Protocol for a Systematic mapping study on

Software Engineering Techniques for Statically Analyzing Mobile Apps: Research Trends, Characteristics, and Potential for Industrial Adoption

Marco Autili (University of L'Aquila)

Ivano Malavolta (Vrije Universiteit Amsterdam)

Alexander Perucci (University of L'Aquila)

Gian Luca Scoccia (Gran Sasso Science Institute)

Roberto Verdecchia (Gran Sasso Science Institute)

Document style © Yuriy Zachia Lun

PROTOCOL FOR A SYSTEMATIC MAPPING STUDY ON

Software Engineering Techniques for Statically Analyzing Mobile Apps: Research Trends, Characteristics, and Potential for Industrial Adoption

ABSTRACT

This document describes the review protocol of a systematic mapping study aimed at identifying, evaluating and classifying characteristics, trends and potential for industrial adoption of existing research in static analysis of mobile apps.

KEYWORDS

Software Engineering, Static Analysis, Mobile apps, Systematic mapping study.

Contents

List of Figures

List of Tables

1 Background and rationale

The development of mobile apps is exponentially growing since the establishment of a number of app stores and market places from where to download and install them, e.g., Apple Store, Google Play, Windows Phone Marketplace. As an indicator of this situation, today the total activity on smartphones and tablets accounts for an incredible 66% of the time spent on digital media in the United States [1].

Mobile apps consist of executable files that are downloaded directly to the end user's device and stored locally. Mobile apps are developed atop the services provided by their underlying mobile platform (e.g., Android). Those services are exposed via a dedicated Application Programming Interface (API) with methods related to communication and messaging, graphics, security, etc.

Mobile apps are distributed via dedicated *app stores*, such as Google Play for Android apps, and the Apple app store for iOS apps. As of today, these stores make available millions of mobile apps of different categories to millions of people, who will use them for their everyday activities like purchasing products, messaging, etc. [1]. Clearly, this is a highly competitive business in which even the smallest error may have a tremendous financial impact. Indeed, revenue and profit of a mobile app is often proportional to the number of its users , which may either positively rate the app or abandon it (and possibly go on social media and complain); this implies that improving the level of users satisfaction is fundamental both to keep existing users active and to attract new ones.

To improve user satisfaction, *static analysis of mobile apps* can be a valuable instrument for both (i) app developers, who can use static analysis to quickly get non-trivial insights about their mobile app (e.g., subtle security issues, energy hotspots due to some programming antipattern, inefficient use of hardware sensors) and (ii) app store moderators like Google and Apple, that can use static analysis for systematically assessing the level of quality of their distributed apps, possibly identifying those apps with an unacceptable level of quality (e.g., apps with well-known security flaws, apps asking for suspicious permissions, apps with strong energy inefficiencies). In this Systematic Mapping Study (SMS) we aim at precisely characterizing existing research on static analysis of mobile apps. This goal is achieved by identifying, classifying, and evaluating the current state of the art on static analysis methods and techniques (referred to also as approaches) of mobile apps from different perspectives, such as the focus of research, potential of industrial adoption, publication trends, research strategies.

1.1 Existing systematic studies on the topic

Literature reviews, surveys and mapping studies on either static analysis approaches or analysis methodologies and techniques applied to mobile apps that can be considered as research related to our study.

Based on our knowledge, we found no systematic mapping study (SMS) and only one systematic literature review (SLR) on the specific topic of static analysis of mobile apps. Thus, in the following, we first discuss in more detail the SLR, the one reported in [2], which is a valuable and solid work study closely related to ours. Then, we discuss those works in the literature that, although having different scopes and objectives, can be related to our research.

Similarly to our SMS, the SLR in [2] reviewed publications on approaches involving the use of static analysis for Android apps only, through vertical analysis only; in addition to Android, our study considered also other platforms, and employed both vertical analysis and horizontal analysis. As per the search strategy, the main difference is that we performed a manual search of top venues for SE and programming languages, followed by backward snowballing and then forward snowballing; in [2], the authors performed automatic search followed by manual search of top venues for SE, programming languages, security and privacy, and then authors' self-check followed by backward snowballing. Concerning the selections criteria, we considered only peer reviewed work, by excluding studies in the form of editorials and tutorial, as well as short and poster papers, secondary or tertiary studies. In [2], only short papers were excluded. Moreover, differently from them, we accounted for the existence of some kind of evaluation together with the availability of an implementation. As a result, they collected 124 research papers, in the timespan 2011-2015; we have a better coverage made of 134 primary studies in the timespan 2007-2017.

