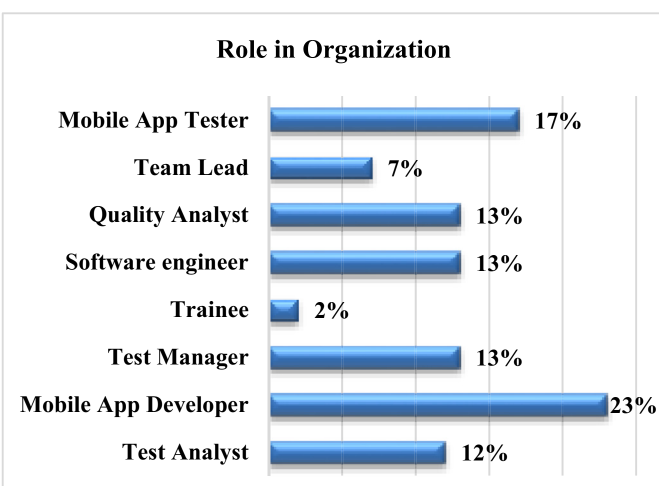

Fig. 9. Different role of participants.

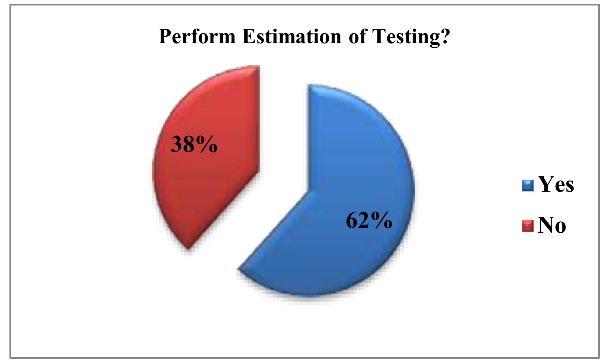

Fig. 10. Percentage of respondents engaged in the testing estimation.

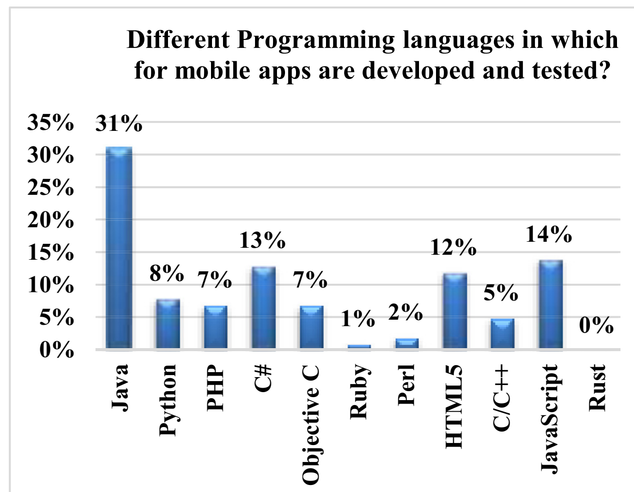

Fig. 11. Preferred Programming languages for mobile app development and testing.

Figs. 11–20 summarize the responses based on a questionnaire and are reported as follows:

- Mobile apps are developed in different programming languages. Most of the respondents have experience in developing and testing mobile apps in Java (31%) as shown in Fig. 11.
- There are many platforms on which mobile apps can run. To ratify that the issues recognized were not explicit to a particular platform only, the survey was extended to users of all platforms. Fig. 12 presents the categorization of participants by platform. As can be seen from Fig. 12 that Android (45%) and iOS (36%) are dominating mobile operating system for which mobile apps are developed and tested.
- The majority of the respondents have experience in developing and testing mobiles apps related to Business (12%) and Health Care (11%) as presented in Fig. 13.
- Foremost focus of respondents was on covering Unit testing (27%) and User Acceptance Testing (26%) followed by Integration (24%) and System testing (23%) for mobile applications as seen in Fig. 14.
- The major emphasis of respondents is on functional testing (17%) and performance testing (14%) Usability Testing (13%) and rest of testing include compatibility (12%), regression (12%), and security testing (12%). Fig. 15 shows the respondents prominence on the type of testing performed on mobile apps.
- Fig. 16 shows that highest responses were received for testing the mobile app on Real Devices (51%), on Emulator (35%) and only (14%) perform testing on Cloud.
- From Fig. 17, it can be analyzed that (64%) of participants find the estimation of testing for mobile applications different from other software.
- The most important observation from the survey is the identification of existing testing estimation methodology used for conventional software is primarily being adapted to the mobile application domain. From Fig. 18, it can be clearly analyzed that organizations are most comfortable in working with Function Point/Test Point

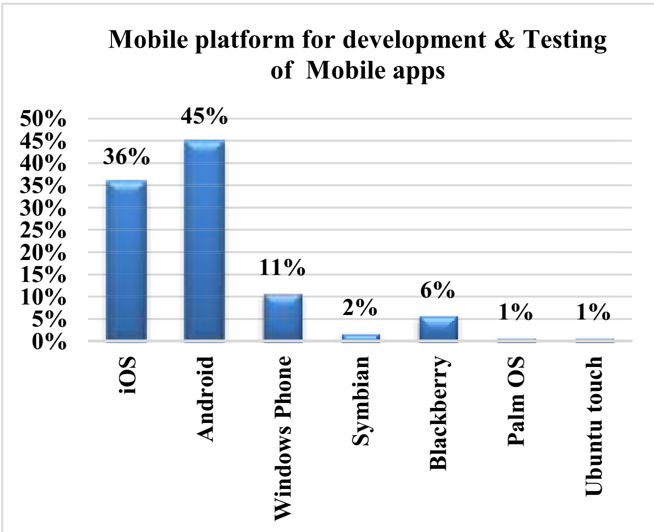

Fig. 12. Preferred mobile platform for app development & testing.

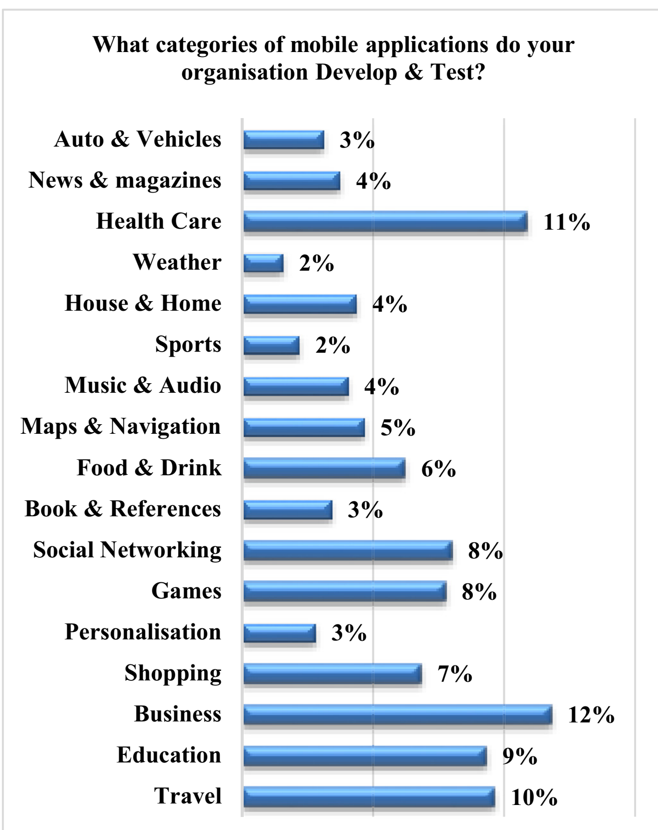

Fig. 13. Categories of mobile apps developed & tested.

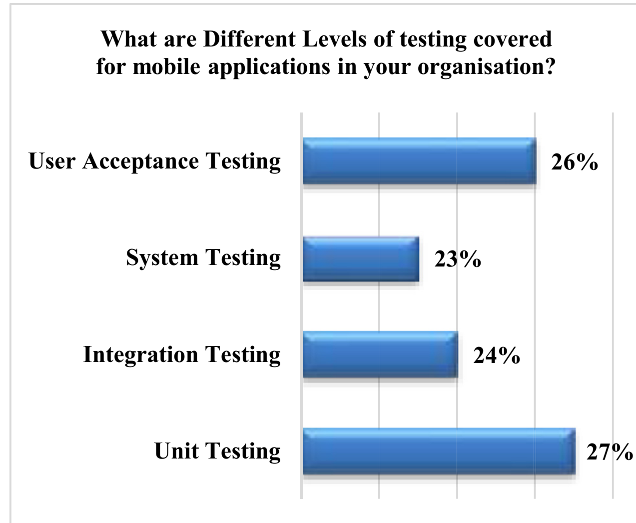

Fig. 14. Different levels of testing covered for mobile apps.

