Faculty of Engineering and Technology

Master of Software Engineering (SWEN)

Thesis

Cross-Platform Mobile App Development: A Qualitative Research and Systematic Mapping Study

By

Student Name: Tasnim Zohud

Student Number: 1165059

Supervised

By

Dr. Samer Zein

1/5/2019

Abstract

Cross-platform frameworks for mobile application (app) development allow developers to deploy solutions on a range of platforms such as iOS and Android. Mobile apps developed using cross-platform frameworks can be run over different platforms, its development process reduces the time, effort, and resources cost. With the increased use of such frameworks, it is of utmost importance to understand the contributions and challenges in this emergent field, especially at industrial contexts. Although there have been numerous studies in this area of research, there is a lack of a coherent view at the industrial contexts. To address this issue, a Systematic Mapping Study (SMS) is conducted to map state-of-the-art empirical studies in this field. The SMS study introduced new perspectives for the intended case study at the industrial context. More specifically, the SMS study revealed a need to focus on code maintenance as well as testing methods. Accordingly, in this thesis, a qualitative research is conducted with *four* different software development companies in Palestine. This study aimed to understand how industrial teams approach mobile cross-platform development. Further, it worked to identify the challenges they face in the areas of code maintenance and software testing. The findings concluded in this research cover the important aspects in the industrial contexts, including the development process of how and why the cross-platform development approach is chosen. From the practitioners' perspectives, the developers' experience is the most influential factor in the development process. Further, it covers the used technologies in the industry and the dominant one from the practitioners' perspectives, where they agreed that React-Native is the promising and dominant technology. Additionally, the main challenges that faced by the developers are introduced, it has been noticed that the libraries written by the providers are the major challenging area. Moreover, this research discussed the testing and found that the majority of the companies support the manual testing only. Finally, the maintenance for the cross-platform apps is also discussed, the developers clarified that the maintenance process becomes complex when the app has many injected native modules.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank my thesis advisor Dr. Samer Zein for the great effort he provided.

Chapter 1 Introduction

1.1 *Introduction and Motivation*

Nowadays, smartphones are becoming widely used in different life aspects such as, business, education, and entertainment to mention just a few. Moreover, mobile apps are being integrated into critical sectors such as m-banking, m-payments, and health (Nagappan et al., 2016). Further, the area of mobile app software engineering emerged to be one of the most rapidly growing areas.

There are three different approaches to build mobile apps: native, mobile web (m-site), and cross-platform. Native mobile app development approach is a methodology through which building apps is done using the programming language for a specific platform. Additionally, native mobile apps operate only on the platform that they're built for, which means that developing native applications for different platforms needs to hire different teams, each for separate technology. Building responsive mobile web apps (m-sites) is another approach for mobile app development. M-site apps are built using web technologies, mainly HTML, CSS, and JavaScript and can be accessed through web browser. Further, m-sites are highly connected and they have a limited access to mobile device hardware and sensors. The final approach is the cross-platform which includes different solutions such as hybrid, interpreted, cross-complied and other approaches. This approach provides solutions to facilitate building an app once then deploy it into a spectrum of platforms. Since the resulted app can be run over different platforms, it reduces the time and resources cost for development.

Cross-platform development approach is becoming the first choice for developing mobile apps, nowadays, companies are taking this approach in consideration when designing a new app or planning to modify a legacy one. Moreover, the number of developed cross-platform apps is rising, and its market is almost hit $7.5 billion by end of 2018 (Furlan, 2018), additionally, the cross-platform development tools amount is growing. This vital development field deserves more investigation in order to identify its strength and gaps, which can be used as a reference for the developers and companies who are planning to take decision of what development approach to use. Although there have been numerous studies in cross-platform area, but to the best of our knowledge, there is no studies investigate the cross-platform development in the industrial

contexts. In this field, most of the studies conducted a comparative analysis between different tools or frameworks of cross-platform development, and other studies analysed specific framework or tool in addition to studies that proposed new development approach in the field of cross-platform. Additionally, to our knowledge, there is no comprehensive systematic review, therefore, we conducted a systematic literature review in order to give an overview of the existing studies, to create a reference for the researchers who want conduct additional studies in this field, and to explore the research needs in the industrial context which we investigated in this thesis.