Importantly, in [2], the authors do not consider the potential for industrial adoption of existing research on static analysis of mobile apps, as we do through our research question RQ3. This is a substantial difference that permitted us to identify in the state of the art those approaches to static analysis of mobile apps that are ready for technological transfer and industrial adoption. Another profitable difference is in the nature of the study, SLR versus SMS, and in the target audience. As already introduced, in our SMS we target both researchers and practitioners, such as app developers, who are interested in selecting/choosing existing static analysis approaches, and want to critically understand what they offer and how, in order to opt for their adoption or possible industrial transfer. The SLR in [2] more specifically targets researchers and practitioners that want to propose a new approach to static analysis or to extend existing ones. In this sense, we believe that our work and the work in [2] complement one another, and together they constitute a valuable asset to the academic and industrial world in the wide spectrum of static analysis.

In [3], a survey about static analysis and model checking approaches for searching patterns and vulnerabilities within a software system is reported. The authors examine the proposed algorithms and their effectiveness in finding bugs. A peculiarity of this research is the comparison between static analysis algorithms and mathematical logic languages for model checking.

In [4], the authors report on a survey about static analysis for identifying security issues and vulnerabilities in software systems in general (not specific to mobile apps). For each type of security vulnerability, the authors present both relevant studies and the implementation details of the used static analysis algorithms.

A systematic mapping study is reported in [5]. The study was conducted for classifying and analysing approaches that combine different static and dynamic quality assurance techniques. The study includes a discussion about reported effects, characteristics, and constraints of the various existing techniques.

A literature review about mobile usability models can be found in [6], as a means for validating a specific usability model. Among the main results, from this literature review it emerges that usability is usually measured in terms of three key indicators, namely, effectiveness, efficiency and satisfaction.

Even if some of the above mentioned works are about static analysis, none of them is specifically focussed on the static analysis of mobile apps, and none of them is a systematic literature review.

1.2 The need for an SMS on static analysis of mobile apps

With this systematic mapping study we aim to characterize existing research on methods and techniques for static analysis of mobile apps. Our study will help researchers and practitioners in identifying the focus, limitations, gaps, and trends of existing research on static analysis of mobile apps. Also, we will assess the potential of research on static analysis of mobile apps with a focus on how its results can be transferred and adopted in industrial projects. By knowing the potential of methods and techniques for static analysis of mobile apps, researchers and practitioners will have a reference framework for better understanding and possibly adopting static analysis of mobile apps, respectively.

2 Research Process

This research will be carried out by following the process shown in Figure 1; it can be divided into three main phases, which are well-established when it comes to systematic literature studies [7, 8]: planning, conducting, and documenting.

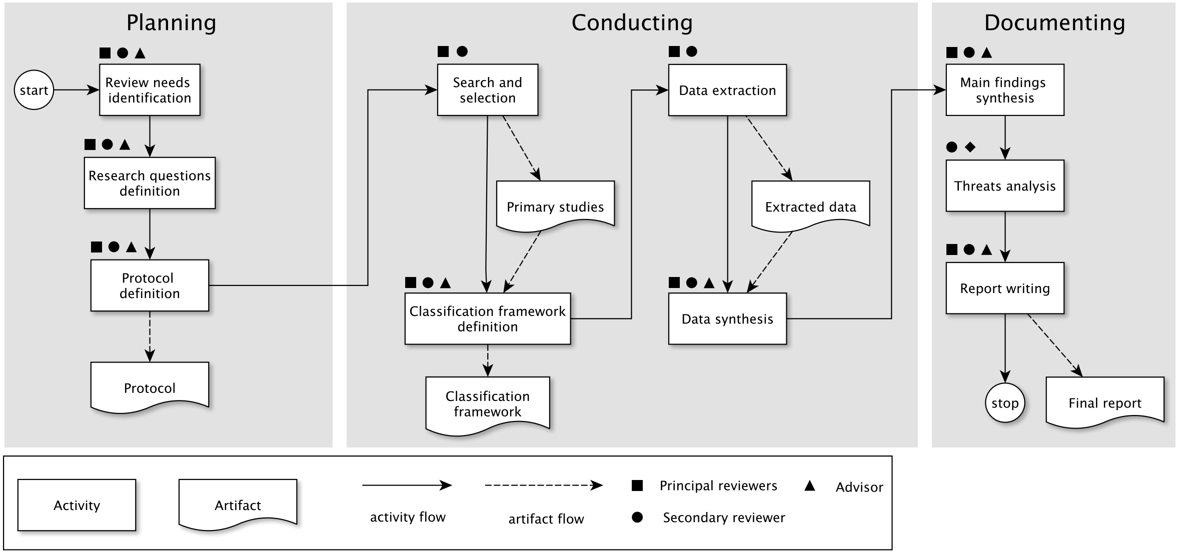

Figure 1: Overview of the whole review process

Each phase has a number of output artefacts, e.g., the planning phase produces the review protocol described in this document. In the following we will go through each phase of the process, highlighting main activities and produced artefacts.