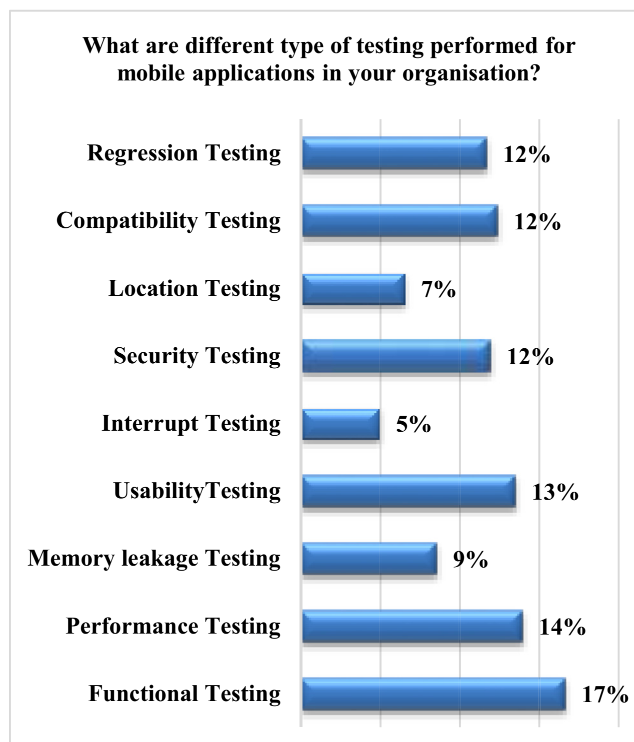

Fig. 15. Different type of testing done for mobile apps.

Analysis method (34%). Work Breakdown Structure (20%) is the second choice and expert judgment (14%) and use case point (13%) being third and fourth respectively. This analysis will further help in improving the most prevalent current testing estimation methodology by incorporating mobile app specific characteristics.

- (45%) respondents consider current estimation method followed in their organization is 21–50% incorrect in providing estimations when compared to the actual effort. (44%) respondents believe estimations are accurate from 51% to 80%. (10%) think they are 20% or less accurate and (1%) think they are not accurate at all. None of the respondents considers that the current estimation method provides 100% accuracy (Fig. 19).
- Fig. 20 depicts that (90%) respondents do not use any tool for the testing estimation of mobile apps. The remaining (10%) are using tools. Out of these (10%) remaining, (4%) used Appium [76] an automation tool for running scripts and testing native applications and mobile-web applications on Android or iOS using a web driver. We are afraid that the respondent had misjudged the question and answered for a question like if they are using any tool for mobile app testing. Same can be said for remaining (6%) as they have named crash analytics which is a crash reporting tool used for testing purpose only. Tableau [77], another reported tool is otherwise a Business Intelligence Tool used for data visualization. Another answer

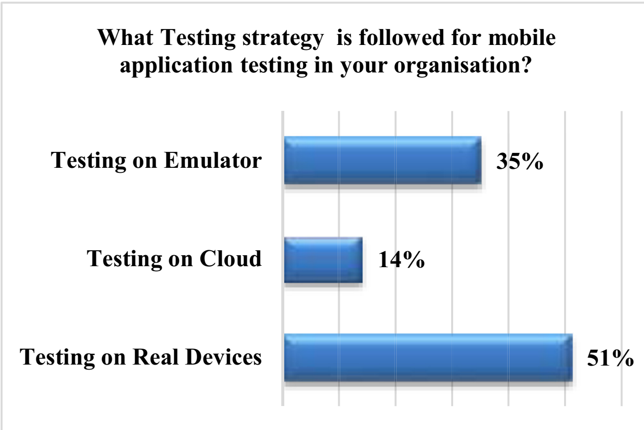

Fig. 16. The testing strategy followed for mobile apps.

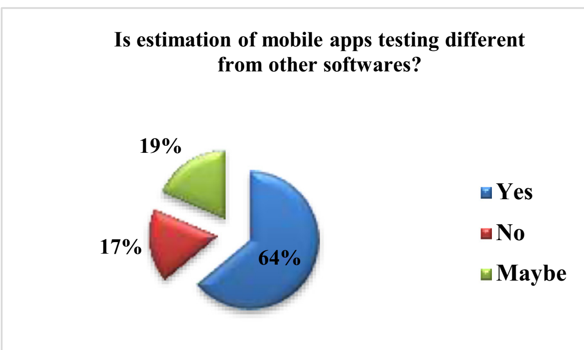

Fig. 17. The testing strategy followed for mobile apps.

related to the testing estimation tool for a mobile app was Quality Center/JIRA. According to [78], Quality Center is a quality management/test management tool and JIRA is full featured test management, ability to create, plan and execute tests and Quality metrics tracking tool. None of them were test estimation tools. We were so much interested in getting an answer to this question, as there was no tool available in prevalent websites or literature for test estimation of mobile apps. Though there were many tools available for mobile application development estimation.

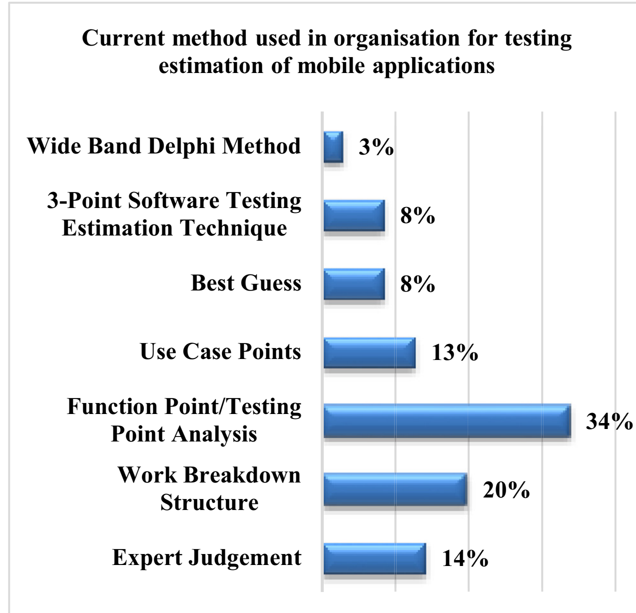

Fig. 18. The current method used in organization for testing estimation of mobile applications.

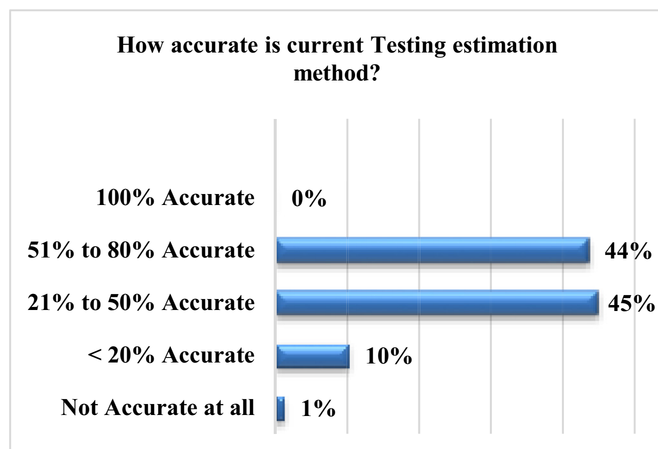

Fig. 19. How accurate is the current testing estimation method?.

4.3.3. *Answering RQ3-What are different issues and challenges faced by testers for estimating the testing effort of mobile applications?*

This subsection presents the survey study results showing most protruding challenges confronted by mobile app testers for estimation in various organizations. The challenges which were testified by respondents are described as follows:

Uncertain requirements: – Respondents believe that Requirements & scope of App keeps on changing during development and testing. One issue reported by a respondent that testing estimation is sporadically underestimated or overestimated or sometimes it is as per estimate. Reason for more than the estimated time is the app is not developed correctly as it is mentioned in Product Requirement Document (PRD). Sometimes the requirement of Clients is misunderstood and there are Unrealistic deadlines. The estimation goes off the beam due to dependencies or blockers.

No tool support: – There is no option available for choosing automation of testing estimation.

Low-quality App development: – The Testing process becomes tedious due to the low-quality development of the app.

Uncertain risks: – There are always some uncovered risks and scenarios involved. Another respondent reported regarding the application instability due to validation of test data.

Client miscommunication: – Sometimes there is a miscommunication with a client which led to improper estimation time. There is no accounting for software glitches and improper client communication.

Complexity in testing: – The major issue according to three respondents was testing for different devices on a different platform. Another major issue was that sometimes testing itself fails at a certain level. One reported concern was, either the testing has not been done properly or estimate was incorrect. There may exist incompatible API's version or unhandled exceptions.

Inexperienced staff: – One respondent conveyed that there is a lack of real knowledge for testing estimation.

4.3.4. *Answering RQ4-What are the recommended practices to overcome the challenges of estimating the testing effort of mobile applications?*

There were some suggestions to improve the testing estimation process. The suggestions which were recommended by many respondents are listed below:

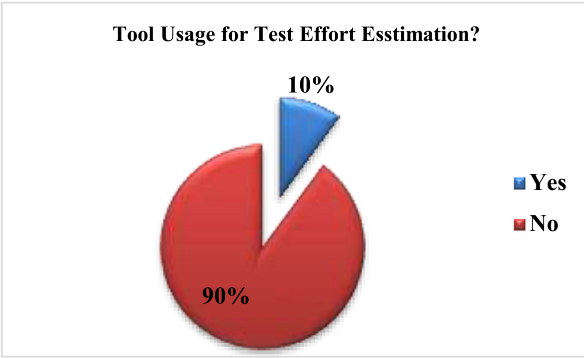

Fig. 20. Usage of a tool for test estimation for mobile apps?.

Quality of development: – Proper build should be provided to the tester.

Freezing of requirements: - Many respondents stated that App should be defined as per Product Requirement Document (PRD) and should be clear. The requirements should freeze and not be changed during development and testing.

