Systematic reviews in mobile app software engineering: A tertiary study

Samer Zein[a], Norsaremah Salleh[b], John Grundy[c]

[a]*Department of Computer Science, Birzeit University, P.O. Box 14, Birzeit, Palestine*
[b]*Department of Computer Science, International Islamic University Malaysia, P.O. Box 10, 50728 Kuala Lumpur, Malaysia*
[c]*Department of Software Systems and Cybersecurity, Faculty of Information Technology, Monash University, Wellington Road, Clayton, VIC 3800, Australia*

Abstract

Context: A number of secondary studies in the form of systematic reviews and systematic mapping studies exist in the area of mobile application software engineering.
Objective: The focus of this paper is to provide an overview and analysis of these secondary studies of mobile app software engineering for researchers and practitioners.
Method: We conducted a systematic tertiary study following the guidelines by Kitchenham et al. to classify and analyze secondary studies in this area.
Results: After going through several filtration steps, we identified 24 secondary studies addressing major software software engineering phases, such as initiation, requirements engineering, design, development and testing. The majority of the secondary studies focused on testing and design phases. Specific research topics addressed by the included studies were: usability evaluation, test automation, context-aware testing, cloud-based development, architectural models, effort and size estimation models, defect prediction, and GUI testing. We found that the trend in secondary studies is towards more specific areas of mobile application software engineering such as architectural design models, context-aware testing, testing of non-functional requirements, mobile cloud computing, and intelligent mobile applications. Research directions and some identified practices for practitioners were also identified.
Conclusions: Mobile application software engineering is an active research area. The area can benefit from additional research in terms of secondary studies targeting evolution, maintenance, requirements engineering, and cross-platform mobile application development. Additionally, some of the secondary studies identify some useful practices for practitioners.

Keywords: Mobile app software development, tertiary study, systematic reviews, mapping study.

1. Introduction

The prevalence of mobile applications – that we term *'apps'* – in every aspect of our modern life is very significant. Nowadays we can hardly find one of our daily activities that is not assisted through one or more mobile apps. In 2016 there were 3.6 billion mobile users worldwide, while in 2022, there were 6.5 billion app users [1]. Latest figures also show that in 2022 there were 2.87 million mobile apps available at Google Play Store and 1.96 million apps available at Apple iOS Store [2]. People all over the globe use mobile apps in contexts ranging from entertainment, such as playing online games, streaming of videos, social networking, to critical ones such as banking, payment of bills, and controlling vehicles and home devices.

Since the creation of the first smart phone device by Apple in 2007, there has been a dramatic increase in the number of mobile apps and mobile app users. Accordingly, this new field of technology has joined the mainstream platform for the software development industry. The research community has also recognized the demands and challenges of this technological shift. Accordingly, many primary research studies have been carried out covering various topics in the mobile app software engineering area. Further, several secondary studies in the form of both systematic reviews (SR) and systematic mapping studies (SMS) have been published since the year 2013. However, there has to date been no tertiary study in this important area of research that identifies, analyzes, and classifies these individual secondary studies.

A tertiary study answers wider research questions by identifying and analyzing SRs and SMSs by following the same methodology as systematic mapping reviews [3]. In this research, we conducted a systematic tertiary study to identify, analyze and classify secondary studies in the area of mobile app software engineering. Only secondary studies which are systematic reviews or systematic mapping

*Samer Zein

studies were included so as to maintain high quality and research rigor. Our focus in this work is specifically on secondary studies that target the technical areas of *mobile app software engineering*, and thus can assist practitioners during the various phases of the mobile app development process. We thus excluded papers that focus on the use or application of mobile apps in specific domains such as COVID 19, Diabetes, Crowdsourcing, mobile learning, and so forth, or secondary studies of mobile networking technologies, energy consumption, and similar. Examples of excluded studies are [4, 5, 6].

We formulated the following set of key research questions for our study:

- RQ1: How many systematic secondary studies on mobile app software engineering have been published?
- RQ2: What research topics were addressed in these SRs/SMSs?
- RQ3: What phases of mobile app development are addressed in these secondary studies?
- RQ4: What are the key trends in mobile app development as reported in these secondary studies?
- RQ5: What is the overall quality of these mobile app SE secondary studies?
- RQ6: What are the key recommendations made for future research studies?
- RQ7: Are any useful recommendations for practitioners made in the secondary studies?

This work makes the following key contributions:

- we carried out the first tertiary study of secondary studies on mobile app software engineering;
- we identified 24 quality secondary studies targeting eight research areas : software development models, effort estimation, quality assurance, usability, cloud-based development, requirements engineering, defect prediction, and architectural design; and
- we analysed these 24 secondary studies to determine quality, focus, coverage, scope, limitations and key recommendations for future work in mobile app software engineering.

2. Study Design

Our research methodology is based on the guidelines provided by [7] as well as our experiences from of our previous secondary and tertiary studies [8, 9].

2.1. Search Process

Our search process employed the online databases of IEEExplore, ACM Digital Library, Springer Link, and Science Direct. We chose these as they are the primary digital libraries provided by the publishers (IEEE, ACM, Springer, Elsevier) of most software engineering studies. We did not specify a starting date and we searched all relevant SRs and mapping studies up to April 2023. In order to find the relevant SRs and mapping studies, we formulated the following search string:

(”mobile” OR ”Android”) AND (”app” OR ”application”) AND (”systematic review” OR ”systematic literature review” OR ”systematic map” OR ”systematic mapping” OR ”mapping study”).

2.2. Study Selection

The inclusion criteria were:

1. The secondary studies must be directly related to the development of mobile apps (including methods, techniques or practices) or application of software engineering activities to the development of mobile apps;
2. The studies must be conducted following SR or systematic mapping study guidelines.

Regarding the second inclusion criterion, the first author investigated if the secondary study clearly stated that they followed the guidelines by Kitchenham [7] or Petersen [10] (SR or SMS). Then the first and second authors reviewed the method section of secondary study to make sure that the authors followed the guidelines and if the study design and conduct is really systematic. This was done by investigating that the secondary study has defined a set of research questions, the overall search process is clear, systematic, and replicable, data extraction and synthesis procedures are clear.

Our exclusion criteria were:

1. Secondary studies that focus on the use of mobile apps or mobile technology only in certain domains eg. eHealth, finance, education, such as [4, 5, 11];
2. Short secondary studies that are less than 6 pages;
3. Papers not written in the English language.

The first exclusion criterion follows from the first inclusion criterion. That is, the secondary study has to be directly related to the area of mobile app software engineering, investigating certain methods, techniques, or practices in this area. Regarding the second exclusion criteria, most systematic studies remove very short poster and new ideas papers e.g. less than 3 or 4 pages. As this is a tertiary study selecting secondary studies, we believe that secondary studies of less than 6 pages are almost certainly going to be very shallow in terms of reporting their methods and results. The study filtration process was divided into 5 main phases as follows:

1. Phase 1: apply search strings to related databases.
2. Phase 2: reading the titles.
3. Phase 3: removing of duplicates and reading abstracts.
4. Phase 4: apply inclusion and exclusion criteria.
5. Phase 5: perform backward snowballing,

Before the start of the first phase, we evaluated our search string against a predefined set of secondary studies we were aware of in advance to see if the search string is effective. The predefined set of papers were carefully selected to contain the terms "mobile apps", "mobile application", and "Android" in their titles. Our search string proved to be effective as it returned all such papers. Then we proceed with the first phase, in which we applied the search string to the digital databases. In the second phase, the titles were read and irrelevant papers were excluded. In the third phase the selection continued by first removing the duplicates, and then reading of abstracts. In phase 4, we applied our predefined inclusion and exclusion criteria. Finally, in last phase, we performed backward snowballing by searching through the references of all papers in the final set to make sure we did not miss any relevant secondary studies. More specifically, we manually went through all the references of each paper looking for papers that did not appear in the search results.