2.1 Planning

This phase aims at (i) establishing the need for performing a review on static analysis of mobile apps (see Section 1.2), (ii) identifying the main research questions (see Section 3), and (iii) defining the protocol to be followed by the involved researchers. The output of the planning phase is a detailed protocol (i.e., this document).

2.1.1 Conducting

In this phase we will perform the mapping study by following all the steps previously defined. More specifically, we will carry out the following activities:

- *Search and selection*: we will perform a combination of manual search and backward and forward snowballing for identifying the set of potentially relevant research articles on static analysis methods and techniques of mobile apps. Then, identified candidate entries will be filtered according to a selection criteria in order to obtain the final list of primary studies to be considered in later activities of the review. Section 4 describes in details the search and selection process.

- *Classification framework definition*: in this activity we will define the set of parameters that will be used to compare primary studies.

- *Data extraction*: in this activity we will go into the details of each primary study, and we will fill a corresponding data extraction form. Filled forms will be collected and aggregated in order to be ready to be analyzed during the next activity. More details about this activity are presented in Section 5.

- *Data synthesis*: this activity will focus on a comprehensive analysis and summary of the data extracted in the previous activity. The main goal of this activity is to elaborate on the extracted data in order to answer each research question of the mapping study (see Section 3).

This activity will involve both quantitative and qualitative analysis of the extracted data. The details about this activity are presented in Section 6.

2.1.2 Documenting

The main activities performed in this phase are: (i) a thorough elaboration of the data extracted in the previous phase with the main aim of setting the obtained results in their context, (ii) the discussion of possible threats to validity, specially to the ones identified during the definition of the review protocol (in this activity new threats to validity may emerge too), and (iii) the writing of a final report describing the performed mapping study. Firstly, the produced report will be evaluated by a set of experts, then it will be submitted to an academic journal, thus undergoing a peer reviewed evaluation by the community too.

2.2 Team

Five researchers will carry out this study, each of them with a specific role within the research team:

- *Principal researcher*: Roberto Verdecchia, Gian Luca Scoccia and Alexander Perucci, PhD students. They will be part of all the activities, i.e., planning the study, conducting it, and reporting;
- *Research methodologist*: Ivano Malavolta, assistant professor with expertise in empirical software engineering, software architecture, and systematic literature reviews; he is mainly involved in (i) the planning phase of the study, and (ii) supporting the principal researchers during the whole study, e.g., by reviewing the data extraction form, selected primary studies, extracted data, produced reports, etc.;
- *Advisor*: Marco Autili, assistant professor with many-years expertise in software engineering methods applied to the modelling, verification, analysis and automatic synthesis of complex distributed systems, and application of context-oriented programming and analysis techniques to the development of (adaptable) mobile applications. He takes final decisions on conflicts and methodological options to 'avoid endless discussions' [9], and supports the other researchers during data and findings synthesis activities.

3 Research questions

This study aims at characterizing the current state of the art for understanding what we know about scientific research on static analysis of mobile apps. The results of this study are targeted to both (i) researchers willing to further contribute to this research area, and (ii) practitioners willing to understand existing research on static analysis approaches of mobile apps and thereby to be able to adopt those solutions that better fit with their business goals. More formally, we formulate the goal of this study by using the Goal-Question-Metric perspectives (i.e., purpose, issue, object, viewpoint [10]). Table 1 shows the result of the above mentioned formulation.

Purpose	Identify, classify, and evaluate
Issue	trends, characteristics and potential for industrial adoption
Object	of existing research in static analysis of mobile apps
Viewpoint	from a researcher's and practitioner's point of view.

Table 1: Goal of this research

This abstract goal can be refined into the following research questions (for each research question we also provide the rationale for it being part of this study):

RQ1: *What are the **research trends** on static analysis of mobile apps?*

Rationale: academic research is a dynamic ecosystem, where a multitude of researchers and research groups investigate on specific scientific problems over time with different degrees of independence and different methodologies. By answering this research question we aim at characterizing the ongoing trends of scientific interest on static analysis approaches of mobile apps, the relevant venues where academics are publishing their results on the topic and their contribution types; depending on the number of primary studies, trends will be assessed over the years.

RQ2: *What are the **characteristics** of existing approaches for static analysis of mobile apps?*

Rationale: static analysis of mobile apps is a multi-faceted research topic, where researchers can focus on very different aspects (e.g., energy consumption, efficient use of computational resources, security flaws, privacy violations, performance issues, reliability), applying very different research methodologies (e.g., industrial case studies, empirical evaluations, feasibility studies), providing different types of contributions (e.g., tools that allow for automatic or semi-automatic analysis process, methods or a techniques to analyze a specific aspect, specification languages for describing semantic characteristics, etc.). By answering this research question we aim at providing (i) a solid foundation for analysing and classifying existing (and future) research on static analysis methods and techniques of mobile apps, and (ii) an understanding of current research trends and gaps in the state of the art on static analysis of mobile apps.