According to the findings revealed from the SMS, there is a lack in research that investigate the code maintenance for cross-platform mobile apps, moreover, there is a need to focus on testing methods used in this development area. In this thesis, a research with four different software development Palestinian companies is conducted to cover the research gaps, which aimed to understand how the development teams in the Palestinian companies develop the cross-platform apps, in addition to identify the issues and challenges that developers face in the areas of code maintenance and testing.

The purpose of this empirical study that conducted in the industry is to focus on the cross-platform development process and challenges form the practitioners' perspectives. In terms of process, it's noticeable that the decision of which development approach and tool to use for developing an application depends on the developers' experience first then some other factors. Moreover, this research found that the dominant cross-platform tool in the industry is React-Native (Nakazawa, 2019), and this could be due to Facebook's support for it. The conducted study found some common challenges faced by the developers, the hardware calls, the libraries provided by the community and the communities for the cross-platform technologies are the most reported challenges. Regarding the testing techniques, the manual testing is the supported technique in most of the companies, since it is effective enough and saves time and resources cost.

1.2 *Research Objectives and Problem Statement*

Selecting the development approach is a critical decision that industrial teams should take. Cross-platform mobile app development approach have many benefits that can make the developers willing to use such as cost effectiveness since the resulted apps are developed once then deployed everywhere. Despite of that, cross-platform approach has many challenging aspects

that are not well investigated at industrial contexts. Studies that explore the reality of cross-platform development especially in maintainability and testability aspects are still highly missing. Based on the research problem identified above, the following research quest ions are formulated:

Q1) How do industrial teams develop cross-platform mobile apps in term of development process and tools used? And what are the challenges they face?

Q2) To what extent do cross-platform frameworks support code maintenance?

Q3) What are the testing techniques supported by cross-platform frameworks? And How much effective are they?

1.3 *Summary of progress and contribution*

- A systematic mapping study is conducted in the field of cross-platform development
    - SMS provides an unbiased overview for the existing studies in this field
    - It analysed the studies to provide a classification scheme and identify the research gaps in the literature
    - The produced classification scheme and the revealed gaps are useful for both the researchers and practitioners
        - Researchers will have an obvious view for the already conducted studies, and the gaps that literature didn't cover so they can investigate more in these areas in their future research
        - Practitioners can use the SMS results to easily find solutions for the issues they face
- The following research gaps that have been concluded in the SMS study from the previous work are investigated through multiple case-study research in a real industrial context
    1. Lack of studies focusing on the testing the cross-platform mobile apps
    2. Lack of studies investigating the maintenance for the cross-platform mobile apps
    3. No qualitative studies investigated the cross-platform mobile apps development in the industry
- This empirical research that conducted in the industrial contexts aims to:
    - Give a better understanding about the industry's perception in the area of cross-platform development
    - Give an opportunity for the future research to focus on the current real-world issues

- Interviews and focus groups have been conducted with cross-platform mobile development teams from **four** different software development companies
- The following points were discussed to answer the research questions:
    - Development process in the field of cross-platform mobile development
    - The used cross-platform technologies in the industrial companies
    - The dominant cross-platform technology in the industry
    - The common challenges faced by the developers
    - The supported testing techniques and their efficiency and limitations
    - The maintainability of the cross-platform app
- The Systematic Mapping Study is published in the **Journal of Computer Science (JCS)**

Chapter 2 Systematic Mapping Study

2.1 *Introduction*

The Systematic Mapping Study is a technique to create a classification scheme and build a structure for certain field of interest. The results of this kind of studies show the distribution of the published studies over specific time, and the frequencies for these studies within identified categories. SMS provides an overview of the existing studies in a research field, and reduces the bias in studies inclusion since there should be a defined inclusion/exclusion criteria for studies selection procedure (Petersen, 2008). The analysis of the included studies shows conclusion that can be taken in consideration in both research and industrial contexts, whereas SMS conclusion shows the categories that are well-investigated through the frequencies of the conducted studies in each category, and it identifies the research gaps existing in the field of mapped studies. This is important for the researchers in order to identify future research options, and industrial teams can rely on the SMS results for technical decisions when needed.