The search process was primarily done by the first author. However, to minimize threats related to interpretive validity, search results of every phase was shared and inspected by second and third authors. Further, we held several joint meetings to discuss and inspect paper selection and search results and to resolve issues related to study selection. All discrepancies were discussed and resolved during the joint meetings.

2.3. Data Extraction and Analysis Process

Once the final set of studies had been identified, data extraction commenced in order to extract relevant data for answering our RQs. All information required for data analysis and synthesis was recorded using Microsoft Excel and Google spreadsheets.The following information was extracted from each study:

- Biblographic information: (title, abstract, publication year, publication type: conference/journal)
- Keywords
- Number of included primary studies
- Search strings
- The secondary study's key research questions
- The online databases searched to find primary studies
- The primary study years covered by this secondary study
- The software engineering phase(s) the primary studies address
- The key mobile app SE research topics/areas discussed
- The systematic guidelines applied
- Main findings of the secondary study
- Future research recommendations
- Any limitations or gaps identified
- Any good practices for practitioners recommended

When deciding on the software engineering phase or phases that a secondary study targets, we employed the generic definition of software engineering phases as described by Sommerville [12]: (i) Initiation Phase: this is the first phase in software development where feasibility studies are conducted and estimation plans are developed; (ii) Requirements Engineering Phase: which includes requirements elicitation, specification, and validation; (iii) System Design Phase: which includes system models design, architecture design, database design, etc; (iv) Development Phase: which includes actual software development; (v) Testing Phase: software verification and validation; and (vi) System Evolution: system maintenance. A secondary study can target more than one software engineering phase. For instance, study S4 targets both the System Design and Testing phases.

In order to answer research questions RQ1 and RQ5, we performed descriptive analysis for the quantitative data collected. To answer research questions RQ2, RQ3, RQ4 and RQ6, we performed thematic coding analysis [13] for the qualitative data. Further, in order to identify good practices for developers (RQ7), we went through all secondary studies that we included, looking for practices or guidelines that are explicitly directed for practitioners.

In order to identify research topics addressed by the included secondary studies, we applied a thematic coding to identify, analyze, and report themes in our data [13]. The thematic process was done by the first author, and consisted of two phases. In the first phase, the first author analyzed the titles, abstracts, and keyword and looked for phrases and concepts that reflected research problems, paper main focus area, and contributions. If the abstracts did not include high-quality information, the introduction, conclusion, and sometimes the discussion sections are referred to.

In the second phase, the similar phrases or concepts identified in the first phase were grouped together to form a higher-level understanding (research topics) regarding contribution and main focus of the paper. Resulting research topics were reviewed by second and third authors in a review meeting. Secondary studies can have several research topics. As an example of the thematic process, the S1 secondary study was assigned to two main research topics: "Usability Evaluation Approaches" and "Mobile App Usability Heuristics". The secondary study of S5 had five research topics: "Mobile App Testing Techniques", "Usability Testing", "Context awareness", "Automation Testing", and "Security Testing". The main focus of S5 is on testing techniques in general. However, each of the resulting research topics was analyzed and discussed in-depth by the study.

2.4. Quality Assessment

Finally, we performed a quality assessment for our included studies. Our quality assessment scoring was based on the guidelines and criteria employed in the tertiary study by Kitchenham et al. [7]. Four quality criteria were formulated based on the criteria defined in [8] to assess the quality of included studies:

- Q1: Are the secondary study's inclusion/ exclusion criteria employed by the included studies described properly?
- Q2: Does the literature search appear to be thorough enough to have covered all related studies?
- Q3: Did the authors assess the quality of their included primary studies?
- Q4: Did the authors adequately describe their selected primary studies?

We evaluated all of our 24 included secondary studies based on the extent of how well they satisfied these quality criteria shown in Table 1. In order to evaluate each of the quality criteria, we applied a point system, in which, YES (Y) = 1 point; Partial (P) = 0.5 point; and No (N) = 0 point. Then we summed the four quality criteria scores. Accordingly, the total score for each assessed study ranged between 0 (very weak) and 4 (very good). Evaluation was done by the first author. The evaluation results were inspected by the other authors and any discrepancies were solved during joint meetings. We note that such quality scoring is vulnerable to subjective assessment and it is sometimes hard to determine the score for some criteria from the selected study (which one may argue is in itself a concern about the selected study). Please note that we did not use the quality score to remove any secondary study from our final selection.

Table 1: Quality assessment criteria [8]

Q	Yes (1.0 point)	Partial (0.5 point)	No (0 point)
Q1	Inclusion /exclusion criteria defined explicitly	Inclusion /exclusion criteria are implicit	inclusion /exclusion criteria not defined and cannot be inferred
Q2	Four or more academic and well-known online databases searched	3-4 online databases searched	Two or less online databases are searched
Q3	Quality criteria defined explicitly	Quality criteria assessed but not defined	No effort has been done to perform quality assessment
Q4	Information presented clearly and can be traced back to individual studies	Information presented in groups or categories, but not linked back to individual studies	Information presented was not referenced

3. Study Results

3.1. Selected Secondary Studies

After applying our search string a total of 7,290 papers, including duplicates, were retrieved: IEEExplore = 113, ACM =1,798, ScienceDirect = 2,717, and SpringerLink = 2,662. Figure 1 shows the conduct of our filtration phases with remaining papers after each step. In the final phase, we performed backward-snowballing on the remaining set of studies. In this phase, we added one paper [14]. A total of 24 systematic secondary studies were finally included in our study.

Table 2 lists the 24 included secondary studies. Full reference details for each study can be found in Appendix A. In Table 2 we summarise their quality score, publication year, publication type as journal or conference, number of papers included by their study, the years covered by the study, and the number of databases they searched. These

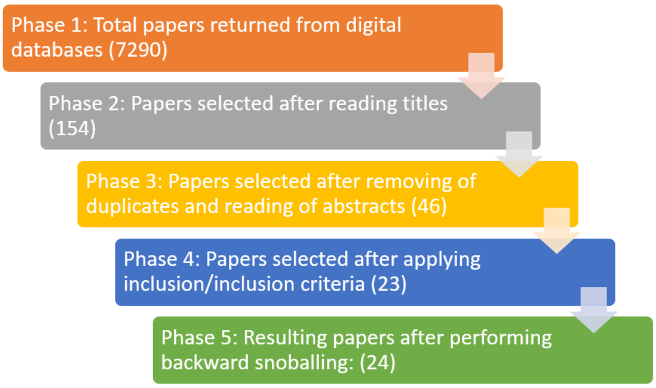


Figure 1: Studies per filtering step

Table 2: Mobile app software engineering SRs published between 2013 and 2022

SR No.	Qual. Score	Pub. year	Paper type	# studies	Years covered	DBs Used
S1	3	2013	Journal	13	2008-2012	4
S2	3.5	2015	Conference	83	2006-2015	4
S3	2.5	2015	Conference	123	1997-2004	4
S4	2	2015	Conference	101	2004-2013	2
S5	3	2016	Journal	79	2005-2015	6
S6	3	2016	Conference	230	2003-2015	4
S7	2.5	2017	Journal	6	2008-2016	4
S8	3	2018	Journal	18	1994-2015	5
S9	4	2018	Journal	21	2004-2018	5
S10	3.5	2018	Journal	20	2004-2018	3
S11	3	2019	Journal	23	2010-2016	6
S12	3	2019	Journal	131	2006-2017	5
S13	4	2019	Journal	49	2005-2018	6
S14	3	2019	Journal	68	2012-2018	4
S15	2	2020	Journal	790	2001-2018	1
S16	3	2021	Journal	87	2008 - 2021	5
S17	3	2021	Conference	39	2011-2021	6
S18	3	2021	Journal	56	2009-2020	4
S19	4	2022	Journal	47	2010-2020	5
S20	2.5	2022	Journal	23	2007-2019	3
S21	4	2018	Journal	75	1999-2018	5
S22	4	2021	Journal	55	2005-2019	4
S23	2.5	2018	Journal	103	2010-2016	4
S24	2.5	2021	Journal	74	2012-2021	4

included systematic secondary studies that cover primary studies of a 27-year period, from 1994 till 2021.