Healthier communication: – Communication between client and developers and also between developer and tester should be healthy. Development and testing team can sit together to discuss the areas for regression testing beforehand so that testers won't have to regress the entire App. Another respondent said that there should be healthier communication with clients and use some software for performing estimation.

Tool support for test estimation: – There should be an automated tool for estimating the testing process. One suggestion was to use a Bug tool.

Proper test planning: – The Planning for testing of apps should be done properly. Planning of entire module testing like time, cost, persons, and high configuration devices with good RAM and processors is required. Testing Time should be properly estimated to ensure on-time deliveries.

Agile methodology: – The agile methodology should be adopted for development and the entire testing process.

Different methods of test estimation: – Testers should gain a thorough knowledge of each module and then note down high-level scenarios, and finally estimate the number of test cases for the same. One contributor to the survey suggested estimating based on the number of cases multiplied by environments. The Expert Judgment and quality of development plays a vital role. The team should keep a historical record and should use it for analysis. Also, Expert Feedback should be taken every time.

Preparation to handle inherent risks: - The developer and tester should be aware of the inherent risks. The test team should be prepared beforehand in case of any medical emergency of its members. Immediate blocker issues should be fixed if any occur during testing.

4.3.5. Answering RQ5-How do mobile specific characteristics affect the testing effort estimation process?

The extent of influence of mobile application characteristics on the testing estimation is collected in the range from 'No effect' to 'Very High Effect'. The results obtained from respondents will be helpful in estimating to what extent particular characteristic may affect the testing estimation process. Then these findings can be used for altering the existing conventional testing estimation methods for mobile software. The percentage influence according to responses for each characteristic is further used for deriving weights using the Analytical Hierarchical Process (AHP). The introduction to the AHP method is presented in Section 4.2. The application of AHP is shown below:-

(1) Formation of hierarchy model: – Fig. 21 shows the hierarchy model for mobile app testing estimation factors whose weight assignment is the objective and App1, App2 are alternatives for which weights of different factors are to be considered.

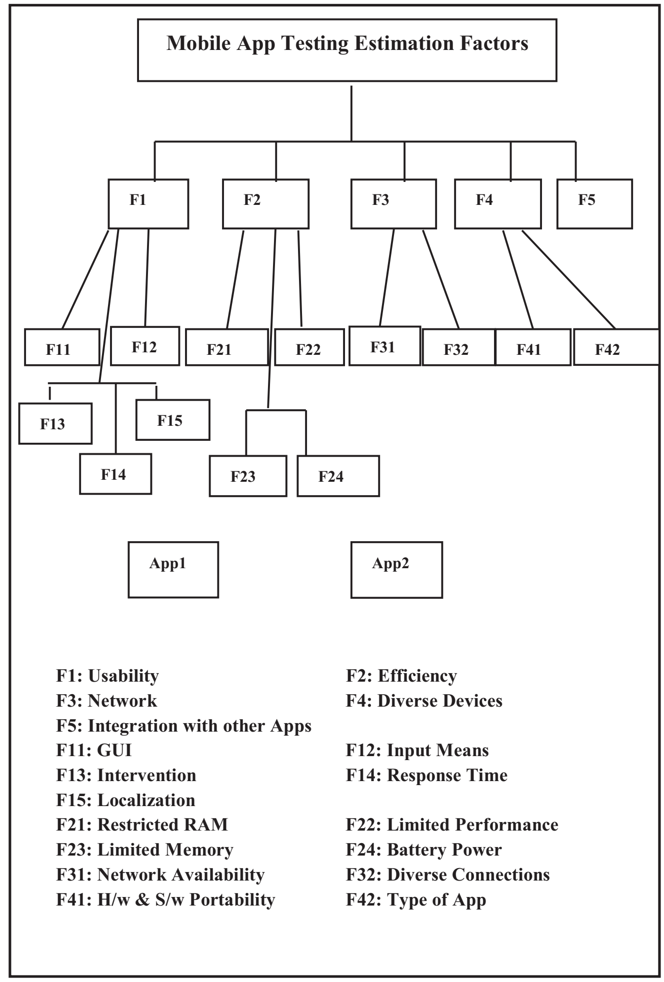

Fig. 21. Hierarchy model.

Table 9
Judgment matrix of factors in 3rd level to F1 in 2nd level.

F1	F11	F12	F13	F14	F15	Weights	Importance
F11	1	2	5	2	7	0.39	39%
F12	0.5	1	4	0.5	6	0.21	21%
F13	0.2	0.25	1	0.2	3	0.07	7%
F14	0.5	2	5	1	7	0.29	29%
F15	0.14	0.16	0.33	0.14	1	0.04	4%
Λmax = 5.150, CI = 0.037, CR = 0.033 < 0.01(Consistent)							

(2) Formation of judgment matrix: – The judgment matrix can be constructed based on factors (F1–F5) by pairwise comparison of those sub-factors immediately below it (e.g. for F1 Factor the sub-factors are F11, F12, F13, F14, F15).

The Tables from 9 to 14 show the judgment matrix for the 3rd level to 2nd level and 2nd level to 1st level. The weight column in each table is calculated in 2 steps. First, each value in the matrix is divided by the sum of its column to generate a normalized comparison matrix. Secondly, the mean of each row is calculated. This gives the weight value for each row and is normalized as their sum is 1. The importance of each factor is shown in the last column in the Tables from 9 to 14.

(3) Checking for Consistency: – λ_{max} for every judgment matrix is shown in the last row of Tables 9–14 along with values of CI and CR. The result of applying AHP is the level of importance and weight for each factor as shown in Tables from 9 to 14.

Table 10
Judgment matrix of factors in 3rd level to F2 in 2nd level.

F2	F21	F22	F23	F24	Weights	Importance
F21	1	0.2	0.2	0.5	0.07	7%
F22	5	1	0.5	4	0.33	33%
F23	5	2	1	5	0.49	49%
F24	2	0.25	0.2	1	0.11	11%
Λmax = 4.060, CI = 0.020, CR = 0.022 < 0.01 (Consistent)						

Table 11
Judgment matrix of factors in 3rd level to F3 in 2nd level.

F3	F31	F32	Weights	Importance
F31	1	6	0.86	86%
F32	0.16	1	0.14	14%
Λmax = 1.979, CI = −0.020, CR = 0 < = 0.01 (Consistent)				

Table 12
Judgment matrix of factors in 3rd level to F4 in 2nd level.

F4	F41	F42	Weights	Importance
F41	1	0.5	0.33	33%
F42	2	1	0.67	67%
Λmax = 2, CI = 0, CR = 0 < = 0.01 (Consistent)				

Table 13
Judgment matrix of factors in 3rd level to F5 in 2nd level.

F5	F5	Weights	Importance
F5	1	1	100%
Λmax = 1, CI = 0, CR = 0 < = 0.01 (Consistent)			

Table 14
Judgment matrix of factors in 2nd level to 1st Level.

	F1	F2	F3	F4	F5	Weights	Importance
F1	1	2	0.5	0.33	3	0.16	16%
F2	0.5	1	0.33	0.25	2	0.10	10%
F3	2	3	1	0.5	4	0.26	26%
F4	3	4	2	1	5	0.42	42%
F5	0.33	0.5	0.25	0.2	1	0.06	6%
Λmax = 5.064, CI = 0.0160, CR = 0.0143 < 0.01 (Consistent)							

4.3.6. Discussion

The survey assists in addressing real challenges and suggestions from mobile app developers and testers. The objective of the survey study was to identify how applicability of traditional testing estimation methods to the mobile domain is affecting the testing process. Also, the characteristic identified in SLR is further investigated in the industry through a survey. From the survey, the challenges and suggestions reported in Sections 4.3.3 and 4.3.4 seem to be common partially for desktop/laptop and for mobile apps too. The motive of the survey was to identify whether the literature findings of SLR-RQ1 is also reported from an industrial perspective as a suggestion or a challenge. Also, one challenge reported by a respondent that there is a lack of real knowledge for test estimation in case of mobile software. So from this challenge, it can be concluded that the origin of the disparity between estimation and actual effort is unclear to testers due to lack of knowledge. From RQ4, which forms its basis from answers of SLR-RQ1, we have tried to extract the influence of mobile app specific characteristics on test estimation. The findings of the survey study show that there is a need for a dedicated testing estimation method that considers mobile app specific characteristics. For this, the influence of each characteristic on testing estimation is congregated from the experts. Analytical Hierarchical Process (AHP) is then used for assigning weights to identified mobile app characteristics. It is expected that the estimation results will improve after considering the weight of each factor in test estimation of mobile software.

5. Threats to validity

In this paper threats that can affect the validity of the results are discussed. The threats related to conducting of SLR and conduct of the survey are presented in this section.

5.1. Systematic literature review (SLR)

The main validity threat in SLR is an incomplete set of final studies for further investigation. To mitigate this threat a systematic method is followed as proposed by [20]. The selection of search strings is based on the research questions and applied to digital data sources. The resultant studies from data sources were further refined by using inclusion/exclusion criteria and quality assessment. The studies reported answering SLR-RQ2 was very less in number. The search results specifically lack to cover 'test effort estimation', otherwise mere 'effort estimation' in mobile application resulted in numerous publications. So, following the exclusion criteria, only three studies passed for final selection.