Cross-platform includes different solutions such as hybrid, interpreted, cross-complied and other approaches. These technologies facilitate developing an app once then deploy it everywhere. The resulted app can be run over different platforms which reduces the time and resources cost for development. Statistics confirms that the market share of the cross-platform mobile apps is $7.5 billion at the end of 2018 (Furlan, 2018). Cross-platform technologies are becoming viable choice for creating mobile apps due to their advantages.

As the best of our knowledge, there is no comprehensive systematic mapping study in the area of cross-platform mobile apps development. This provided a motivation for conducting a mapping study in this field. Our mapping study collected and described all the relevant studies in order to highlight the research gaps in the field. Initial search results produced 295,816 results. However, after applying a multi-step inclusion/exclusion criteria, a total of 30 studies were finally included in this mapping. We present the studies based on four classification scheme (i) structure of the topic; (ii) contribution facets; (iii) applied techniques; and (iv) research facets. We argue that evaluation studies should be based on more complex and real-world apps; there is a lack of studies focusing on testing solutions/challenges for cross-platform app development; and more research should be directed on the maintenance aspect of cross-platform development.

2.2 *Motivation & Related Work*

To our best of knowledge, the study by (Amatya et al., 2013) is the only study that provides a survey about the trends in cross-platform development. In their study, the authors mainly focused on the raised development issues in cross-platform field. Further, they didn't include the studies that investigated development tools or approaches. Moreover, their study didn't summarize the included studies, they merely presented identified problems with the proposed solutions in each study. In contrast, our study includes the most recent research in the area of cross-platform development. Additionally, we covered all the relevant studies with different areas of interests, and we provide a brief summary for each study.

In another study conducted by (Latif et al., 2017), the authors gave an overview of the recently applied development tools and approaches. They provided a recap for the cross-platform approaches and their platforms. The described approaches are: web, hybrid, interpreted, cross-compiled, and model-driven approaches. They compared the approaches based on their use and most popular platforms. They concluded that the hybrid approach is good for developing low complex features. Further, they argue that cross-compiled platforms are good for enterprises development work, since it requires developing once then can be deployed everywhere as native app.

Another study conducted by (Ribeiro et al., 2012) provided a global view of the cross-platform development technologies, and highlighted their pros and cons. In their study, six different tools have been analyzed and compared against each other. The analyzed tools are: Rhodes, PhoneGap, DragonRAD, Titanium, mobile, and mdsl. The study identified the strength and weakness points for each tool, it also discovered that the majority of the tools don't apply model driven engineering, only two of them applied it.

The study of (S. Alamri et al., 2014) identified the challenges in cross-platform development. They summarized that the major issues are (1) poor performance; (2) low user interface quality; (3) limited access to hardware features. A study of (El-Kassas et al., 2015) surveyed the existing cross-platform tools to provide an overview for the most recent used tools. Their study can be used as a reference for the developers in cross-platform area. It also provided the open issues in this area which are (1) find a solution that is compatible for different mobile platforms; (2) support the native programming language and consider the APIs differences; (3) support the native user interface; (4) support code reuse.

There is no rigorous systematic mapping study in the area of cross-platform development. We found only one literature survey (Amatya et al., 2013) which conducted in a very early stage of cross-platform development. We found relevant studies that provided an overview for the existing development tools (e.g. (Latif et al., 2017), (Ribeiro et al., 2012) and others). Further, we found some studies that reported the challenges in this field such as (S. Alamri et al., 2014) and (El-Kassas et al., 2015). This motivated us to apply a rigorous systematic mapping study in order to provide a wide overview for the existing and most recent research in this emergent field.

2.3 *Method*

This study applied the systematic mapping method by following the guidelines provided in (Petersen et al., 2008), Further, the design of our study is inspired by other systematic mapping study by (Zein et al., 2016). In general, systematic mapping studies provide an overview for the existing research in a specific field.

As described by (Petersen et al., 2008), a systematic mapping study has five phases. The first phase is specifying the research questions. The second phase is the search process where the researcher follows a pre-defined search strategy to search for and select studies. The third phase is about skimming the studies, then extracting keywords from the abstracts during the fourth phase. The last phase is extracting the data from the studies and mapping them into the defined scheme.

2.3.1 *Research Questions*

This study built a classification scheme based on the included studies in the field of cross-platform mobile app development. Moreover, we presented an overview of the existing research and tried to find the research gaps and challenges in order to guide the future research in this field. We specified four research questions to achieve the goal of this mapping study. The primary question is to identify the contribution for each study included in this mapping study. Then, we stated three sub-questions to present the main achievement; main challenges that have been investigated; and present the main research techniques they applied:

Primary RQ: What are the empirical studies that have been done in the area of mobile app cross-platform development?