Almost all studies (21 out of 24) applied the SR guidelines proposed by Kitchenham [7] or Kitchenham and Charters [3]. One study, S17, applied the SR guidelines of [10]. Among the 24 systematic secondary studies, thirteen (54%) were systematic mapping studies, and eleven (46%) were systematic reviews, five (5) studies were published in conferences, and seventeen (19) were published in journals. After inspecting studies' objectives, research questions, and data collection and analysis methods, we were able to identify which studies were quantitative, qualitative, or mixed methods. The results show that six (25%) studies (S2, S5, S11, S15, S17, S22) applied a mixed methods approach, and the rest of the studies applied a quantitative approach. The quality score was calculated based on the quality criteria described in Table 1. We did not exclude any study based on the quality score.

The number of online databases used by the systematic secondary studies included are summarised in Figure 2. It can be seen that IEEExplore, ACM Digital Library, Science Direct, and SpringerLink are the most used databases. This is expected as these are where by far the majority of software engineering for mobile papers are indexed. Some studies used secondary indexing databases such as Scopus, Web of Science, Google scholar, DBLP and Cite Seer. Google Scholar and Research Gate are very uncontrolled sources where almost anything can be uploaded or indexed, with quality implications.

4. Answers to our research questions

4.1. RQ1: How many systematic secondary studies on mobile app software engineering were published?

Figure 3 shows the number of included secondary studies publication dates. There is no particular pattern that we observe, although many secondary studies were published in years 2018, 2019 and 2021. The graph in Figure 4a shows the years covered by each systematic secondary study included in their searches. Most studies cover the range early 2000s to late 2010s. This is expected as mobile app development primary studies began appearing in quantity in the early 2000s and have grown significantly in the 2010s with smart phone predominance. During our search for systematic reviews, we did not specify a starting date. Primary studies in the area of effort estimation and automation testing and Model Based Development (MDD) appeared as early as mid 2000s, which is even before the release of the Apple iPhone in 2007. The period of late 2000s saw the emergence of usability evaluation, cloud based testing, functional and non-functional requirements testing, and employing machine learning in bug prediction. A small number of early papers about very early mobile platforms were covered in only a small number of studies e.g. S8, S3, S21. Most secondary studies restricted their primary studies to mid 2000s onwards.

The graph in Figure 4b shows a visualization of the number of studies each secondary study included. This shows that most secondary studies have a relatively modest number of primary studies ranging from low 20s to fewer than 100. Secondary studies S15 and S6 reported very large numbers of primary studies (S15 = 790, S6=230). After checking both studies for reasons why they included so many papers, we found that S15 had a very loose search string ("usability", "mobile application") used in titles, abstracts and keywords. This resulted in a very large

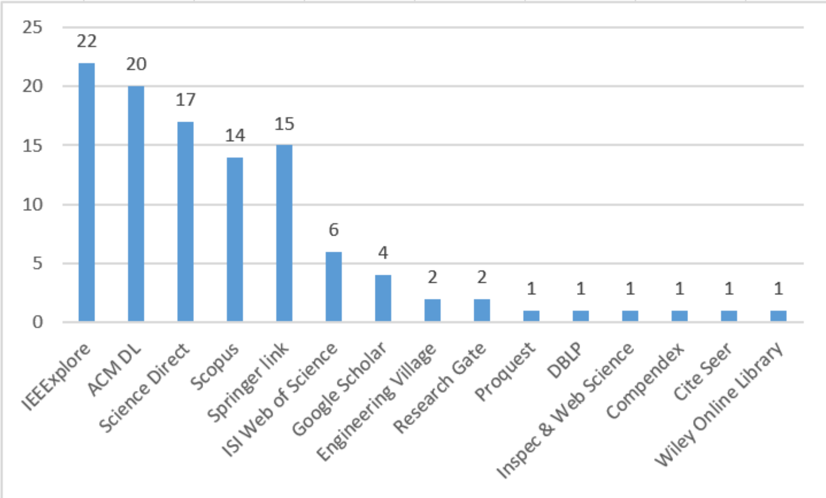


Figure 2: Used online databases by included systematic secondary studies

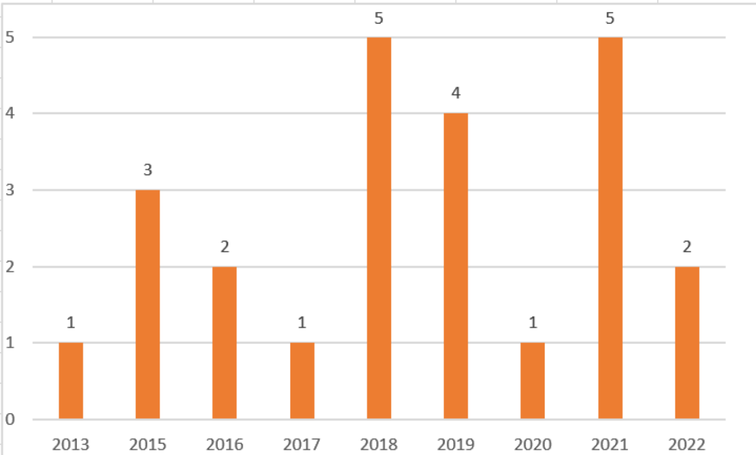


Figure 3: Number of systematic secondary studies published each year.

number of hits. Further, the study did not assign any year range limit and thus covered a wide period of time from 2001 till 2018. It had no filters based on the quality criteria. Their search was run on Scopus, which is an indexing database that covers large number of publishers. S6 is an SMS that uses quite general search criteria ("test", "mobile app", "mobile test") that also ended with large number of studies. The study applied the search strings using four different reference libraries and their initial set of resulting studies counted 3,129 studies. A few of the secondary studies have a very low number of primary studies e.g. S7 (6), S1 (13), S8 (18).

4.2. RQ2: What research topics are being addressed in these secondary studies?

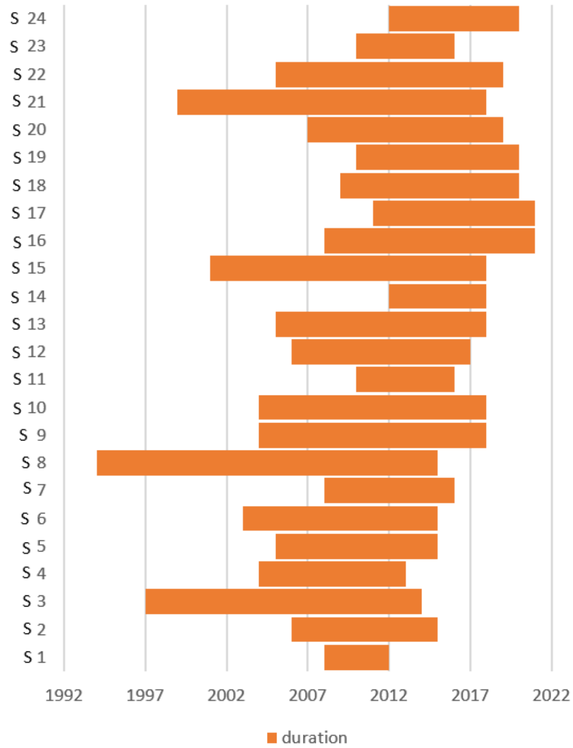
Table 3 shows the key research topics discussed for each included systematic secondary study. Table 4 shows the research topics that are discussed by more than one secondary study. Three secondary studies go beyond the research topics that are related to the software engineering phases and include further important aspects such as relevancy and involvement of industry. For example secondary study S10 assesses the usefulness of its included studies to industry. Additionally, secondary study S12 analysed contributions by industry in testing automation research. Secondary study S18 analyzed the state of industry-academy collaboration. Further, secondary study S22 highlighted that 80% of the Model Driven Development (MDD) approaches were applied to use cases in academia and very few to use cases from the industry. Overall, the research topics that are related to software testing were the dominant topics. Examples on such testing topics are validation methods, testing automation, and usability evaluation.