The mobile application characteristics reported in results tried to conclude maximum features from selected studies but there may be others which can be further investigated as the list is not exhaustive. The authors tried to summarize the findings of SLR from different aspects of estimation techniques but still, it might miss the in-depth analysis of the results.

5.2. Survey

The first threat is in constructing the questions for the questionnaire. This is concerned with how well the questions address the problem at hand. Authors have designed the survey questions based on their experience and from a literature review in mobile application development and testing. To increase the proficiency of addressing the RQs in the study, the authors also consulted three experts in the field of mobile application development and testing. After designing the questionnaire, the feedback was collected from colleagues and pilot run was conducted to check if the participants understand the questions easily.

Second threat concerns with data collection and extraction. In order to avoid errors while gathering data in the questionnaire, the Google form feature was used to extract the data from.CSV files and import it into the Excel sheets that were used for statistical analysis. So, chances of feeding human errors by typing can be minimal, but can't be said anything about those recorded by respondents themselves.

The third threat is regarding questions asked in the online survey can be misinterpreted by respondents, which may further influence the results. To alleviate this threat a discussion among authors and then with experts in the field was performed. Some suggestions and improvements recommended by experts were later instigated to lessen the chances of misinterpretation.

The fourth threat to the survey study is a low response rate. Many participants were not willing to disclose organization detail due to company policy and were difficult to convince that they are not being asked for any private organization details that could harm them in any way. So, the responses presented are not sufficient to generalize the results. But, they cannot be ignored, as they were from professional testers and developers.

6. Related work

According to the best of our insight, there is no systematic literature review (SLR) performed on test effort estimation in the area of mobile applications. Likewise, there is no survey performed that specifically focuses on mobile application test estimation. However, literature review and survey publications exist that focus on how mobile applications are different from other software. But unlike approach followed in this paper, others have not done it systematically (not following SLR approach).

7. Conclusion and future remarks

The paper presented a Systematic Literature Review (SLR) and an online survey conducted with industry professionals. It aims to find answers to five research questions on mobile app test effort estimation. Two RQs are answered through SLR focusing on differing mobile apps from other software and reporting techniques for test effort estimation on mobile apps. The results from SLR presented 14 mobile app characteristics and their implication on testing and only three studies are reported addressing the test estimation techniques. None of the three reported studies consider mobile app characteristics in test estimation which further opens future research avenues. The other three RQs were answered by means of a survey. The survey questions were concerned with the issues and challenges faced by developers and testers for the mobile application testing estimation process. The influence of mobile app characteristics identified in SLR is reflected from an industrial perspective. The challenges presented in the paper can be counted as probable research topics in mobile application testing estimation process. It cannot be claimed that the list of challenges and issues presented are exhaustive, but they can be considered as a reference point by practitioners and stakeholders for extending their prospects. For future work, new improved model considering these characteristics can be proposed and validated on some mobile applications of an organization.

Ethics approval and consent to participate

All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards. Informed consent was obtained from all individual participants included in the study.

Acknowledgment

The authors are thankful to the Department of RIC, I.K.G. Punjab Technical University, Kapurthala, Punjab, India and providing an opportunity to carry out this research work.

Appendix A. Quality assessment scores.

Study ID	Reference to study ID	RQ addressed	Q1	Q2	Q3	Q4	Q5	Total scores
[S1]	[59]	RQ1	1	1	0	0	0	2
[S2]	[47]	RQ1	1	1	0	0	0	2
[S3]	[54]	RQ1	1	1	0	0	0	2
[S4]	[55]	RQ1	1	1	0	0	0	2
[S5]	[44]	RQ1	1	1	0	0	0	2
[S6]	[64]	RQ1	1	1	0	0	0	2
[S7]	[45]	RQ1	1	1	0	0	0	2
[S8]	[50]	RQ1	1	1	0	0	0	2
[S9]	[48]	RQ1	1	1	0	0	0	2
[S10]	[56]	RQ1	1	1	0	0	0	2
[S11]	[53]	RQ1	1	1	0	0	0	2
[S12]	[62]	RQ1	1	1	0	0	0	2
[S13]	[35]	RQ1	1	1	0	0	0	2
[S14]	[57]	RQ1	1	1	0	0	0	2
[S15]	[65]	RQ1	1	1	0	0	0	2
[S16]	[41]	RQ1	1	1	0	0	0	2
[S17]	[46]	RQ1	1	1	0	0	0	2
[S18]	[36]	RQ1	1	1	0	0	0	2
[S19]	[51]	RQ1	1	1	0	0	0	2
[S20]	[37]	RQ1	1	1	0	0	0	2
[S21]	[21]	RQ1	1	1	0	0	0	2
[S22]	[23]	RQ1	1	1	0	0	0	2
[S23]	[22]	RQ1	1	1	0	0	0	2
[S24]	[42]	RQ1	1	1	0	0	0	2
[S25]	[32]	RQ1	1	1	0	0	0	2
[S26]	[34]	RQ1	1	1	0	0	0	2
[S27]	[63]	RQ1	1	1	0	0	0	2
[S28]	[38]	RQ1	1	1	0	0	0	2
[S29]	[24]	RQ1	1	1	0	0	0	2
[S30]	[33]	RQ1	1	1	0	0	0	2
[S31]	[60]	RQ1	1	1	0	0	0	2
[S32]	[27]	RQ1	1	1	0	0	0	2
[S33]	[43]	RQ1	1	1	0	0	0	2
[S34]	[31]	RQ1	1	1	0	0	0	2
[S35]	[2]	RQ1	1	1	0	0	0	2
[S36]	[52]	RQ1	1	1	0	0	0	2
[S37]	[26]	RQ1	1	1	0	0	0	2
[S38]	[49]	RQ1	1	1	0	0	0	2
[S39]	[58]	RQ1	1	1	0	0	0	2
[S40]	[25]	RQ1	1	1	0	0	0	2
[S41]	[61]	RQ1	1	1	0	0	0	2
[S42]	[39]	RQ1	1	1	0	0	0	2
[S43]	[28]	RQ1	1	1	0	0	0	2
[S44]	[29]	RQ1	1	1	0	0	0	2
[S45]	[40]	RQ1	1	1	0	0	0	2
[S46]	[30]	RQ1	1	1	0	0	0	2
[S47]	[18]	RQ2	0	0	1	1	1	3
[S48]	[66]	RQ2	0	0	1	1	1	3
[S49]	[19]	RQ2	0	0	1	1	1	3

Appendix B. Data extraction for SLR-RQ1.

Study ID	Title and author	Year	Reference	Type of electronic data source	Source of publication	Study type	Characteristic identified
[S1]	Challenges, methodologies, and issues in the usability testing of mobile applications by Dongsong Zhang & Boonlit Adipat	2005	[59]	Journal	Research Gate	Review	Limited CPU, Diverse Mobile Connections, Network Availability
[S2]	Writing mobile code: essential software engineering for building mobile applications by Ivo Salmre	2005	[47]	Book	ACM	Review	Response time, Limited Memory, Limited CPU, Input Interface Means
[S3]	Testing requirements for mobile applications by V. L. L. Dantas, F. G. Marinho, A. L. da Costa, and R. M. C. Andrade	2009	[54]	Conference Proceeding	IEEE	A survey	Limited Memory, Limited Battery power, Limited CPU, Diverse Mobile Connections, Network Availability, Response Time, Integration with other Apps Different application types (Native, Hybrid, Web)
[S4]	Performance testing of mobile applications at the unit test level by H. Kim, B. Choi, W. Eric Wong	2009	[55]	Conference Proceeding	IEEE	Experimental	Limited Memory, Diverse Mobile Connections, Different application types, Diverse operating systems (software), Diverse devices(hardware), Network Availability, Response Time
[S5]	Software characteristics of m-learning applications by Boja C. and Bătăgan L.,	2009	[44]	Conference Proceedings	ACM	Review	Input Interface Means, Limited CPU, H/w and S/w Portability
[S6]	Mobile application development: essential new directions for IT by Chia-Chi Teng, R. Helps	2010	[64]	Conference Proceedings	IEEE	Experimental	Diverse operating systems (software), Diverse devices(hardware), Network Availability
[S7]	Software engineering issues for mobile application development by Wasserman	2010	[45]	Conference Proceedings	ACM	Review	Limited CPU, Limited Memory, integration with other applications, Localization(sensors position), Different application types (Native, Hybrid, Web), Diverse operating systems (software), Diverse devices(hardware), Input Interface Means, Limited Battery power
[S8]	Mobile application development: web vs. native by Andre Charland and Brian Leroux	2011	[50]	Journal	ACM	Review	Limited Memory, Limited Battery power, Limited CPU, Diverse Mobile Connections, Different application types (Native, Hybrid, Web), Intervention, Integration with other Apps
[S9]	A gui crawling-based technique for android mobile application testing by Amalfitano, D., Fasolino, A. R., & Tramontana, P.	2011	[48]	Conference Proceeding	IEEE	Experimental	Limited Memory Limited CPU, Input Interface Means
[S10]	Software testing of mobile applications: challenges and future research directions by Muccini, H., A. Di Francesco, and P. Esposito	2012	[56]	Conference Proceeding	ACM	Review	Limited Memory, Limited Battery power, Diverse Mobile Connections, Different application types, Diverse operating systems (software), Diverse devices(hardware), Integration with other Apps, Network Availability, Response Time, Limited CPU