Sub-RQ1: What accomplishments do these studies present, and what contribution facets do they provide?

Sub-RQ2: What are the main challenges addressed by these studies?

Sub-RQ3: What are the main research techniques applied by these studies?

2.3.2 *Sources of Evidence*

The present study was conducted at Birzeit Univerity at Palestine, the information sources were the online libraries provided by the university, namely: IEEExplore, ACM Digital Library, and Google Scholar.

2.3.3 *Search Strategy*

Included studies in this review are from both quantitative and qualitative approaches, they are all related to cross-platform mobile app development. The adopted search strategy in this study to form the search string is inspired from the study of (Kitchenham, 2007):

- Use the Boolean operators OR and AND:
    - Use AND to limit the search by major terms
    - Use OR to broaden the search
- Search for alternative spellings

In our search process, it took several attempts to finalize the good search string, which has been evaluated according to the number of retrieved studies and their relevance. Our challenge was handling the fact the mobile word is a general term that could refers to many things rather than the smartphone, cross-platform term is used in different contexts not only in mobile development, and the words application and app are not equals in the search, each one could returns different studies.

The terms included in the search string were basically inspired from the proposed research questions. The terms “cross-platform mobile applications” formed the main search terms, we included the synonym for the cross-platform term by using hybrid term, and the alternative for applications which is apps.

Additionally, we followed the search approaches mentioned in (Feldrer et al., 2018) which are:

- Backward Snowballing: get the studies that are cited by the other studies included in the references
- Forward Snowballing: get the studies the are cited in the existing studies

2.3.4 *Study Selection Criteria and Process*

The search process was done using the advanced search mechanism provided by the online libraries. We restricted out search to cover only the computer science field since hybrid term is widely used in other engineering fields.

The research studies selection phases were iterative and incremental, each study passed through three different stages in order to filter the studies which will be included in the mapping study. The earlier phase was applying a search string to get the relevant studies. Next in the first phase, we excluded the papers that are not relevant to cross-platform mobile development, the decision is based on their titles and abstracts. In the second phase we applied selection criteria. We excluded the old papers that have been published before 2013. Further, we excluded short studies that are less than five papers. Additionally, we excluded the non-empirical studies by reading the introduction, methodology and conclusion. The final steps were completed by reading the remaining papers (see Figure 1).

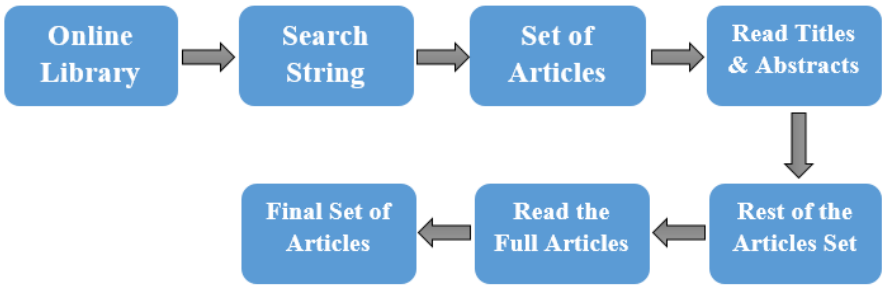

Figure 1: Studies selection process

2.3.5 *Keywording of Abstracts (Classification Scheme)*

The keywording process is performed based on the guidelines provided by (Petersen et al., 2008). This process is done through two steps. The first step is to read the abstracts of the selected papers in order to find the main keywords which indicate the studies contribution. If the abstracts don't have the good quality that allows keywords extracting, then researcher can read the introduction and conclusion sections. The goal of this process is to build the classification scheme which represents the included studies.

The second step is applying a thematic analysis approach in which the chosen keywords from different studies are combined together to build a high-level understanding of the selected papers contribution. This step led to identify a set of categories for the classification scheme.

The keywords and concepts resulting from the first phase reflected the contribution for the papers. Example of concepts we got are compare cross-platform approaches, analyze cross-platform approach, introduce new approach, impact of using cross-platform, and cross-platform challenges and opportunities. During the second phase, the resulted concepts were combined together to identify the contribution area for each paper. The results were the main categories in the classification scheme which are "comparative analysis", "framework analysis" and "new approach".