4.3. RQ3: What phases of mobile app development are addressed in the secondary studies that were included?

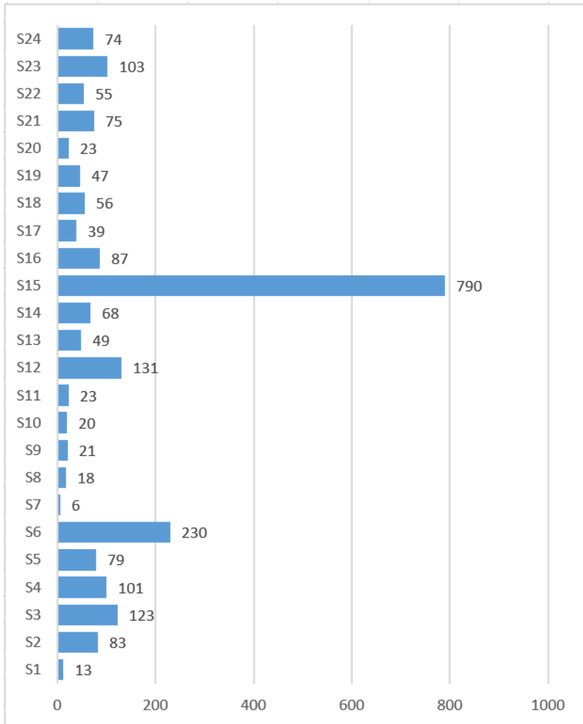
Figure 5 shows the number of secondary studies targeting specific software engineering phases. The majority of

Table 3: Topics covered and SE phases in selected secondary studies

SR No.	Topics addressed	SE Phase
S1	usability heuristics	Design & Testing
S2	automated testing approaches	Testing
S3	verification methods and metrics	Testing
S4	usability evaluation approaches	Design & Testing
S5	Mobile app testing, usability testing, context awareness, automation testing, security	Testing
S6	approaches for quality assurance, testing challenges	Testing
S7	mobile apps on the cloud, model driven development, mobile app portability	Design & Development
S8	mobile app requirements analysis and elicitation	Requirements Eng
S9	effort and size estimation models	Initiation
S10	software development models	Initiation
S11	cloud-based testing, Automation testing, security testing, compatibility testing, functional testing	Testing
S12	testing techniques and tools, evaluation methodologies	Testing
S13	test effort estimation techniques, challenges	Initiation
S14	context-aware test automation	Testing
S15	usability definitions, attributes, measures, evaluation methods	Design & Testing
S16	Android app reliability, context awareness, self-healing, ageing and rejuvenation, runtime event handling.	Testing
S17	software architectural design approaches, quality attributes	Design
S18	nonfunctional requirements testing techniques, automation tools	Testing
S19	software defect prediction using machine learning	Testing
S20	UI interaction design patterns	Design
S21	development and testing effort estimation techniques	Initiation
S22	MDD approaches &tools, code generation, domain model & evaluation	Development
S23	Android automation testing approaches	Testing
S24	Third-party libraries solutions and challenges	Design & Development



(a) Years covered by included systematic secondary studies.



(b) Number of studies covered by systematic secondary studies

Figure 4: Studies and years covered.

studies, thirteen studies, 54%, we grouped under the software engineering phase Testing/Design and Testing; six (6) studies target System Design; four (4 studies) target the Initiation Phase; three studies target the Development

Table 4: Research topics discussed by more than one SR

Topic	Related SRs	Count
Testing automation	S2, S5, S12, S14, S18, S23	6 (25%)
Verification methods	S3, S5, S6, S11, S23	5 (21%)
Usability evaluation approaches	S1, S4, S5, S15	4 (17%)
Context-awareness testing	S5, S14, S16	3 (14%)
effort and size estimation models	S9, S13, S21	3 (13%)
Security testing	S5, S11	2 (8%)
Requirements engineering	S8, S18	2 (8%)
GUI testing	S5, S11, S23	3 (13%)
Model Driven Development	S7, S22	2 (8%)
Development using third-party libraries	S24	1 (4%)

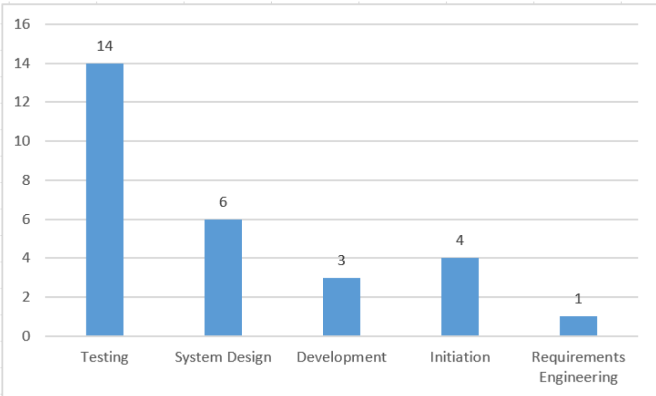


Figure 5: Number of systematic secondary studies per software engineering phase.

Phase; and one study targets Requirements Engineering Phase. As noted in Section 2.3, one study can target more than one software engineering phase.

The majority of secondary studies, 14, (58%) target the "Testing Phase". This indicates that the testing of mobile apps is an active and important research area that received attention among researchers. This is due in large part to the fact that mobile apps are utilized in critical domains such as health, payments, banking, social and even military ones [9, 15], hence ensuring the credibility and reliability of the apps are significantly important. The second group of secondary papers, (6) 25%, targets the "System Design" phase in terms of usability (5 studies) and software architecture design (one study). Obviously, the usability of mobile apps is drawing attention from researchers. This is because traditional web and desktop usability guidelines do not necessarily hold true for mobile apps and that mobile apps are unique in terms of the need to consider physical screen size, technology, user burst attention, interaction styles, power consumption, and connectivity. [15, 16].

Three secondary studies S7, S22, and S24, were classified under the development phase. The secondary study S7 provides in-depth analysis of primary studies that apply Model Driven Development (MDD) in the development of mobile apps that have functions on the cloud and also considers the portability of those apps. Secondary study S22 provides in-depth analysis of MDD techniques and methodologies that were applied in the overall development of mobile apps, as well as, how these techniques were applied and evaluated. The study further identified supporting tools and artifacts for MDD mobile app development and shows that architecture, domain model, and generation of code are the main goals for mobile app MDD. They identified that mobile app productivity, scalability, and productivity are the three quality attributes that can benefit from MDD approaches. Secondary study S24 conducted a SR to summarize and analyse primary studies that are closely related to the area of Android mobile app development using third-party libraries (TPLs). The importance of this area emerges from the fact that even though TPLs can ease the development of Android apps by saving considerable amount of time end effort, they still impose security risks and privacy leaks. The SR first developed a detailed taxonomy covering research objectives of primary studies, target libraries, and analysis methods for TPLs detection. The authors conclude that most TPL detection tools have low resiliency due to code obfuscation. Further, most analysis tools focus on Java TPL analysis but not on native library analysis.