RQ3: *What is the **potential for industrial adoption** of existing research on static analysis of mobile apps?*

Rationale: while it is well known that mobile apps have their roots in industry, many research groups focus on them from an academic perspective. Therefore, it is natural to ask ourselves how the produced research findings and contributions can be actually transferred back to industry. By answering this research question we aim at assessing how and if the current state of the art on static analysis of mobile apps is ready to be adopted in industry.

Identified research questions will drive the whole study, with a special influence on (i) search and selection of primary studies, (ii) data extraction, and (iii) data analysis.

4 Search and selection process

The main goal of our search and selection process is to retrieve a comprehensive set of research studies that are relevant and representative enough for the topic being considered. More specifically, it is fundamental to achieve a good trade-off between the coverage of existing research on the topic considered, and to have a manageable number of studies to be analysed. In order to achieve the above mentioned trade-off, our search strategy consists of two complementary methods: a manual inspection of the researches published in thetop-level software engineering venues and a manual snowballing process on the identified researches. Our search and selection process has been designed as a multi-stage process in order to have full control on the number and characteristics of the studies being either selected or excluded during the various stages. In the following we detail each step of our search and selection process.

The search strategy is divided into two subsequent and complementary steps. The first step is carried out by manually inspecting all the publications of the top-level software engineering venues. The papers identified through this first step will then be subsequently utilized as input for a backward and forward snowballing[1] process [11].

[1] Inspection of the studies referenced by a paper (*backward snowballing*) and of the studies referencing it (*forward snowballing*)

From the results of a preliminary study [12] the research field of mobile static analysis resulted to be heterogeneous and many keywords, e.g., “program analysis”, resulted to be profoundly overloaded, leading to imprecise and inaccurate automatic search results. In order to prevent biases associated to automatic searches, we adopt two complementary manual search activities. This decision is supported by the evidence that automatic searches and backward snowballing activities lead to similar results, and that the decision on which to prefer is context specific [13]. In order to ensure the correctness of the adopted manual approach, the backward snowballing activity will be based exclusively on the papers selected from the top-level software engineering venues. Furthermore, the backward snowballing results will be further contemplated by adopting a forward snowballing process, that will ensure the soundness and relevance of the set of selected primary studies.

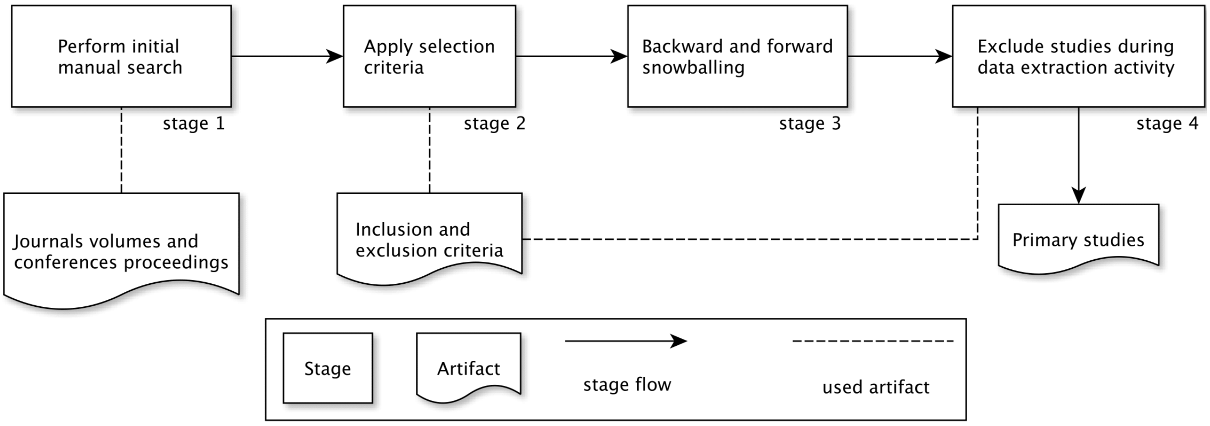
Perform initial manual search
stage 1
Apply selection criteria
stage 2
Backward and forward snowballing
stage 3
Exclude studies during data extraction activity
stage 4
Journals volumes and conferences proceedings
Inclusion and exclusion criteria
Primary studies
Stage
Artifact
stage flow
used artifact

Figure 2: The search and selection process of this study

Figure 2 shows our search and selection process. Specifically, the search and selection processes are conceived as multi-stage processes in order to control through a rigorous and pre-defined methodology the number and characteristics of the studies being either selected or excluded during the various stages. In the following sections each of the steps composing the search and selection processes are presented and detailed.