2.3.6 *Data Extraction and Mapping of Studies*

The aim of this phase is to analyze the selected studies to extract the data required to answer the proposed research questions. The primary studies were sorted according to the defined classification scheme. Data was organized in different tables, the relevant frequencies for each category were calculated. The extracted data addressed the accomplishments and the contribution facets, the applied techniques, and the main challenges.

The main challenges were extracted from the primary studies and summarized according to the identified categories. In addition, we extracted the research methods applied in the selected studies. The defined classification scheme gives a coherent overview of the included studies.

2.4 *Results*

2.4.1 *Search Results*

We did several attempts to make sure that we have the most reliable search string. As mentioned earlier, our search strings are based on "mobile", "applications", "apps" and "hybrid" terms. Finally, and after different attempts with different pilot search strings, we agreed on the final one that could be considered an appropriate since it has all the necessary terms and almost all the retrieved research studies are resulted from this search string, which is "cross-platform/hybrid mobile application/app".

At the beginning, our search results retrieves 295,816 studies from all sources as mentioned in source of evidence section. After that, we applied the three filtration steps to decide what studies

will be included in the mapping study. Table 1 shows the online libraries we searched in, the filtration steps we applied and their results. At the end, 30 studies have been included in the mapping study, 13 studies retrieved from ACM library which forms the highest percentage (43%) in the total results. And, 7 studies (24%) came from the IEEE library, while 10 studies (33%) resulted from google scholar.

Figure 2 shows the distribution of the selected studies according to the publication year. The earliest studies are those published in 2013, and most of the studies are published in 2016.

Online Database	Search Results	Phase 1	Phase 2
ACM Digital Library	200,080	25	13
IEEExplore	2736	23	7
Google Scholar	93,000	16	10
Total	295,816	64	30

Table 1: Studies in each filtration step

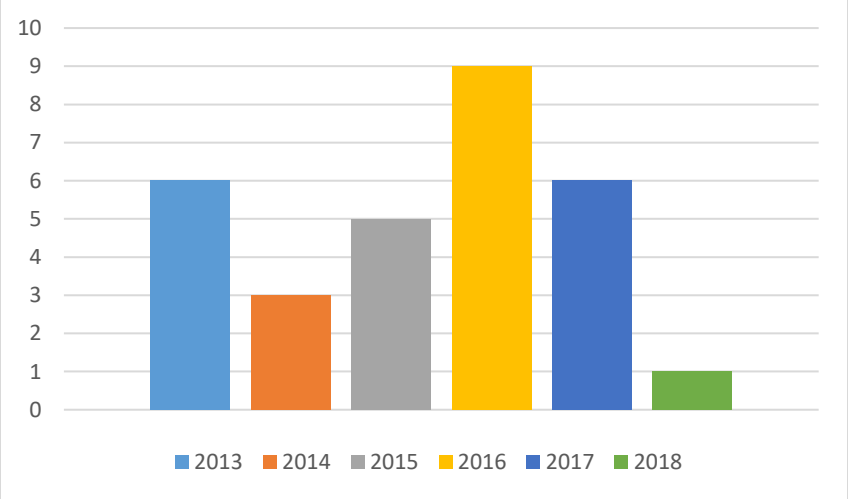

Figure 2: Studies distribution based on the publication year

2.4.2 *Classification Scheme*

We defined a classification scheme of four categories: (1) structure of the topic; (2) contribution facets; (3) applied techniques; and (4) research facets. The first category (structure of the topic) is defined based on the thematic analysis, it's classified into three sub-categories which are: "comparative analysis", "framework analysis", and "new approach". These sub-categories

were identified during the keywording phase according to the main topic investigated in each included study.

The second category that forms the answer for sub-RQ1 is the contribution facets, which are inspired from the study by (Shahrokni, 2013). We concluded with six contribution facets: metrics, tool, framework, review, evaluation, and method. The study that presents a guideline in a specific area to be followed belongs to metrics facet. Those which provide specific software tool for development or testing are grouped under tool facet. The studies that provide an overview for a set of existing studies in the field are classified under review facet. Evaluation facet which contains the highest number of studies, all provided a comparison for tools, approaches or frameworks. The studies that explain specific goal or research questions belong to method facet.