Only one secondary study, S8, targeted the phase of requirements engineering and discussed mobile app requirements elicitation techniques as well as challenges in this phase. This could be due to fact that mobile apps development process has a relatively very short time-to-market cycle, in which developers put more emphasis on design and development than on requirements analysis. We did not find any secondary study that targets other software engineering phases such as the maintenance or evolution phases.

4.4. RQ4: What is the trend in mobile app development topic focus in the secondary studies that were included?

By investigating the timeline and research topics of the included secondary studies, including the number and size of studies (Figures 3, 4a) and topics (Tables 3, 4,) we can see the emergence of primary studies in different trends in mobile app software engineering, shown in Figure 6. The relatively large number of secondary studies in the area of testing and usability evaluation over several time periods shows that these two topics are still very active and on-going research topics. We also need to take into consideration that the field of mobile app software engineering is relatively young (about 15 years).

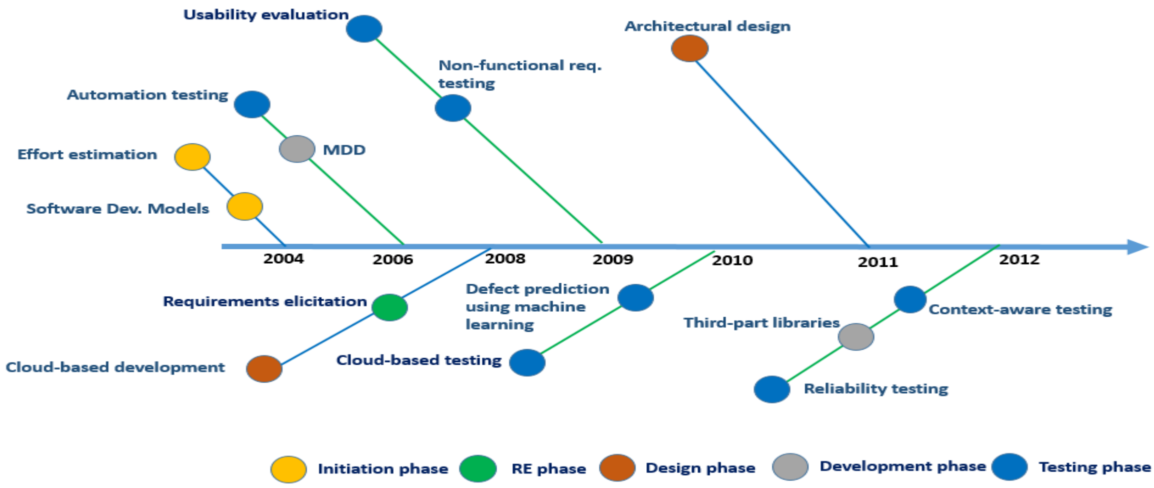


Figure 6: Emergence of primary studies in trends in mobile app software engineering.

The period of the early 2010s and beyond saw the emergence of some key new research topics, including architectural design models, reliability testing, context-aware app testing, and third-party libraries detection and analysis. This reflects that since mobile apps are ubiquitous, even in critical contexts, users are expecting such apps to be more reliable, and equipped with new context-aware features. This indicates that the research in the area of mobile app software engineering is becoming more specific and is targeting new and more current research topics.

4.5. RQ5: What is the quality of these secondary studies?

Table 5 shows the quality scores we computed for each study as well as average scores for each of the quality criteria. Overall, almost all studies (96%) fulfilled the first quality criteria (Q1), i.e. inclusion/exclusion are clearly defined. In many secondary studies (79%), primary studies are adequately described and information are clearly presented and traceable, hence achieved quality criteria 4 (Q4).

In terms of studies coverage (quality criteria 2), most studies (83%) use 4 or more academic databases and the literature search is likely thorough enough to have covered related studies. Out of the four quality criteria, quality criteria 3 (Q3) presented the lowest score (average 0.31 out of 1). Quality criteria 3 refers to whether or not the secondary study defines its own quality score for its own selected primary studies. We found that only 7 out of 24 studies (29%) have actually assessed the quality of their included primary studies. This might indicate that there is still a lack of awareness among researchers on the importance and perhaps reasons for conducting quality assessment when undertaking an SR. Our observation is also similar to Kitchenham et al. [17] that mentioned reports of secondary studies are of variable quality. We urged researchers to refer to the recent SEGRESS guidelines (Software Engineering Guidelines for REporting Secondary Studies) in order to improve quality studies and reporting.

Further, Figure 7 shows the overall distribution of total quality score. The average quality score of all secondary studies is about 3.1 out of 4. The average quality score for journals studies is about 3.13 out of 4. While the average of quality score for conferences studies is slightly lower than journals with a score of 2.8 out of 4. We note that a few studies are questionable in terms of very small numbers of primary studies included (S1, S7), or very large number of primary studies included (S6, S15). It is difficult to draw lessons from the very small numbers of primary studies in these (13 for S1 and only 6 for S7). It is also difficult to extract, analyse and identify detailed lessons from very large numbers of included studies (230 for S6 and an enormous 790 for S15).

Table 5: Quality score for included SRs

SR No.	Type	Q1	Q2	Q3	Q4	Total
1	SR	1	1	0.5	0.5	3
2	Map	0.5	1	1	1	3.5
3	Map	1	1	0	0.5	2.5
4	Map	1	0	0	1	2
5	Map	1	1	0	1	3
6	Map	1	1	0	1	3
7	Map	1	1	0	0.5	2.5
8	SR	1	1	0	1	3
9	SR	1	1	1	1	4
10	SR	1	0.5	1	1	3.5
11	Map	1	1	0	1	3
12	Map	1	1	0	1	3
13	SR	1	1	1	1	4
14	Map	1	1	0	1	3
15	SR	1	0	0	1	2
16	Map	1	1	0	1	3
17	Map	1	1	0	1	3
18	Map	1	1	0	1	3
19	SR	1	1	1	1	4
20	Map	1	0.5	0	1	2.5
21	SR	1	1	1	1	4
22	SR	1	1	1	1	4
23	SR	1	1	0	0.5	2.5
24	SR	1	1	0	0.5	2.5
	Average	0.98	0.87	0.31	0.9	3.06

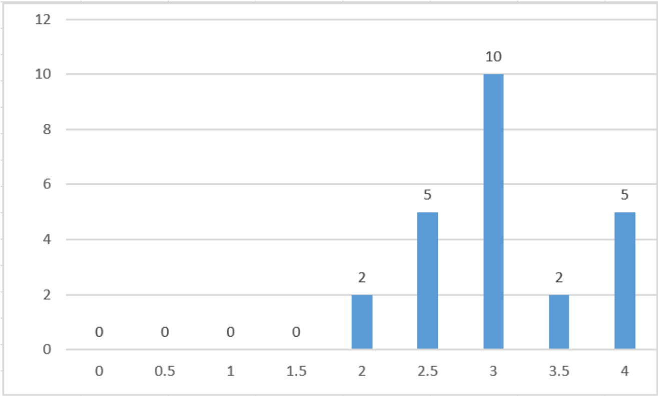


Figure 7: Quality score for systematic secondary studies.