1. Perform initial manual search. In this initial stage we will perform a manual search by considering exclusively the researches published in the top-level software engineering conferences[2] and international journals[3] according to well known sources in the field. The publications of the considered venues will be thoughtfully examined by adopting several exclusion rounds in an adaptive reading depth fashion [14]. In the first round, the title of the researches will be examined. This first step will enable us to discard all those researches that clearly do not fall in the domain of static analysis of mobile applications. In the second exclusion round, the abstract and the conclusion of the remaining researches will be inspected. Finally, the selected researches will be further inspected by considering their full text in order to ensure that only the ones relevant for answering the research questions.

The time span of our search will be from January 2007[4] to December 2016, summing up to 8402 potentially relevant studies distributed across more than 9 years of research in software engineering (see Table 2).

2. Apply selection criteria. Once the papers will be selected through the initial search phase, the resulting studies will be filtered according to a set of well-defined selection criteria. The adopted criteria are detailed in Section 4.0.1. An adaptive reading depth [14] will be utilised, in order to

[2] http://goo.gl/auU7su

[3] http://www.webofknowledge.com

[4] given that the concept of mobile application exists only since 2007)

Conferences	#Studies	Journals	#Studies
International Conference on Software Engineering (ICSE)	810	IEEE Transactions on Software Engineering (TSE)	616
European Software Engineering Conference (ESEC)\ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE)	638	ACM Transactions on Software Engineering and Methodology (TOSEM)	205
International Conference on Fundamental Approaches to Software Engineering (FASE)	285	Information and Software Technology (IST)	1026
IEEE/ACM International Conference on Automated Software Engineering (ASE)	624	Automated Software Engineering (ASE journal)	149
ACM SIGPLAN conference on Systems, Programming, Languages and Applications: Software for Humanity (SPLASH)	480	Software Maintenance & Evolution - Research & Practice (JSEP)	352
European Conference on Object-Oriented Programming (ECOOP)	275	Software and Systems Modeling (SoSyM)	381
International Symposium on Software Testing and Analysis (ISSTA)	317	Empirical Software Engineering (ESEJ)	371
		Journal of Systems and Software (JSS)	1873
Total	3429	**Total**	4973

Table 2: Searched data sources

carry out the exclusion process in a time-efficient and objective manner.

3. Backward and forward snowballing. In order to mitigate a potential bias with respect to the construct validity of the study, the manual search previously presented is complemented with an additional snowballing process [15]. The snowballing activity will be adopted in order to further expand the number of considered researches by taking into account also studies that were published outside the contexts of the conferences and journal considered in the initial search phase. In particular, this process will be carried out by considering the studies selected in the initial search, and subsequently selecting relevant papers among those cited by the initially selected ones. This method is commonly referred to as a *backward snowballing* activity [16].

In addition to the backward snowballing, we also analyzed the researches citing the studies selected through the initial search. This process is usually referred to as a *forward snowballing* activity [16].

Specifically, we include this further literature search method in order to further expand and refine the selection of studies gathered through the initial search and the backward snowballing activity. Regarding the forward snowballing process, the *Google Scholar*[5] bibliographic database will be adopted to retrieve the studies citing the ones selected through the initial search phase. Elements discovered through this search activity that will not correspond to research papers, such as textbooks or technical reports, will not be included in the set of primary studies.

4. Exclude studies during data extraction activity. In this final stage the results gathered through the initial search, and the backward and forward snowballing activity. During this process duplicated entries resulting from the merge of the search process will be identified and merged into a single one. This latter identification process will be carried out in a automatic way by confronting the `title` and `year` attributes of the selected studies. The final decision about the inclusion of the papers will be based on the adherence of the full text of the studies to the predefined selection criteria presented in Section 4.0.1.

4.0.1 Selection criteria

Following the guidelines for systematic literature review for software engineering [7], in order to reduce the likelihood of biases, we have to define a rigorous set of inclusion and exclusion criteria during the initial protocol definition phase of the literature review. In the following we detail the set of inclusion and exclusion criteria that will guide the selection of the primary research studies for the proposed systematic mapping study. A research paper will be included in the set of primary studies if it satisfies *all* the inclusion criterion stated below. A study will be discarded if it satisfies *at least one* of the exclusion criteria reported below.

[5] `https://scholar.google.it/`

Inclusion criteria

I1) Studies proposing or using a static analysis method or technique for mobile applications.

I2) Studies in which the static analysis method or technique take as input one or more a mobile applications, in the form of binary files or the source code.

I3) Studies providing some kind of evaluation of the proposed method or technique (e.g., via formal analysis, controlled experiment, exploitation in industry, application to a simple example).

Exclusion criteria

E1) Studies not describing any implementation of the proposed method or technique.