The third category which is concerned with applied techniques, answers the sub-RQ3. The empirical studies were classified based on their research method, more specifically: experiment, survey, case study, interviews, and mixed methods. Most of the included studies applied an experiment, and two of them supported their experiment with survey. Further, only one study conducted an interviews.

The fourth category is inspired from (Petersen et al., 2008) which classified the studies based on the research facets. It contains three research facets, validation research, evaluation research, and solution research.

- Solution research are those propose a solution and provide a discussion about how it's doable without validate it.
- Validation research are the studies that provide an investigation for the new proposed solution that are not implemented in practice yet.
- Evaluation research facet contains the studies that provide an investigation for some implemented technique or for known issue.

2.4.3 *Answering the research question*

Primary RQ: What are the empirical studies that have been done in the area of mobile app cross-platform development?

The studies presented in this mapping study have been categorized based on the classification scheme discussed previously. The basic category to cover and highlight the main research topic includes: comparative analysis, framework analysis, and new approach. This study

covered 30 research studies, 14 of them are under comparative analysis category, 7 studies grouped under framework analysis, and 9 studies came under new approach category. Table 2 shows the mentioned categories and the studies belong to each one, it's clear that the highest number of studies are those under comparative analysis.

Category	**Studies (S)**	**Total # studies**
Comparative Analysis	S2, S4, S5, S6, S7, S10, S11, S12, S14, S15, S16, S17, S21, S25	14
Framework Analysis	S1, S3, S9, S13, S18, S23, S26	7
New Approach	S8, S19, S20, S22, S24, S27, S28, S29, S30	9

Table 2: Main categories and the mapped studies

2.4.3.1 *Comparative Analysis*

Past comparative studies have been done on different ways, the applied research methods differs for the studies. Some studies executed the comparison through an experiment, others with case study, few of them using survey and some of them decided to support the main method with another one so they done the evaluation using mix of methods.

The contribution for each study varies from one to another, some studies compared the cross-platform tools against each other and against native development tool, and other studies compared the applications that have been developed in a hybrid way against those developed using the native technologies. This section presents 14 studies that conducted a comparison analysis S2, S4, S5, S6, S7, S10, S11, S12, S14, S15, S16, S17, S21 and S25.

(Taneja et al., 2016) [S2] analyzed different development approaches that can be used to achieve the platform independence. Three approaches have been discussed and compared against each other in order to test their performance and decide the approach that can solve the platform dependence. Virtual Machine based, Distributed Computation based and Hardware based approaches have been evaluated according to two performance parameters which are throughput and response time. They executed the experiment on four different applications that vary in their complexity and length. The best throughput and response time are achieved from the hardware based approach, and the worst throughput and delay record was for the distributed computation based approach, while the virtual machine based performance was in between the previous approaches.

The survey applied by (Latif et al., 2016) [S4] presented a comparison of cross-platform development approaches, it analyzed five approaches, web, hybrid, interpreted, cross-compiled and model driven approach. It's also discussed the desirable features for the eligible development tool which consider the scalability and maintainability of the application, the access capability to device features, the resources consumption, the security level and the development environment. They concluded that each approach has its advantages and challenges, but the cross-platform development can be considered as the best approach when the time and cost are limited

(Heitkotter et al., 2013) [S5] evaluated the development approaches of cross-platform apps which are the Web apps and the apps developed by PhoneGap and Titanium. The evaluation has been done through interviews with experienced developers and based on some criteria that was divided into two sections, one is related to the infrastructure perspective such as the license, cost, and supported platforms and more, the other is about the development perspective like ease of development, maintainability, scalability and others. After that, they compared those apps with native apps. This evaluation study found that developing the apps natively isn't necessary since the cross-platform approaches are mature enough to cover the needs of building mobile apps.

The study of (Xanthopoulos et al., 2013) [S6] investigated the cross-platform application types, which are web, hybrid, interpreted and generated apps. Then, a comparative analysis is done based on the historical review of each app type and according to set of criteria. They concluded that there is no perfect development approach, the approach selection decision should be taken with considering the needed requirements and expectations of the app. For example, web apps are a good beginning which can be used in another approaches, the hybrid and interpreted apps are must when the publication on the store is required. A case study is applied by designing an interpreted application of RSS reader, the application is created using Titanium tool. The case study supports that the interpreted approach is promising, they built the application using JavaScript without any experience in the environment platform.