4.6. RQ6: What are key recommendations for future research?

To answer this research question, we looked for future research recommendations that are recurring across multiple systematic reviews, along with common identified limitations and gaps reported in the selected secondary studies. We looked for research gaps/limitations/recommendations common to multiple studies that would evidence the key future research topics that are more strongly needed. After analysing future research recommendations, gaps and limitations in the systematic reviews we included, we found four topics that are mentioned in more than one study, as summarised in Table 6. This shows emergence of the need for more automation, industrial evaluation and testing support in primary studies of mobile app development. It also highlights a gap in evaluation of app usability, both theories and practical techniques and tools.

Table 6: Future research topics highlighted by more than one SR

Research topic	Related SRs	Count
More support for automation testing	S3, S14, S23	3
Evaluating testing methods at real industrial contexts	S5, S6, S12, S16	4
More support for testing of non-functional requirements	S3, S18	2
Need for more developed theory and heuristics for usability evaluation	S1, S15,	2

Secondary studies of S3, S14, S23 stressed the need for more research in the area of automation testing for mobile apps. For instance, S3 recommends more research for automation of test execution, performance testing, and testing in the cloud. In addition to automatic test execution, secondary study S14 suggested the need for automatic test case generation for context-aware apps. S23 highlighted the lack of availability of automation testing tools to the open-source community.

Secondary studies of S5, S6, S12, S16 highlighted the need to conduct more research in real industrial contexts. S5 stressed the need for involving real industrial mobile apps when evaluating new testing methods as opposed of using proof-of-concept apps. S6 and S12 also highlighted the low participation of the industry in the mobile app research area.

Secondary studies S3 and S18 suggested the need for more research to address testing of non-functional requirements of mobile apps. S3 argues that performance testing is still underdeveloped. Further, S18 suggests more research for testing of non-functional requirements of hybrid mobile apps. Finally, S1 and S15 argue more research is needed to define new usability interaction models for mobile apps.

4.7. RQ7: Are any useful recommendations specifically aimed at practitioners made in the secondary studies?

Table 7: Some identified practices and guidelines for practitioners

Practice	Key Phase	SRs
Mobile apps should be fast, easy, and comfortable to use and focus on the most important user tasks	Design	S1
Mobile apps need to be tested on different devices as they can behave differently due to hardware fragmentation	Testing	S2
Mobile app life-cycle events should be tested thoroughly to ensure proper functionality	Testing	S5
User reviews represent a valuable input and should be utilized when evaluating mobile app overall quality	Testing	S6
Interviews and direct discussion with stakeholders are effective requirements elicitation techniques	Requirements	S8
The selection of development, design, and testing tools should be considered at early stages of the project as they have a profound impact on the overall development process	Initiation	S10
Quality of development, healthier communication between developers and clients, proper test planning, and agile methods are important practices when estimating testing effort	Testing	S13
Use of third-party libraries in Android app development saves time and effort. However, practitioners need to be aware that third-party libraries also bring potential security risks and privacy leaks	Design	S24

To answer this question, we had to look into the discussions, conclusions, recommendations, and in some cases we had to look into other parts of the secondary studies since that these recommendations were not explicit. We could not find many practical recommendations, as most of the results of the majority of secondary studies we included are directed towards the research community. We summarise the set of practices identified in Table 7. We have categorised these based on the main software development phase(s) they try to address.

Most of the practical recommendations we identified are around testing practices (4 recommendations). S2 conducts a systematic mapping of automated testing techniques. A common issue with many apps is failure to work on some target platform or handset e.g. with older OS, screen size too small, etc. App testing with a wide variety of likely target end user handsets is highly recommended. S5 provides a systematic mapping of mobile app testing techniques in general. It highlights that most mobile apps exhibit rich event-based life-cycles. A common reported issue is failure of the app during an unexpected event, and rigorous lifecycle event-based testing is recommended before release. S6 is a systematic mapping study focuses on quality assurance of mobile apps. The authors identify that app reviews are a rich source of useful feedback for mobile app developers, and mobile app development teams should proactively use these when refining their apps, and also when considering key app characteristics. S13 is an SLR focusing on test effort estimation. The study identifies several related issues impacting on testing effort estimation, and recommends their careful consideration in a test effort estimation exercise. Key amongst these are test planning, richness of developer and stakeholder communication, and overall app development quality.

Two practical recommendations focus on design. S1 provides a systematic literature review on a range of usability heuristics for mobile apps. The study identifies some key heuristics that need to be focused on during app development in order to achieve suitable usability measurements. These include task prioritisation and ease of use around speed and efficiency. S24 provides a systematic literature review focusing on third-party library usage in mobile app design and implementation. The study makes a number of recommendations, including highlighting the benefits of such library usage. However, the authors also emphasise the need for practitioners to carefully consider various potential negative consequences, particularly around data privacy and security, as some behaviours of such third party libraries are difficult to discern.

One each project initiation and requirements elicitation practical recommendation was highlighted. S10 provides an SLR of process models used in app development. Key practical recommendations relate to careful choice of tool choices, including testing, IDE and others, due to their critical impact on the development process. S8 presents a systematic study of requirements techniques for mobile app development. The need to closely involve key stakeholders in requirements elicitation is emphasised. No clear best practice guidelines were identified around mobile app development management or app evolution.

5. Related Work

Several non-SR/SMS secondary studies in the area of mobile app software engineering are available in the literature. Following a mixed-methods approach based on grounded theory and a survey, Joorabchi et al. [18] investigated challenges and state-of-practice of mobile app development in industrial contexts. Their survey highlights several interesting challenges faced by mobile app developers such as developing apps for different platforms, lack of tools to support analysis, testing, and monitoring of mobile phones. Their survey reveals that functional testing is the dominant testing practice by developers and that there is a lack of automation testing frameworks. Another survey by Dalmasso et al. [19] also targets the challenge of developing mobile apps for different platforms. Their survey presents decision criteria that can assist in choosing the correct cross-platform framework, such as quality of User Experience (UX), app development cost, access of built in features, security of app, supportability, ease of updating, and time-to-market. This survey also presents a comparison between several cross-platform frameworks based on resources consumption and suggests further research directions.

A literature survey by Corral et al. [20] investigated mobile apps software assurance practices and assesses their contribution and success. Their survey investigates different approaches targeting the needs and peculiarities of the mobile app development area. Further, their survey stresses the need to adapt general-purpose practices to the specific and changing needs of mobile app ecosystem. A survey by Martin et al. [21] investigates studies in the area of App Store Analysis to identify trends and behaviour that affect development teams. Their survey reveals the emergence of several key sub-fields as well as techniques and applications in this area. Additionally, it highlights several future research directions such as leveraging the business, customer, and technical aspects of mobile apps. The survey also mentions some problems faced by researchers such as restrictions on data availability imposed by mobile app stores.

Nagappan et al. [22] present an investigation into research trends in different phases of the mobile app development process. Several fields are investigated such as cross-platform development and testing challenges. Further, the survey covers research trends for extraction of suggestions and complaints from user reviews to gather requirements and solutions to lower energy consumption. Francese et al. [15] performed qualitative research to explore and understand the state of practice of mobile app development and management practices at industrial contexts. Their study results show that agile methods are commonly adopted during mobile app development projects, and that native mobile development is preferred over cross-platform solutions due to their limitation. Additionally, they find that most automated testing tools for mobile apps still have limited capabilities.

Ahmad et al. [23] performed an empirical investigation to identify and analyse challenges faced by mobile development companies when choosing between different mobile platforms. In their study, the authors argue that developers should be fully aware about development challenges when choosing between native, web, and cross-platform mobile technologies. Results of their study show that, for native mobile app development, fragmentation, code reuse, low tool support for testing, and change management are common challenges faced by development teams. For the web mobile development, other challenges exist such as user experience and testing. Regarding cross-platform mobile app development, common challenges faced by development teams are automation testing, fragmentation, and compatibility among others. Asfour et al. [24] conducted a qualitative investigation to explore and understand how agile methods are adopted for mobile app development at industrial contexts. Their study shows that agile methods are frequently adopted for mobile app development since they are better fit for rapid nature of such apps and quick adaptation for changes in requirements and technologies. Further, the study highlights those agile practices that suit mobile app development projects, such as time boxing, internal and external release, and on-site customer.