E2) Secondary or tertiary studies (e.g., systematic literature reviews, surveys).

E3) Studies in the form of editorials, tutorial, short, and poster papers, because they do not provide enough information.

E4) Studies not published in English language.

E5) Studies not peer reviewed.

E6) Studies in which the static analysis method or technique takes as input only store metadata (e.g., user reviews, ratings) or other app artifacts (e.g., manifest files).

In order to reduce possible bias, three researchers performed the studies selection independently by applying the above mentioned criteria.

5 Data extraction

The main goal of the activity reported in this section is to (i) create a classification framework for the primary studies and (ii) to collect data from each primary study.

In our study, the classification framework will be composed of three distinct parts, each of which addresses one of the research questions of our study (see Section 3):

1. *Intensity of research*, addressing RQ1, see Section 5.1,
2. *Characteristics*, addressing RQ2, see Section 5.2,
3. *Potential for industrial adoption*, addressing RQ3, see Section 5.3.

In order to carry out a rigorous data extraction process, as well as to ease the control and the subsequent analysis of the extracted data, a predefined data extraction form will be designed prior the data extraction process.

The structure will be composed of the various categories of the classification framework. For each primary study, the principal researchers will collect in a spreadsheet a record with the extracted information for subsequent analysis: the spreadsheet columns will be the parameters, while each spreadsheet row will represent the data of each primary study.

As suggested in [8], the principal researchers will pilot the data extraction form independently. In order to validate our data extraction strategy, we will perform a sensitivity analysis to check whether the results are consistent independently from the researcher performing the analysis.

More specifically, the principal researchers will get a random sample of 5 primary studies and will analyze them independently by filling the data extraction form for each primary study. Then, they will assess their level of agreement and each disagreement will be discussed and resolved with the intervention of the research methodologist, if needed.

When going through a primary study in detail for extracting information, researchers can agree that the currently analysed study may be semantically out of the scope of this research, and so it can be excluded.

We will follow a systematic process called *keywording* [17] for defining some of the parameters of our comparison framework (namely the *analysis goal*, the *additional inputs* and the *analysis technique*, detailed in the sections below). Basically, keywording aims at reducing the time needed to develop the comparison framework while ensuring that it takes the existing studies into account.

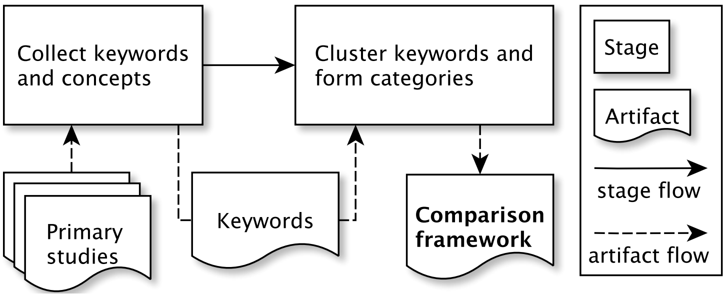

Figure 3: Overview of the keywording process

Figure 3 shows our keywording process in more details. Keywording is done in two steps:

1. *Collect keywords and concepts*: the three principal researchers will collect keywords and concepts by reading the abstract of each primary study. When all primary studies will be analysed, all keywords and concepts will be combined together to clearly identify the context, nature, and contribution of the research. Bearing in mind that the authors of the primary studies may use different terms for the same concepts and viceversa (e.g., program analysis vs static analysis), we will collate different keywords and terms to ensure consistency and compatibility. The output of this stage will be the set of keywords extracted from the primary studies.
2. *Cluster keywords and form categories*: once the keywords and concepts will be finalized, the three principals researchers will perform a clustering operation on them in order to have a set of representative clusters of keywords. The output of this stage will be the classification framework containing all the identified parameters (each of them having a specific type and possible values), representing a specific aspect of static analysis for mobile apps.

5.1 Research trends (RQ1)

In the following we list all the parameters that will be extracted in order to answer the first research question of our study.

1. *Year of publication*: for extracting the publication tendency per year.
2. *Publication venue*: identifies in which venues researches on static analysis of mobile applications appear more frequently.
3. *Publication venue type*: identifies the type of the publication venue, i.e., (i) journal, (ii) conference, and (iii) workshop.
4. *Analysis goal*: identifies which is the goal of the static analysis, e.g., privacy leaks identification, malware detection, energy assessment, etc. The previously defined *keywording* process will be utilized to identify the discrete values of this attribute.
5. *Macro analysis goal*: identifies which is the generic goal of the static analysis considering three distinct values: (i) *external quality*, if the approach evaluates some external quality attribute, e.g. performance; (ii) *internal quality*, if the approach evaluates some internal quality

attribute, e.g. maintainability; (iii) *improving of methodology*, if the approach is conceived to improve a static analysis technique.