The comparison analysis done in (Willocx et al., 2016) [S7] evaluated an application that has been built using both native and hybrid technologies, it is PropertyCross app. This research applied a quantitative analysis approach to measure the performance of the cross-platform development tools. The measurements parameters are the response time, CPU usage, memory usage and disk usage, they have been calculated on five different devices with the three main platforms, Android, iOS and Windows Phone. This study discussed several findings that can help

developers in selecting the tool when developing a hybrid mobile application. The cross-platform tools have the most CPU and memory usage and they recorded the slowest app launch time, but they respond as fast as native apps during pages navigation.

(Delia et al., 2015) [S10] presented a case study that discussed the most important features of four different cross-platform development approaches. The analyzed cross-platform approaches are mobile web, hybrid, interpreted and cross-compiled. The presented case study is a web application designed only for desktops named WebUNLP application, which is a virtual environment for teaching and learning, it contains many features to enable sharing materials and facilitate the communication between the educators and students and other beneficial features. As part of the case study, one feature has been chosen to be adopted into mobile platform which is the electronic notice board. The feature is developed with the all cross-platform approaches mentioned above. In case of mobile web application, it was simple to develop and deliver, but the notifications are not received when there is new post on the board. Regarding the hybrid application, it was developed by PhoneGap which provides the simplicity of web approach and use all device features. Although its performance was worse than the native app but it was much better than the mobile web app. Interpreted version was developed using Titanium, it was helpful in achieving a high level of performance by generating the code for the interface. But, it doesn't afford a tool that enables interface design visualization. Finally, the cross-compiled application was developed using Xamarin/Visual Studio and Delphi XE6. Fully native versions were obtained in both technologies.

A survey of different cross-platform tools is conducted in (Dalmasso et al., 2013) [S11], it covered the PhoneGap, Titanium and Sencha Touch tools. The conducted survey used example of applications developed by the mentioned tools, and then had been evaluated in terms of CPU, memory usage and power consumption. The results of this survey showed that the PhoneGap utilized less CPU and memory, and consumed less power than Titanium. Moreover, PhoneGap and Sencha Touch are working very well together when the memory availability is not an issue, they also provide a good UI design.

A comparison analysis for hybrid apps and their tools is conducted in (Ali et al., 2016) [S12]. 15,512 hybrid apps were collected from the stores, they identified what cross-platform tools (CPT) they used and how users rate them. Then, the rating of hybrid apps is compared against the native apps rating from the same category. At the end, they compared the hybrid app rating for

Android against iOS. The success of the CPT is measured according to the number of reviewers, rating and number of downloads for the app in the store which have been analyzed through four research questions. Although PhoneGap tool is the most popular CPT in the app stores with 86% but the apps developed by AdobeAir CPT have more downloads and reviews number. Despite that, higher downloads and reviews number doesn't indicate user satisfaction from the apps, PhoneGap CPT got better user reviews.

An experiment is done in (Tvilcek et al., 2017) [S14] by building a simple mobile app that does simple calculation which is adding two numbers. This app was developed using six different IDEs, Android Studio, Xcode, Visual Studio, Ionic, PhoneGap and NativeScript. In order to explore the advantages and disadvantages for these tools, they were compared according to the some criteria such as supported computer OS, supported mobile OS, programming languages and others. Android Studio has an advantage of ability to develop applications on all computer operating systems. Based on the analyzed criteria, hybrid development tools were considered as reliable tools, PhoneGap is one of the hybrid tools that has a good advantages due to number of supported mobile platforms.

(Majchrzak et al., 2017) [S15] assessed three cross-platform development tools through design-oriented experimental analysis combined with survey to support the findings. The study analyzed the ReactNative, Ionic and Fuse frameworks, they have been assessed according to survey result of 100 responses, and then, prototype app is designed based on the survey results. This practice-assessment analysis focused on the user experience through developing a real-world use case, it's also considered the remote fetching data, navigation and developer experience as a comparison factors. The analyzed cross-platform tools have common problems related to compatibility, performance, testability, and others. ReactNative developers had to handle many architectural decisions while the Ionic framework made it easier by providing the component library.