6. Limitations

The first author performed the literature search using commonly used digital databases and performed selection based on the inclusion/exclusion criteria. Like other systematic reviews, the construction of the search terms and execution of searches on databases with very different query formats and query operation raises the risk of missing some studies. To minimize the possible threat of study selection bias and missing important studies, we used multiple strategies which are described below. First, we evaluated our search string against a predefined set of secondary studies that we were aware of. Second, we executed several pilot searches through which the search string was systematically improved in order to retrieve as many relevant papers as possible. The second author performed cross-checking of the searches and terms, and on a set of randomly selected studies to ensure the validity of study search and selection. Finally, we performed backward snowballing to ensure wider coverage of potential relevant studies.

Informal literature reviews i.e. non-systematic surveys were excluded to maintain high reliability of the results. Accordingly, there may be topics in mobile app SE that have informal literature reviews but were not included in this study. Additionally, we excluded secondary studies that are not directly related to the area of mobile app software engineering, such as studies that focus on the use or application of mobile apps in certain domains. We do admit that some of these excluded studies may address certain development practices or issues. We also excluded secondary studies that are not systematic or their methodology was not convincing. Although this process was checked by more than one author, there is a possibility that the process may be biased due to subjective interpretation. Further, there do exist other mobile app SE systematic secondary studies that are written in non-English languages and that we excluded, namely [25, 26, 27, 28].

Data extraction was done by the first author and was cross-validated by second and the third authors. Regular meetings were held to resolve disagreements in the data extraction. In the descriptive analysis, we applied thematic coding to identify secondary studies main research topic to answer RQ2. During our thematic analysis we focused on the most dominant themes of each secondary study. Accordingly, less dominant themes that may have been present in the secondary study that were not covered. Finally, it should be noted that the same primary studies may have appeared in more than one secondary study. This may have biased our overall classification by making issues raised by the shared primary studies look more important.

7. Conclusion

In this research, we conducted a tertiary study of systematic reviews and systematic mapping studies in the area of mobile app software engineering. We followed the guidelines for conducting tertiary studies presented by [7]. After applying several filtration steps, we identified 24 secondary studies, among which 13 were SMSs and 11 SRs. The included secondary studies cover numerous research areas, such as effort estimation, quality assurance and test automation, usability, software development models, requirements engineering, architectural design, defect prediction using machine learning, cloud-based development, and model driven development. An in-depth discussion and analysis is provided about focus, quality, coverage, scope, and key recommendations for the secondary studies included. Further, we identified which phases of mobile app development process that were addressed by included secondary studies and we concluded that most of the studies address the phases of testing and design, and very few secondary studies addressing requirements engineering and development. No secondary studies were found focusing on the "evolution" software development phase.

When investigating the trend in mobile app development focus, we found that the trend in recent years is moving towards more specific software engineering topics such as architectural design, testing of non-functional requirements, and context-aware testing. Based on the analysis and results in this tertiary study, we highlight areas and topics that can benefit from further research, namely mobile app evolution and requirements engineering. Additionally, since cross-platform mobile app development is now more common [29, 30], both secondary and primary studies in this area can also be very beneficial.

Acknowledgements

Grundy is supported by ARC Laureate Fellowship FL190100035.

Appendix A. List of secondary studies and full citation

- **S1:** Salazar, L. H. A., Lacerda, T., Nunes, J. V., & von Wangenheim, C. G. (2013). A systematic literature review on usability heuristics for mobile phones. International Journal of Mobile Human Computer Interaction (IJMHCI), 5(2), 50-61.
- **S2:** Méndez Porras, A., Quesada López, C. U., & Jenkins Coronas, M. (2015). Automated testing of mobile applications: A systematic map and review. Proceedings of the 28th Ibero-American Conference on Software Engineering (CIBSE). URP,SPC,UCSP, Lima, Peru, 1-14.
- **S3:** Sahinoglu, M., Incki, K., & Aktas, M. S. (2015, June). Mobile application verification: a systematic mapping study. In International Conference on Computational Science and Its Applications (pp. 147-163). Springer, Cham.
- **S4:** Reis, R. A. C., Fontão, A. D. L., Gomes, L. L., & Dias-Neto, A. C. (2015). Usability evaluation approaches for (ubiquitous) mobile applications: a systematic mapping study. In Proceedings of International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies. UBICOMM.
- **S5:** Zein, S., Salleh, N., & Grundy, J. (2016). A systematic mapping study of mobile application testing techniques. Journal of Systems and Software, 117, 334-356.
- **S6:** Holl, K., & Elberzhager, F. (2016, December). Quality assurance of mobile applications: A systematic mapping study. In Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia (pp. 101-113).
- **S7:** Sanchiz, E., González, M., Aquino, N., & Cernuzzi, L. (2017). Development of mobile applications with functions in the cloud through the model driven approach: a systematic mapping study. CLEI Electron. J., 20(3).
- **S8:** Dar, H., Lali, M. I., Ashraf, H., Ramzan, M., Amjad, T., & Shahzad, B. (2018). A systematic study on software requirements elicitation techniques and its challenges in mobile application development. IEEE Access, 6, 63859-63867.
- **S9:** Altaleb, A., & Gravell, A. (2018). Effort estimation across Mobile app platforms using agile processes: a systematic literature review. Journal of Software, 13(4), 242.
- **S10:** Jabangwe, R., Edison, H., & Duc, A. N. (2018). Software engineering process models for mobile app development: A systematic literature review. Journal of Systems and Software, 145, 98-111.
- **S11:** Yaâ, B. I., Salleh, N., Nordin, A., Idris, N. B., Abas, H., & Alwan, A. A. (2019). A systematic mapping study on cloud-based mobile application testing. Journal of Information and Communication Technology, 18(4), 485-527.
- **S12:** Tramontana, P., Amalfitano, D., Amatucci, N., & Fasolino, A. R. (2019). Automated functional testing of mobile applications: a systematic mapping study. Software Quality Journal, 27(1), 149-201.
- **S13:** Kaur, A., & Kaur, K. (2019). Investigation on test effort estimation of mobile applications: Systematic literature review and survey. Information and Software technology, 110, 56-77.
- **S14:** Almeida, D. R., Machado, P. D., & Andrade, W. L. (2019). Testing tools for Android context-aware applications: a systematic mapping. Journal of the Brazilian Computer Society, 25(1), 1-22.
- **S15:** Weichbroth, P. (2020). Usability of mobile applications: a systematic literature study. IEEE Access, 8, 55563-55577.
- **S16:** Wimalasooriya, C., Licorish, S. A., da Costa, D. A., & MacDonell, S. G. (2021). A systematic mapping study addressing the reliability of mobile applications: The need to move beyond testing reliability. Journal of Systems and Software, 111166.
- **S17:** Del Carpio, A. F., & Vera, Y. P. (2021, August). Architectural Approaches for Emerging Technologies-based Mobile Apps–a Systematic Mapping Study. In 2021 IEEE 12th International Conference on Software Engineering and Service Science (ICSESS) (pp. 16-23). IEEE.
- **S18:** Júnior, M. C., Amalfitano, D., Garcés, L., Fasolino, A. R., Andrade, S. A., & Delamaro, M. (2022). Dynamic Testing Techniques of Non-functional Requirements in Mobile Apps: A Systematic Mapping Study. ACM Computing Surveys (CSUR), 54(10s), 1-38.
- **S19:** Jorayeva, M., Akbulut, A., Catal, C., & Mishra, A. (2022). Machine learning-based software defect prediction for mobile applications: A systematic literature review. Sensors, 22(7), 2551.
- **S20:** da Silva, L. F., Parreira Junior, P. A., & Freire, A. P. (2022). Mobile User Interaction Design Patterns: A Systematic Mapping Study. Information, 13(5), 236.
- **S21:** Kaur, A., & Kaur, K. (2018). Systematic literature review of mobile application development and testing effort estimation. Journal of King Saud University-Computer and Information Sciences, Volume 34, Issue 2, February 2022, Pages 1-15.
- **S22:** Md. Shamsujjoha, John Grundy, Li Li, Hourieh Khalajzadeh, Qinghua Lu. (2021). Developing Mobile Applications Via Model Driven Development: A Systematic Literature Review, Information and Software Technology, Volume 140, 2021.
- **S23:** Kong, Pingfan, Li Li, Jun Gao, Kui Liu, Tegawendé F. Bissyandé, and Jacques Klein. Automated testing of android apps: A systematic literature review. IEEE Transactions on Reliability 68, no. 1 (2018): 45-66.
- **S24:** Zhan, Xian, Tianming Liu, Lingling Fan, Li Li, Sen Chen, Xiapu Luo, and Yang Liu. Research on third-party libraries in android apps: A taxonomy and systematic literature review. IEEE Transactions on Software Engineering, vol. 48, no. 10 (2021): 4181 - 4213.