6. *Paper goal*: characterized by two different and mutually exclusive values: (i) *Quality attribute assessment*, if the research reported in the primary study focuses on assessing a quality attribute of mobile apps (e.g., security); (ii) *Improvement of methodology*, if the research reported in the primary study focuses on improving existing static analyses for mobile apps.

5.2 Characteristics (RQ2)

In the following we present the data attributes extracted from the primary studies in order to obtain an overview of the characteristics of static analysis of mobile applications.

1. *Platform specificity*: identifies whether the proposed approach is specifically designed for a specific platform (e.g., *Android* or *iOS*) or if it is *generic* and can in principle be applied to any platform.

2. *Implementation*: identifies whether the implementation used for evaluation purposes is implemented for a specific platform, e.g., *Android* or *iOS*, or it is *Generic*, applicable to apps developed for any platform.

3. *Static/Hybrid approach*: identifies whether an approach relies on static analysis only (*Static*) or utilizes some form of dynamic analysis also (*Hybrid*).

4. *Usage of machine learning*: identifies whether the static analysis approach under evaluation complements its analysis with machine learning techniques.

5. *App artifact*: identifies what formats are accepted as input by the selected studies for the apps to be analyzed by the static analysis approach presented: (i) binary packages (*Binary*), i.e., APK (Android PacKage) files for the Android platform or IPA (iPhone Application Archive) packages for the iOS platform, or (ii) the *source code* of the application considered.

6. *Additional inputs*: identifies what other inputs, if any, are required by the primary studies to perform the proposed analysis (in addition to the app itself), e.g., source code mappings, platform descriptions, bug information. As for the *analysis goal* attribute, a *keywording* process will be adopted to identify appropriate clusters of keywords for this attribute.

7. *Analysis pre-steps*: identifies whether the studies under evaluation require steps that must be executed manually before the analysis can be performed. Examples of possible pre-steps include, but are not limited to, building models of the platform APIs or libraries used by the application under analysis, collecting execution traces, collecting runtime power consumption measures, creating rule sets or security policies.

8. *Analysis technique*: identifies the family of static analysis techniques performed by the approaches proposed in the primary studies, e.g., flow analysis, taint analysis, data mining. The discrete values that this attribute can assume will be defined through the *keywording* process.

5.3 Potential for industrial adoption (RQ3)

This latter set of attributes has been selected in order to assess to what extent the static analysis of mobile applications presented in the primary studies can be adopted in an industry projects. The selected parameters are as follows:

1. *Target stakeholder*: identifies the potential consumer of the static analysis tool, i.e., *app developer*, *platform vendor*, *user*, or *researcher*.

2. *Tool availability*: reports if the static analysis tool presented in the research is available online or not.

3. *Number of analysed apps*: identifies the number of applications considered during the evaluation phase. We categorize the *number of analysed apps* according to the following sets: (i) *low*, if the number of applications used for evaluating the proposed approach is less than 100; (ii) *medium*, if the number of applications is between 100 and 1,000; (iii) *high*, if the number of applications is greater than 1,000.

4. *Applied research method*: represents the type of applied research method used to assess the proposed technique. Possible values of this parameter are *Validation* and *Evaluation*. *Validation* is done in lab contexts using applications specifically created or customized for the purpose of their approach evaluation. *Evaluation* takes place in real-world (industrial) contexts, using exclusively unmodified applications. The latter generally provides a higher level of evidence about the practical applicability of a proposed technique.

5. *Industry involvement*: assessing the affiliation of the authors of the research. This attribute can assume three distinct values: (i) *Academia*, if the authors are affiliated exclusively to an academic organization, e.g., university or research center; (ii) *Industry* if the authors are affiliated exclusively to an industrial organization, e.g., a company, startup, or software house; (iii) *Academia and Industry* if some of the authors are affiliated to an academic organization and some others to an industrial one.

6 Data synthesis

The data synthesis activity involves collating and summarising the data extracted from the primary studies [18, § 6.5] with the main goal of understanding, analysing, and classifying current research on static analysis of mobile applications.

In this phase we will have a fully populated spreadsheet with all the information coming from the data extraction form of each primary study. According to this, our data synthesis will be divided into two main phases: vertical analysis and horizontal analysis. When performing *vertical analysis*, we will analyze the extracted data to find trends and collect information about each parameter of each category of our classification framework. When performing *horizontal analysis*, we will analyse the extracted data to explore possible relations across different parameters of our classification framework. In both phases we will perform a combination of content analysis (mainly for categorizing and coding the studies under broad thematic categories) and narrative synthesis (mainly for explaining in details and interpreting the findings coming from the content analysis).