(continued on next page)

Study ID	Title and author	Year	Reference	Type of electronic data source	Source of publication	Study type	Characteristic identified
[S11]	Software testing strategy for mobile phone by Guitao Cao, Jie Yang, Qing Zhou, and Weiting Chen	2012	[53]	Book	Research Gate	Review	Limited Memory, Limited CPU, Limited Battery power, Limited RAM, Different application types, Response Time
[S12]	Testing conformance of lifecycle-dependent properties of mobile applications by D. Franke, S. Kowalewski, C. Weise, and N. Prakobkosol	2012	[62]	Conference Proceedings	IEEE	Case study	Diverse Mobile Connections, Diverse devices(hardware), Network Availability
[S13]	Mobile applications software testing methodology by Kim HK.	2012	[35]	Book	SpringerLink	Implementation of a tool that does program analysis and testing for mobile applications	Limited Memory, Limited Battery power, Limited CPU, Limited RAM, Input Interface Means, Diverse operating systems, Diverse devices
[S14]	Activity page based functional test automation for android application by Lu, L., Hong, Y., Huang, Y., Su, K., Yan, Y.	2012	[57]	Conference Proceedings	IEEE	Case study	Limited Memory, Limited Battery power, Limited CPU, Limited RAM, Diverse Mobile Connections, Diverse devices, Diverse operating systems (software), Different application types(hardware), Network Availability, Response Time
[S15]	Mobile enterprise applications-current state and future directions by A. Giessmann, K. Stanoevska-Slabeva, and B. de Visser	2012	[65]	Conference Proceeding	ACM	Overview and Future Directions	Diverse Mobile Connections, Different application types, Diverse operating systems, Integration with other Apps
[S16]	Factors influencing quality of experience of commonly used mobile applications by Selim Ickin et al.	2012	[41]	Journal	IEEE	Survey	Input Interface Means, Limited CPU, Network Availability, Diverse devices(hardware), Limited Battery power
[S17]	Usability characteristics of mobile applications by Shamsudeen et al.	2012	[46]	Conference Proceedings	Research Gate	Review	Input Interface Means and sub categories (learnability, operability, attractiveness)
[S18]	Mobile application testing-challenges and solution approach through automation by B. Kirubakaran, V. Karthikeyani	2013	[36]	Conference Proceeding	IEEE	Review	Input Interface Means, Diverse Mobile Connections, Network Availability
[S19]	The survey, comparison and evaluation of cross-platform mobile application development tools by I. Dalmasso, S. K. Datta, C. Bonnet, and N. Nikaein	2013	[51]	Conference Proceeding	IEEE	Cross-platform development tools comparison	Diverse Mobile Connections, Diverse operating systems(Software), Diverse devices(hardware), Intervention
[S20]	Software quality testing model for mobile application by Liu Z., Hu Y., Cai L.	2014	[37]	Journal	SpringerLink	Proposed quality model and quality attributes testing requirements for mobile applications	Limited Memory, Limited Battery power, Limited CPU, Limited RAM, Input Interface Means, Different application types (Native, Hybrid, Web), Response Time

(*continued on next page*)

Study ID	Title and author	Year	Reference	Type of electronic data source	Source of publication	Study type	Characteristic identified
[S21]	Criteria for selecting mobile application testing tools by Arzenšek, Boštjan, and Marjan Heričko	2014	[21]	Conference Proceeding	Citeseer	Review	GUI, Input Interface Means, Diverse Mobile Connections, Diverse operating systems (software), Network Availability, Localization(sensors position)
[S22]	Using combinatorial approaches for testing mobile applications by Vilkomir, Sergiy & Amstutz, Brandi.	2014	[23]	Conference Proceedings	IEEE	Experimental	Limited Memory, GUI, Different application Types, Diverse operating systems (software), Diverse devices(hardware), Response Time
[S23]	Pattern-based GUI testing for mobile applications by P. Costa, A. C. R. Paiva and M. Nabuco	2014	[22]	Conference Proceedings	IEEE	Experimental	GUI, Input Interface Means
[S24]	Modeling the mobile application development lifecycle by Tejas Vithani and Anand Kumar	2014	[42]	Conference Proceedings	Research Gate	Review	Limited Battery power, Diverse devices(hardware), Diverse operating systems (software), Limited Memory, Input Interface Means
[S25]	An investigation on the characteristics of mobile applications: a survey study by Harleen Kaur Flora et al.	2014	[32]	Journal	Research Gate	Survey	Limited Battery power, Input Interface Means, GUI, Network Availability, Response Time, Diverse devices(hardware), Integration with other Apps, Localization(sensors position)
[S26]	Mobile application development: how to estimate the effort? By De Souza, Silva L., and de Aquino G.S	2014	[34]	Book	SpringerLink	Experimental	Limited Battery power, GUI, Limited CPU, Network availability, Localization(sensors position), Limited memory, Limited RAM, H/w and S/w Portability, Different application types (Native, Hybrid, Web), Response time, Integration with other Apps, Intervention
[S27]	Testing techniques for mobile device applications by Göth, B. R.	2015	[63]	Diploma Thesis	Research Gate	Review and case study	Diverse Mobile Connections, Network Availability
[S28]	Mobile application testing in industrial contexts: an exploratory multiple case-study by Zein S., Salleh N., Grundy J.	2015	[38]	Book	SpringerLink	A case study	Limited Memory, Input Interface Means, Diverse Mobile Connections, Different application types, Diverse operating systems (software), Network Availability Response Time
[S29]	Mobile application testing matrix and challenges by B. Amen, M. Sardasht, and J. Lu.	2015	[24]	Conference Proceeding	Research Gate	A case study	GUI, Diverse operating systems (software), Diverse devices(hardware)
[S30]	Compatibility testing service for mobile applications by Zhang T, Gao J, Cheng J, Uehara T.	2015	[33]	Conference Proceedings	IEEE	A case study for compatibility testing service for mobile apps	GUI, Diverse operating systems (software), Diverse devices(hardware)
[S31]	A survey on mobile users' software quality perceptions and expectations by Nitze Andre, Schmietendorf Andreas	2015	[60]	Conference Proceedings	IEEE	Review and survey	Limited Memory, Limited Battery power, Limited CPU, Network Availability
[S32]	Reviews on agile methods in the mobile application development process by Dewi, M.M, Nur Atiqah Sia, A.	2015	[27]	Conference Proceedings	IEEE	Review	Different application types (Native, Hybrid, Web), GUI, Diverse operating systems (software), Diverse devices (hardware), Localization (sensors position)

(*continued on next page*)

Study ID	Title and author	Year	Reference	Type of electronic data source	Source of publication	Study type	Characteristic identified
[S33]	Mobile testing in software industry using agile: challenges and opportunities by A. Santos and I. Correia,	2015	[43]	Conference Proceedings	IEEE	Review	Interaction with other applications, Limited Memory, Input Interface Means, Diverse Mobile Connections, Different application types, Diverse operating systems (software), Network Availability, Limited Battery power, Diverse devices(hardware)
[S34]	Software assurance practices for mobile applications A survey of the state of the art by Luis Corral et al.	2015	[31]	Journal	SpringerLink	Review	GUI, Diverse Mobile Connections, Diverse operating systems (software), Intervention
[S35]	A systematic mapping study of mobile application testing techniques by S. Zein, N. Salleh, and J. Grundy	2016	[2]	Journal	Science Direct	Review	Limited Memory, Limited Battery power, Limited CPU, Limited RAM, Input Interface Means, Diverse operating systems (software)
[S36]	Model driven development approaches for mobile applications: a survey by Umuhoza E., Brambilla M.	2016	[52]	Book	SpringerLink	Survey on model-driven approaches to the development of mobile apps	Diverse operating systems (software), Diverse devices(hardware), Integration with other Apps, Intervention
[S37]	Quality assurance of mobile applications: a systematic mapping study by Konstantin Holl and Frank Elberzhager.	2016	[26]	Conference Proceedings	ACM	Review and survey	GUI, Network Availability
[S38]	An approach for evaluating and improving the test processes of mobile application developments by Konstantin Holl, et al.	2016	[49]	Journal	Science Direct	Survey and Experimental	Limited Memory, Input Interface Means, Diverse Mobile Connections, Different application types, Diverse operating systems (software), Network Availability, Limited Battery power
[S39]	An empirical analysis of energy consumption of cross-platform frameworks for mobile development by Matteo Ciman, Ombretta Gaggi,	2017	[58]	Journal	Science Direct	Experimental	Limited Memory, Limited Battery power, Limited CPU, Limited RAM, Diverse operating systems (software), Diverse devices(hardware)
[S40]	A new method for mobile application testing using lean canvas to improving the test strategy by P. Nidagundi and L. Novickis	2017	[25]	Conference Proceedings	IEEE	Proposal	Limited Battery power, Limited CPU, Limited RAM, GUI, Diverse Mobile Connections, Different application types, Diverse Operating systems (software), Diverse devices(hardware), Intervention, Network Availability, Response Time, Localization(sensors position)
[S41]	PETrA: a software-based tool for estimating the energy profile of android applications by Dario Di Nucci et al.	2017	[61]	Conference Proceedings	ACM	Estimation tool for energy consumption	Limited Battery
[S42]	Reuse of model-based tests in mobile apps by Guilherme de Cleva Farto and Andre Takeshi Ando	2017	[39]	Conference Proceedings	ACM	Experimental	Localization(sensors position), Intervention, Input Interface Means