References

[1] Statista, Number of smartphone subscriptions worldwide, `https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/`, accessed November 6, 2022 (2022).

[2] Mobile app download statistics and usage statistics, `https://buildfire.com/app-statistics/`, accessed November 6, 2022 (2022).

[3] B. Kitchenham, S. Charters, Procedures for performing systematic literature reviews in software engineering, Keele University & Durham University, UK 33 (2007) 28.

[4] H. Kondylakis, D. G. Katehakis, A. Kouroubali, F. Logothetidis, A. Triantafyllidis, I. Kalamaras, K. Votis, D. Tzovaras, et al., Covid-19 mobile apps: a systematic review of the literature, Journal of medical Internet research 22 (12) (2020) e23170.

[5] R. Nussbaum, C. Kelly, E. Quinby, A. Mac, B. Parmanto, B. E. Dicianno, Systematic review of mobile health applications in rehabilitation, Archives of physical medicine and rehabilitation 100 (1) (2019) 115–127.

[6] P. P. Brzan, E. Rotman, M. Pajnkihar, P. Klanjsek, Mobile applications for control and self management of diabetes: a systematic review, Journal of medical systems 40 (9) (2016) 1–10.

[7] B. Kitchenham, R. Pretorius, D. Budgen, O. P. Brereton, M. Turner, M. Niazi, S. Linkman, Systematic literature reviews in software engineering–a tertiary study, Information and software technology 52 (8) (2010) 792–805.

[8] R. Hoda, N. Salleh, J. Grundy, H. M. Tee, Systematic literature reviews in agile software development: A tertiary study, Information and software technology 85 (2017) 60–70.

[9] S. Zein, N. Salleh, J. Grundy, A systematic mapping study of mobile application testing techniques, Journal of Systems and Software 117 (2016) 334–356.

[10] K. Petersen, S. Vakkalanka, L. Kuzniarz, Guidelines for conducting systematic mapping studies in software engineering: An update, Information and software technology 64 (2015) 1–18.

[11] Y. Wu, X. Yao, G. Vespasiani, A. Nicolucci, Y. Dong, J. Kwong, L. Li, X. Sun, H. Tian, S. Li, et al., Mobile app-based interventions to support diabetes self-management: a systematic review of randomized controlled trials to identify functions associated with glycemic efficacy, JMIR mHealth and uHealth 5 (3) (2017) e6522.

[12] I. Sommerville, Engineering software products, Pearson London, 2020.

[13] V. Braun, V. Clarke, Using thematic analysis in psychology, Qualitative research in psychology 3 (2) (2006) 77–101.

[14] X. Zhan, T. Liu, L. Fan, L. Li, S. Chen, X. Luo, Y. Liu, Research on third-party libraries in android apps: A taxonomy and systematic literature review, IEEE Transactions on Software Engineering.

[15] R. Francese, C. Gravino, M. Risi, G. Scanniello, G. Tortora, Mobile app development and management: results from a qualitative investigation, in: 2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft), IEEE, 2017, pp. 133–143.

[16] J. Rochat, F. Ehrler, J. N. Siebert, A. Ricci, V. G. Ruiz, C. Lovis, et al., Usability testing of a patient-centered mobile health app for supporting and guiding the pediatric emergency department patient journey: Mixed methods study, JMIR pediatrics and parenting 5 (1) (2022) e25540.

[17] B. Kitchenham, L. Madeyski, D. Budgen, Segress: Software engineering guidelines for reporting secondary studies, IEEE Transactions on Software Engineering 49 (3) (2022) 1273–1298.

[18] M. E. Joorabchi, A. Mesbah, P. Kruchten, Real challenges in mobile app development, in: 2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, IEEE, 2013, pp. 15–24.

[19] I. Dalmasso, S. K. Datta, C. Bonnet, N. Nikaein, Survey, comparison and evaluation of cross platform mobile application development tools, in: 2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC), IEEE, 2013, pp. 323–328.

[20] L. Corral, A. Sillitti, G. Succi, Software assurance practices for mobile applications, Computing 97 (10) (2015) 1001–1022.

[21] W. Martin, F. Sarro, Y. Jia, Y. Zhang, M. Harman, A survey of app store analysis for software engineering, IEEE transactions on software engineering 43 (9) (2016) 817–847.

[22] M. Nagappan, E. Shihab, Future trends in software engineering research for mobile apps, in: 2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER), Vol. 5, IEEE, 2016, pp. 21–32.

[23] A. Ahmad, K. Li, C. Feng, S. M. Asim, A. Yousif, S. Ge, An empirical study of investigating mobile applications development challenges, IEEE Access 6 (2018) 17711–17728.

[24] A. Asfour, S. Zain, N. Salleh, J. Grundy, Exploring agile mobile app development in industrial contexts: A qualitative study, International Journal of Technology in Education and Science 3 (1) (2019) 29–46.

[25] T. F. Bernardes, M. Y. Miyake, Cross-platform mobile development approaches: A systematic review, IEEE Latin America Transactions 14 (4) (2016) 1892–1898.

[26] I. Galeano, J. Casariego, M. Merln, M. Gonzalez, Mobile application development approaches: A systematic mapping study, in: 2016 35th International Conference of the Chilean Computer Science Society (SCCC), IEEE, 2016, pp. 1–12.

[27] E. Sanchiz, M. González, N. Aquino, L. Cernuzzi, Mobile cloud applications development through the model driven approach: A systematic mapping study, in: 2016 XLII Latin American Computing Conference (CLEI), IEEE, 2016, pp. 1–10.

[28] M. Norberto, L. Gaedicke, M. Bernardino, G. Legramante, F. P. Basso, E. M. Rodrigues, Performance testing in mobile application: a systematic literature map, in: Proceedings of the XVIII Brazilian Symposium on Software Quality, 2019, pp. 99–108.

[29] T. Zohud, S. Zein, Cross-platform mobile app development in industry: A multiple case-study, International Journal of Computing 20 (1) (2021) 46–54.

[30] H. Tufail, F. Azam, M. W. Anwar, I. Qasim, Model-driven development of mobile applications: A systematic literature review, in: 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), IEEE, 2018, pp. 1165–1171.