Vertical analysis. Depending on the parameters of the classification framework (see Section 5), in this research we will apply both quantitative and qualitative synthesis methods, separately. When considering quantitative data, depending on the specific data to be analysed, we will apply descriptive statistics for better understanding the data. When considering qualitative data, we will apply the *line of argument* synthesis [8], that is: firstly we will analyse each primary study individually in order to document it and tabulate its main features with respect to each specific parameter of the classification framework, then we will analyse the set of studies as a whole, in order to reason on potential patterns and trends. When both quantitative and qualitative analyses are completed, we will integrate their results in order to explain quantitative results by using qualitative results [18, § 6.5].

Horizontal analysis. We will cross-tabulate and group the data, and make comparisons between two or more nominal variables. The main goal of the horizontal analysis is to (i) investigate on the existence of possible interesting relations between data pertaining to different parameters of the comparison framework. We will use contingency tables for evaluating the actual existence of those relations and we will identify perspectives of interest.

7 Dissemination strategy

In the following we list the actions we will undertake in our dissemination strategy:

1) we will report our main research-oriented findings and a detailed description of this study into an scientific publication in an international scientific journal;

2) a preliminary study [12] has been conducted in order to gather an initial overview of existing approaches for static analysis of mobile apps. The research methodology described in this paper is based on the results gathered through the preliminary study.

3) an accompanying *technical report* will present all the details and raw data of the study; the chief aim of the technical report is to make our study replicable by interested researchers.

References

[1] Adam Lella, Andrew Lipsman, The U.S. Mobile App Report, comsCore white paper (2017).

[2] L. Li, T. F. Bissyandé, M. Papadakis, S. Rasthofer, A. Bartel, D. Octeau, J. Klein, Y. Le Traon, Static analysis of android apps: A systematic literature review, Information and Software Technology.

[3] I. Garcıa-Ferreira, C. Laorden, I. Santos, P. G. Bringas, A survey on static analysis and model checking, in: International Joint Conference SOCO'14-CISIS'14-ICEUTE'14: Bilbao, Spain, June 25th-27th, 2014, Proceedings, Vol. 299, Springer, 2014, p. 443.

[4] M. Pistoia, S. Chandra, S. J. Fink, E. Yahav, A survey of static analysis methods for identifying security vulnerabilities in software systems, IBM Systems Journal 46 (2) (2007) 265–288.

[5] F. Elberzhager, J. Münch, V. T. N. Nha, A systematic mapping study on the combination of static and dynamic quality assurance techniques, Information and Software Technology 54 (1) (2012) 1–15.

[6] R. Harrison, D. Flood, D. Duce, Usability of mobile applications: literature review and rationale for a new usability model, Journal of Interaction Science 1 (1) (2013) 1–16.

[7] B. Kitchenham, P. Brereton, A systematic review of systematic review process research in software engineering, Information and software technology 55 (12) (2013) 2049–2075.

[8] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, A. Wesslén, Experimentation in Software Engineering, Computer Science, Springer, 2012.

[9] H. Zhang, M. A. Babar, Systematic reviews in software engineering: An empirical investigation, Information and Software Technology 55 (7) (2013) 1341–1354.

[10] V. R. Basili, G. Caldiera, H. D. Rombach, The Goal Question Metric Approach, in: Encyclopedia of Software Engineering, Vol. 2, Wiley, 1994, pp. 528–532.

[11] C. Wohlin, Guidelines for snowballing in systematic literature studies and a replication in software engineering, in: Proceedings of the 18th international conference on evaluation and assessment in software engineering, ACM, 2014, p. 38.

[12] M. Autili, I. Malavolta, A. Perucci, G. L. Scoccia, Perspectives on static analysis of mobile apps (invited talk), in: Proceedings of the 3rd International Workshop on Software Development Lifecycle for Mobile, ACM, 2015, pp. 29–30.

[13] S. Jalali, C. Wohlin, Systematic literature studies: Database searches vs. backward snowballing, in: Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM '12, ACM, New York, NY, USA, 2012, pp. 29–38.

[14] K. Petersen, R. Feldt, S. Mujtaba, M. Mattsson, Systematic mapping studies in software engineering, in: Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering, EASE'08, British Computer Society, Swinton, UK, UK, 2008, pp. 68–77.
URL http://dl.acm.org/citation.cfm?id=2227115.2227123

[15] T. Greenhalgh, R. Peacock, Effectiveness and efficiency of search methods in systematic reviews of complex evidence: audit of primary sources, BMJ 331 (7524) (2005) 1064–1065.

[16] C. Wohlin, Guidelines for snowballing in systematic literature studies and a replication in software engineering, in: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, EASE '14, ACM, New York, NY, USA, 2014, pp. 38:1–38:10.

[17] 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.

[18] B. A. Kitchenham, S. Charters, Guidelines for performing systematic literature reviews in software engineering, Tech. Rep. EBSE-2007-01, Keele University and University of Durham (2007).