(*continued on next page*)

Study ID	Title and author	Year	Reference	Type of electronic data source	Source of publication	Study type	Characteristic identified
[S43]	Perspectives on usability guidelines for smartphone applications: An empirical investigation and systematic literature review by Ahmad, Naveed;Rextin, Aimal;Kulsoom, Um E.	2018	[28]	Journal	Science Direct	Review and empirical evaluation	Diverse operating systems (software), Diverse devices(hardware), GUI, Input Interface Means
[S44]	Characterizing mobile apps from a source and test code viewpoint by Davi Bernardo Silva et al.	2018	[29]	Journal	Science Direct	empirical study	Network Availability, GUI, Localization(sensors position), Diverse operating systems (software), Diverse devices(hardware)
[S45]	Model-driven development of mobile applications for Android and iOS supporting role-based app variability by Steffen Vaupe	2018	[40]	Conference Proceedings	SpringerLink	Experimental	Diverse operating systems (software), Diverse devices(hardware), Input Interface Means, Different application types (Native, Hybrid, Web)
[S46]	Testing embedded software: A survey of the literature by Vahid Garousi et al.	2018	[30]	Journal	Science Direct	Review	Limited Memory, Limited CPU, Limited Battery power, GUI

Data extraction for SLR-RQ2.

Study ID	Title of paper and author	Year	References	Type of electronic data source	Source of publication	Study type	Method	Tool support
[S47]	Architecture based reliability and testing estimation for mobile applications by Wadhwani V, Memon F, Hameed MM.	2008	[18]	Conference Proceeding	SpringerLink	Experimental	Architecture-Based	No
[S48]	Estimating manual test execution effort and capacity based on execution points by E. Aranha & P. Borba	2009	[66]	Journal	Citeseer	Experimental	Test size and Execution Complexity Measure	Yes
[S49]	Efficiency factor and risk factor based user case point test effort estimation model compatible with agile software development by Parvez, A.W.M.M.	2013	[19]	Conference Proceeding	IEEE	Model proposal	Use Case Point	No

Appendix C. Questionnaire

Questionnaire

My name is Anureet Kaur and I am a Ph.D. research scholar. My work is related to Mobile Application Software. Because you are working in a Mobile Application Development organization, I humbly invite you to participate in this research study by completing the attached survey. The information collected will be used for academic research purpose only. The aim of the survey is to determine if the current techniques in estimating the effort required in testing of mobile applications is accurate and what kind of things affect both the estimates and the actual effort needed to finish testing. Based on the current state I will then present possible suggestions for improvement in my research work.
The following questionnaire will require approximately 10 minutes in completing. There is no compensation for responding nor is there any known risk. In order to ensure that all information will remain confidential, please do not include your name. Participation is strictly voluntary and you may refuse to participate at any time. Thank you for taking the time to assist me in my educational endeavors. If you require additional information or have questions, please contact me at the number listed below.
Sincerely,
(Anureet Kaur)
(Mobile no. +91-9814612521 and e-mail address anumahal@gmail.com)
(Supervisor's Name: Dr. Kulwant Kaur)
(+91-9417633102 and/or e-mail address kulwantkrgill@yahoo.com)

Electronic Consent:
By clicking the next button you agree to participate in this survey voluntarily and you have read the above information.

*Required

1. Name

2. Organization *

3. City *

4. Country *

5. Total Number of employees in organization *

6. Role in Organization *

7. E-mail Address

8. Experience in software industry *Mark only one oval.*

- 0-5 years
- 5-10 years
- 10-15years
- More than 15 years

9. Do you perform testing for mobile applications? *Mark only one oval.*

- Yes
- No

If Yes to previous question, experience in testing for mobile applications *Mark only one oval.*

- 1-5 years
- 5-10 years
- More than 10 years
- Other:

10. Do you take part in estimation of testing for mobile application? *Mark only one oval.*

- Yes
- No

If yes to previous question, Please specify experience in Estimation of Mobile Application Testing. *Mark only one oval.*

- 0-5 years
- 5-10 years
- More than 10 years
- Other:

11. Do you think estimation of testing for mobile applications are different from other software (Desktop/laptop/web)? *Mark only one oval.*

- Yes
- No
- Maybe

12. What is the current method used in your organization for testing estimation of mobile applications? *Mark only one oval.*

- Expert Judgment
- Best Guess
- Work Breakdown Structure
- Function Point/Testing Point Analysis
- 3-Point Software Testing Estimation Technique
- Wide Band Delphi Method
- Use Case Points
- Other:

13 .How long (years) have you been using the method answered in previous question? *Mark only one oval.*

- 0-1 year
- 1-2 years
- 2-3 years
- > 3 years
- Other:

14. How is the current method adopted for testing estimation is accurate to actual testing effort?
Mark only one oval.

	1	2	3	4	5	
Not accurate at all	○	○	○	○	○	Very accurate

15. Who does the estimation for testing in your organization? *Mark only on oval.*

- Single person in team
- Entire Team
- Manager/Team Leader
- Some members in team

16. Does your organization use Historical data for the previous testing estimation for improvement and accuracy? *Mark only one oval.*

- Yes
- No

17. What data would you gather and analyze before starting testing estimation?

18. Do you use any tool for testing effort estimation for mobile applications? *Mark only one oval.*

- Yes
- No

If yes, in previous question, specify the tool name.

19. Do you include buffer time along with Testing Estimation? *Mark only one oval.*

- Yes
- No

If answer to previous question is yes, then how much buffer time.

20. What measurement unit is used to for testing effort estimation? *Mark only one oval.*

- Person Hours
- Person Days
- Person Months
- Other:

21. Do you keep evidence of record for previous mobile application Testing estimates? *Mark only one oval.*

- Yes
- No

If answer is yes to previous question, then in specify format.

22. What method do you use for testing of mobile applications? *Mark only one oval.*

- Manual Testing
- Automated Testing

23. What is the average size in your organization for testing team for a mobile applications? *Mark only one oval.*

- 1-3 persons
- 3-5 persons
- >5 persons

24. What are different Levels of testing covered for mobile application *Tick all that apply.*

- Unit Testing
- Integration Testing
- System Testing
- User Acceptance Testing

25. What are different Types of testing performed for mobile applicati *Tick all that apply.*

- Functional Testing
- Performance Testing
- Memory leakage Testing
- Usability Testing
- Interrupt Testing
- Security Testing
- Location Testing
- Compatibility Testing
- Regression Testing
- Other:

26. What testing strategy is followed for mobile application testing *Tic all that apply.*

- Testing on Real Devices
- Testing on Emulator
- Testing on Cloud
- Other:

27. Are testing effort estimations were underestimated or overestimated for the earlier completed mobile applications and if yes, then by how much? *Mark only one oval.*

- By 20% or less
- By 21-50 %
- By more than 50%
- Total success (as per estimated)

28. If there were deviations between estimated and actual testing effort in the earlier completed mobile applications, then what do you think the reason behind these deviations?

29. Are you satisfied with estimated and actual testing effort results using current testing estimation methodology in earlier completed mobile applications? *Mark only one oval.*

	1	2	3	4	5	
Very unsatisfied	◯	◯	◯	◯	◯	Very satisfied

If answer to previous question 1, 2 or 3, then specify the problem with current methodology.

30. Specify some suggestions to improve the results of testing estimation.

31. What programming languages are used for mobile application development? *Tick all that apply.*

- Java
- Python
- PHP
- C#
- Objective C
- Ruby
- Perl
- HTML5
- C/C++
- JavaScript
- Rust
- Other:

32. What platform are mobile applications developed for? *Tick all that apply.*

- iOS
- Android
- Windows Phone
- Symbian
- Blackberry
- Palm OS
- Ubuntu touch
- Other:

33. What categories of mobile applications does your organization develop? ***Tick all that apply.***

- ☐ Games
- ☐ Travel
- ☐ Health Care
- ☐ Social Networking
- ☐ Education
- ☐ Business
- ☐ Book & References
- ☐ Food & Drink
- ☐ Maps & Navigation
- ☐ Music & Audio
- ☐ News & Magazines
- ☐ Shopping
- ☐ Weather
- ☐ Sports
- ☐ Auto & Vehicles
- ☐ House & Home
- ☐ Personalization
- ☐ Other:

34. Do you consider estimation of mobile application development and estimation of testing as two separate processes? ***Mark only one oval.***

- ◯ Yes
- ◯ No

35. How much does the following characteristics of mobile application influence the testing estimation ****Mark only one oval per row.***

	No Effect	Low Effect	Medium Effect	High Effect	Very High Effect
Restricted RAM	◯	◯	◯	◯	◯
Graphical User Interface	◯	◯	◯	◯	◯
Input Interface Means	◯	◯	◯	◯	◯
Network Availability	◯	◯	◯	◯	◯
Limited Performance	◯	◯	◯	◯	◯
Limited Memory	◯	◯	◯	◯	◯
Diverse Connections	◯	◯	◯	◯	◯
Intervention	◯	◯	◯	◯	◯
H/w and S/w Portability	◯	◯	◯	◯	◯
Response time	◯	◯	◯	◯	◯
Battery Power	◯	◯	◯	◯	◯
Localization (sensors position)	◯	◯	◯	◯	◯
Integration with other mobile Apps	◯	◯	◯	◯	◯
Type of Mobile App (Native, Web or Hybrid)	◯	◯	◯	◯	◯

References

[1] G. Chen, D. Kotz, A Survey of Context-Aware Mobile Computing Research, 2000 Dartmouth Comput. Sci. Tech. Rep. doi:10.1.1.117.4330.

[2] S. Zein, N. Salleh, J. Grundy, A systematic mapping study of mobile application testing techniques, J. Syst. Softw. 117 (2016) 334–356, doi:10.1016/j.jss.2016.03.065.

[3] L.S. de Souza, G.S. de Aquino Jr, MEFFORTMOB: a effort size measurement for mobile application development, Int. J. Softw. Eng. Appl. 5 (2014) 63–81, doi:10.5121/ijsea.2014.5405.

[4] S.K. Sehra, Y.S. Brar, N. Kaur, S.S. Sehra, Research patterns and trends in software effort estimation, Inf. Softw. Technol. 91 (2017) 1–21, doi:10.1016/j.infsof.2017.06.002.

[5] K.R. Jayakumar, A. Abran, A survey of software test estimation techniques, J. Softw. Eng. Appl. 6 (2013) 47–52, doi:10.4236/jsea.2013.610A006.

[6] P.R. Srivastava, A. Bidwai, A. Khan, K. Rathore, R. Sharma, X.S. Yang, An empirical study of test effort estimation based on bat algorithm, Int. J. Bio-Inspired Comput. 6 (2014) 57, doi:10.1504/IJBIC.2014.059966.

[7] S. Nageswaran, Test effort estimation using use case points, in: Proc. 14th Int. Internet Softw. Qual. Week., 2001.

[8] C. Abhishek, V.P. Kumar, H. Vitta, P.R. Srivastava, Test effort estimation using neural network, J. Softw. Eng. Appl. 03 (2010) 331–340, doi:10.4236/jsea.2010.34038.

[9] P.P. Souza, M.W. Barbosa, Tailoring the test point analysis estimation technique in a software testing process, IV Encontro Bras, Testes Recife., 2010.

[10] C.M. Zapata-Jaramillo, D.M. Torres-Ricaurte, Test effort: a pre-conceptual-schema-based representation, DYNA 81 (2014) 132–137, doi:10.15446/dyna.v81n186.39753.

[11] P.R. Srivastava, A. Varshney, P. Nama, X.S. Yang, Software test effort estimation: a model based on cuckoo search, Int. J. Bio-Inspired Comput. 4 (2012) 278, doi:10.1504/IJBIC.2012.049888.

[12] A. Sharma, D.S. Kushwaha, An empirical approach for early estimation of software testing effort using SRS document, CSI Trans. ICT 1 (2013) 51–66, doi:10.1007/s40012-012-0003-z.

[13] P.R. Srivastava, Estimation of software testing effort using fuzzy multiple linear regression, Int. J. Softw. Eng. Technol. Appl. 1 (2015) 145, doi:10.1504/IJSETA.2015.075602.

[14] A. Bhattacharyya, T. Malgazhdarov, PredSym: estimating software testing budget for a bug-free release, in: Proc. 7th Int. Work. Autom. Test Case Des. Sel. Eval. – A-TEST 2016, 2016, pp. 16–22, doi:10.1145/2994291.2994294.

[15] S. Islam, B.B. Pathik, M.H. Khan, M. Habib, Software test estimation tool: comparable with COCOMOII model, in: IEEE Int. Conf. Ind. Eng. Eng. Manag., 2016, pp. 204–208, doi:10.1109/IEEM.2016.7797865.

[16] C. Arumugam, C. Babu, Test size estimation for object oriented software based on analysis model, J. Softw. 10 (2015) 713–729, doi:10.17706/jsw.10.6.713-729.

[17] E. Aranha, P. Borba, An estimation model for test execution effort, in: First Int. Symp. Empir. Softw. Eng. Meas. (ESEM 2007), IEEE, 2007, pp. 107–116, doi:10.1109/ESEM.2007.73.

[18] V. Wadhwani, F. Memon, M.M. Hameed, Architecture based reliability and testing estimation for mobile applications, in: Commun. Comput. Inf. Sci., Springer, Berlin, Heidelberg, 2008, pp. 64–75, doi:10.1007/978-3-540-89853-5-9.

[19] A.W.M.M. Parvez, Efficiency factor and risk factor based user case point test effort estimation model compatible with agile software development, in: Proc. – 2013 Int. Conf. Inf. Technol. Electr. Eng. "Intelligent Green Technol. Sustain. Dev. ICITEE 2013, Yogyakarta, Indonesia., 2013: pp. 113–118. doi:10.1109/ICITEED.2013.6676222.

[20] B. Kitchenham, S. Charters, Guidelines for Performing Systematic Literature Reviews in Software Engineering, 2007, doi:10.1145/1134285.1134500.

[21] B. Arzenšek, M. Heričko, Criteria for selecting mobile application testing tools, in: CEUR Workshop Proc., 2014, pp. 1–8.

[22] P. Costa, A.C.R Paiva, M. Nabuco, Pattern based GUI testing for mobile applications, in: Proc. – 2014 9th Int. Conf. Qual. Inf. Commun. Technol. QUATIC 2014, 2014, pp. 66–74, doi:10.1109/QUATIC.2014.16.

[23] S. Vilkomir, B. Amstutz, Using combinatorial approaches for testing mobile applications, in: 2014 IEEE Seventh Int. Conf. Softw. Testing, Verif. Valid. Work., 2014, pp. 78–83, doi:10.1109/ICSTW.2014.9.

[24] B.M. Amen, S.M. Mahmood, J. Lu, Mobile **application testing matrix and cha**llenges, in: Comput. Sci. Inf. Technol. (CS IT), Sydney, Australia, 2015, pp. 27–40, doi:10.5121/csit.2015.50403.

[25] P. Nidagundi, L. Novickis, New method for mobile application testing using lean canvas to improving the test strategy, in: 12th Int. Sci. Tech. Conf. Comput. Sci. Inf. Technol., 2017, pp. 171–174, doi:10.1109/STC-CSIT.2017.8098761.

[26] K. Holl, F. Elberzhager, Quality assurance of mobile applications : a systematic mapping study, in: 15th Int. Conf. Mob. Ubiquitous Multimed, ACM, New York, NY, USA, 2016, pp. 101–113, doi:10.1145/3012709.3012718.

[27] M. Dewi, A. Nur Atiqah Sia, Reviews on agile methods in mobile application development process, in: 2015 9th Malaysian Softw. Eng. Conf., 2015, pp. 161–165, doi:10.1109/MySEC.2015.7475214.

[28] N. Ahmad, A. Rextin, U.E. Kulsoom, Perspectives on usability guidelines for smartphone applications: an empirical investigation and systematic literature review, Inf. Softw. Technol. 94 (2018) 130–149, doi:10.1016/j.infsof.2017.10.005.

[29] D.B. Silva, M.M. Eler, V.H.S. Durelli, A.T. Endo, Characterizing mobile apps from a source and test code viewpoint, Inf. Softw. Technol. 101 (2018) 32–50, doi:10.1016/j.infsof.2018.05.006.

[30] V. Garousi, M. Felderer, Ç.M. Karapıçak, U. Yılmaz, Testing embedded software: a survey of the literature, Inf. Softw. Technol. (2018), doi:10.1016/J.INFSOF.2018.06.016.

[31] L. Corral, A. Sillitti, G. Succi, Software assurance practices for mobile applications, Computing 97 (2015) 1001–1022, doi:10.1007/s00607-014-0395-8.

[32] H.K. Flora, X. Wang, S.V Chande, An investigation on the characteristics of mobile applications: a survey study, Int. J. Inf. Technol. Comput. Sci. 6 (2014) 21–27, doi:10.5815/ijitcs.2014.11.03.

[33] T. Zhang, J. Gao, J. Cheng, T. Uehara, Compatibility testing service for mobile applications, in: Serv. Syst. Eng. (SOSE), 2015 IEEE Symp., 2015, pp. 179–186, doi:10.1109/SOSE.2015.35.

[34] L.S. de Souza, G.S. de Aquino, Mobile application development: how to estimate the effort? in: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 2014, pp. 63–72, doi:10.1007/978-3-319-09156-3_5.

[35] H.K. Kim, Mobile Applications Software Testing Methodology, Springer, Berlin, Heidelberg, 2012, doi:10.1007/978-3-642-35270-6_22.

[36] B. Kirubakaran, V. Karthikeyani, Mobile application testing—challenges and solution approach through automation, in: 2013 Int. Conf. Pattern Recognition, Informatics Mob. Eng., 2013, pp. 79–84, doi:10.1109/ICPRIME.2013.6496451.

[37] Z. Liu, Z. Hu, L. Cai, Software quality testing model for mobile application, MobiWIS 2014 Mob. Web Inf. Syst., Springer, Cham, 2014, doi:10.1007/978-3-319-10359-4.

[38] S. Zein, N. Salleh, J. Grundy, Mobile application testing in industrial contexts: an exploratory multiple case-study, in: G.G. Fujita H. (Ed.), Intell. Softw.

Methodol. Tools Tech. (SoMeT). Commun. Comput. Inf. Sci., Springer, Cham, 2015, doi:10.1007/978-3-319-22689-7_3.
[39] G. de Cleva Farto, A.T. Endo, Reuse of model-based tests in mobile apps, in: Proc. 31st Brazilian Symp. Softw. Eng. – SBES'17, ACM Press, New York, New York, USA, 2017, pp. 184–193, doi:10.1145/3131151.3131160.
[40] S. Vaupel, G. Taentzer, R. Gerlach, M. Guckert, Model-driven development of mobile applications for Android and iOS supporting role-based app variability, Softw. Syst. Model. 17 (2018) 35–63, doi:10.1007/s10270-016-0559-4.
[41] S. Ickin, K. Wac, M. Fiedler, L. Janowski, J.-H. Hong, A.K. Dey, Factors influencing quality of experience of commonly used mobile applications, IEEE Commun. Mag. 50 (2012) 48–56, doi:10.1109/MCOM.2012.6178833.
[42] T. Vithani, A. Kumar, Modeling the mobile application development, in: Proc. Int. MultiConference Eng. Comput. Sci., 2014, pp. 596–600.
[43] A. Santos, I. Correia, Mobile testing in software industry using agile: challenges and opportunities, in: 8th IEEE Int. Conf. Softw. Testing, Verif. Validation, ICST 2015, 2015, pp. 1–2, doi:10.1109/ICST.2015.7102625.
[44] C. Boja, L. Batagan, Software characteristics of m-learning applications, in: 10th WSEAS Int. Conf. Math. Comput. Bus. Econ., 2009, pp. 88–93.
[45] A.I. Wasserman, Software engineering issues for mobile application development, in: Proc. FSE/SDP Work. Futur. Softw. Eng. Res. ACM, 2010., 2010, pp. 397–400, doi:10.1145/1882362.1882443.
[46] R. Shamsudeen, A. Semiu, O.P. Hector, Usability characteristics of mobile applications, in: First Int. Conf. Behavioural Soc. Sci. Res. 2012, 2012, pp. 1–5.
[47] I. Salmre, Writing Mobile Code: Essential Software Engineering for Building Mobile Applications, Addison-Wesley Professional, 2005.
[48] D. Amalfitano, A.R. Fasolino, P. Tramontana, A GUI crawling-based technique for android mobile application testing, in: Proc. – 4th IEEE Int. Conf. Softw. Testing, Verif. Valid. Work. ICSTW 2011, 2011, pp. 252–261, doi:10.1109/ICSTW.2011.77.
[49] K. Holl, V. Vieira, I. Faria, An approach for evaluating and improving the test processes of mobile application developments, Procedia Comput. Sci. 94 (2016) 33–40, doi:10.1016/J.PROCS.2016.08.009.
[50] A. Charland, B. Leroux, Mobile Application Development : Web vs . native, Commun, ACM, 2011, doi:10.1145/1941487.
[51] I. Dalmasso, S.K. Datta, C. Bonnet, N. Nikaein, Survey, comparison and evaluation of cross platform mobile application development tools, in: 2013 9th Int. Wirel. Commun. Mob. Comput. Conf., 2013, pp. 323–328, doi:10.1109/IWCMC.2013.6583580.
[52] E. Umuhoza, M. Brambilla, Model driven development approaches for mobile applications: a survey, in: T.D. Younas M, I. Awan, N. Kryvinska, C. Strauss (Eds.), Int. Conf. Mob. Web Inf. Syst, Springer, Cham, 2016, pp. 93–107, doi:10.1007/978-3-319-44215-0_8. 2016.
[53] G. Cao, J. Yang, Q. Zhou, W. Chen, Software Testing Strategy for Mobile Phone, InTech, 2012, doi:10.5772/38105.
[54] V.L. Dantas, M.F.G. L. da Costa A, R.M.C. Andrade, Testing requirements for mobile applications, in: 2009 24th Int. Symp. Comput. Inf. Sci. Isc. 2009, 2009, pp. 555–560, doi:10.1109/ISCIS.2009.5291880.
[55] H. Kim, B. Choi, W.E. Wong, Performance testing of mobile applications at the unit test level, in: SSIRI 2009 – 3rd IEEE Int. Conf. Secur. Softw. Integr. Reliab. Improv., 2009, pp. 171–181, doi:10.1109/SSIRI.2009.28.
[56] H. Muccini, A. di Francesco, P. Esposito, Software testing of mobile applications: challenges and future research directions, 7th Int. Work. Autom. Softw. Test (AST 2012), 2012, doi:10.1109/IWAST.2012.6228987.
[57] L. Lu, Y. Hong, Y. Huang, K. Su, Y. Yan, Activity page based functional test automation for android application, in: Proc. 2012 3rd World Congr. Softw. Eng. WCSE 2012, 2012, pp. 37–40, doi:10.1109/WCSE.2012.15.
[58] M. Ciman, O. Gaggi, An empirical analysis of energy consumption of cross-platform frameworks for mobile development, Pervasive Mob. Comput. 39 (2017) 214–230, doi:10.1016/j.pmcj.2016.10.004.
[59] D. Zhang, B. Adipat, Challenges, methodologies, and issues in the usability testing of mobile applications, Int. J. Hum. Comput. Interact. 18 (2005) 293–308, doi:10.1207/s15327590ijhc1803_3.
[60] A. Nitze, A. Schmietendorf, A survey on mobile users' software quality perceptions and expectations, 2015 IEEE Eighth Int. Conf. Softw. Testing, Verif. Valid. Work., 2015, doi:10.1109/ICSTW.2015.7107417.
[61] D. Di Nucci, F. Palomba, A. Prota, A. Panichella, A. Zaidman, A. De Lucia, PETrA: a software-based tool for estimating the energy profile of android applications, in: Proc. – 2017 IEEE/ACM 39th Int. Conf. Softw. Eng. Companion, ICSE-C 2017, 2017, pp. 3–6, doi:10.1109/ICSE-C.2017.18.
[62] D. Franke, S. Kowalewski, C. Weise, N. Prakobkosol, Testing conformance of life cycle dependent properties of mobile applications, in: Proc. – IEEE 5th Int. Conf. Softw. Testing, Verif. Validation, ICST 2012, 2012, pp. 241–250, doi:10.1109/ICST.2012.104.
[63] B.R. Göth, Testing Techniques For Mobile Device Applications, Masaryk University, 2015.
[64] C.-C.T.C.-C. Teng, R. Helps, Mobile application development: essential new directions for IT, Inf. Technol. New Gener. (ITNG), 2010 Seventh Int. Conf., 2010, doi:10.1109/ITNG.2010.249.
[65] A. Giessmann, K. Stanoevska-Slabeva, B. de Visser, Mobile enterprise applications – current state and future directions, in: 45th Hawaii Int. Conf. Syst. Sci., 2012, pp. 1363–1372, doi:10.1109/HICSS.2012.435.
[66] E. Aranha, P. Borba, Estimating manual test execution effort and capacity based on execution points, Int. J. Comput. Appl. 31 (2009) 167–172, doi:10.1080/1206212X.2009.11441938.
[67] T.L. Saaty, How to make a decision: the analytic hierarchy process, Interfaces 24 (1994) 19–43, doi:10.1287/inte.24.6.19.
[68] M. Velasquez, P.T. Hester, An analysis of multi-criteria decision making methods, Int. J. Oper. Res. 10 (2013) 56–66, doi:10.1007/978-3-319-12586-2.
[69] R. Ramanathan, Data envelopment analysis for weight derivation and aggregation in the analytic hierarchy process, Comput. Oper. Res. 33 (2006) 1289–1307, doi:10.1016/j.cor.2004.09.020.
[70] A. Ishizaka, M. Lusti, How to derive priorities in AHP: a comparative study, Cent. Eur. J. Oper. Res. 14 (2006) 387–400, doi:10.1007/s10100-006-0012-9.
[71] B. Blagojevic, B. Srdjevic, Z. Srdjevic, T. Zoranovic, Deriving weights of the decision makers using AHP group consistency measures, Fundam. Informaticae. 144 (2016) 383–395, doi:10.3233/FI-2016-1342.
[72] T. Saaty, L. Vargas, Models, methods, concepts & applications of the analytic hierarchy process, 2012. doi:10.1007/978-1-4614-3597-6.
[73] M. Karanik, L. Wanderer, J.A. Gomez-Ruiz, J.I. Pelaez, Reconstruction methods for AHP pairwise matrices: how reliable are they? Appl. Math. Comput. 279 (2016) 103–124, doi:10.1016/j.amc.2016.01.008.
[74] A. Shaout, M.K. Yousif, Performance evaluation – methods and techniques survey, Int. J. Comput. Inf. Technol. 3 (2014) 966–979.
[75] T.L. Saaty, The analytic hierarchy process, Education (1980), doi:10.3414/ME10-01-0028.
[76] D. Cuellar, Appium, 2011 Accessed 26 March 2018. http://appium.io/.
[77] Tableau Software Company, 2003 Accessed 26 March 2018. https://www.tableau.com/about.
[78] J. Allen, HP Quality Center and Jira, 2013 Accessed 26 March 2018. http://quality-center-basics.blogspot.in/.