The experiment conducted by (Angulo et al., 2014) [S16] aimed to evaluate the impact of creating app using cross-platform tool on the UX. The study done by having two independent development teams, one developed a Titanium app version, and the other built two native versions. The evaluation results showed that there is no extreme differences in the UX between the developed versions, especially in the Android platform.

(Ali et al., 2017) [S17] implemented a comparative study using mixed-methods, quantitatively and qualitatively. They used open source tools to mine apps from both Google Play Store and Apple Store, 80K hybrid apps were collected from the stores. These apps were compared according to user reviews, versions, prices and other factors. The analysis was done in order to get a better understanding about the challenges that developers face in developing these kinds of apps, and what makes user rates the same app differently over different platforms. The results were concluded based on the qualitative feed backs from the users and the quantitative research questions. They found that the most developed apps are those built for specific platform, additionally, although most Android apps are free but the paid apps have higher prices than the iOS paid apps. More than 80% of the top-rated apps are the hybrid apps while the Android versions for the pair-apps have higher user rating.

After conducting an analysis study on hybrid apps based on developers' perception, (Malavolta et al., 2015) [S21] conducted another study to analyze the hybrid apps according to end users perception. They mined 11,917 apps from the Google Play Store with 3,041,315 user reviews, those apps have been analyzed by studying the users' reviews for the hybrid apps compared with the reviews for the native apps. The analysis done through four qualitative research questions, all of them focused on the users' perception of the hybrid apps with respect to their perception of the native ones. The study found that the hybrid development frameworks have a good perception when they used for mobile apps of data-intensive, but they perceived poorly when building apps that require an access to system features. The end user value for both native and hybrid is the same. Additionally, hybrid apps in some categories get lower perception than the native apps.

The study by (Ciman et al., 2016) [S25] conducted an experimental analysis to compare the existing cross-platform frameworks according to energy consumption. They tried to prove that the adoption of specific cross-platform development framework can affect the energy consumption. The experiment used hardware and software setups, and the used applications are for both Android and iOS. Each application is executed for 2 minutes and it's repeated for three times. The results confirmed that developing using cross-platform approach is absolutely reveals an increase in the energy consumption.

Additionally, the studies under comparative analysis category are classified into sub-categories according to the object involved in each study. The comparison in these studies is done

on different objects, these research objects are (I) approaches, which identify the differences between various development approaches, such as hybrid vs. native; (II) frameworks which investigated the developed apps based on the used framework, e.g. JavaScript frameworks vs. code translation frameworks; (III) tools, the evaluation is done based on the development tool used to build the applications such as PhoneGap vs. Titanium; (IV) application attributes, the apps are evaluated according to their attributes, e.g. user reviews and number of downloads. Table 3 shows each sub-category and its relevant studies.

Research Object	**Study**
Approaches	S2, S4, S5, S6, S10, S16, S25
Frameworks	S7, S15
Tools	S11, S12, S14
Applications Attributes	S17, S21

Table 3: Comparative analysis sub-categories

2.4.3.2 *Frameworks Analysis*

Part of the applied studies analyzed the cross-platform development in different ways, some conducted a survey or a case study that focused on the development process, and others analyzed the developed hybrid apps. Mainly, the purpose for those kinds of studies was to identify the challenges and opportunities for hybrid development approach. In this section, seven relevant studies will be discussed S1, S3, S9, S13, S18, S23 and S26.

The study by (Martinez et al., 2017) [S1] analyzed the process of building cross-platform mobile apps, studied the maintenance process for the cross-platform apps in terms of frequent discovered bugs and how they are fixed, finally, the study tried to define a tool to automatically fix the bugs. The analysis study is done through an experiment which built a mobile apps using Xamarine and React-Native frameworks.

(Amatya et al., 2013) [S3] conducted a literature survey study in order to assess the trend for cross-platform development in the last few years. They used the keywords of “mobile” and “cross-platform” to mine the relevant research papers from ACM and IEEE libraries, then, they applied some inclusion criteria, and left with 17 articles. Finally, they showed the research results by describing the cross-platform development problems that each paper addressed, and the

proposed solutions with the used technologies to handle the problem. The results showed that despite that the cross-platform development tools are not highly powerful yet, but they are promising. The impact of using the multi-platform development tool on the users' reviews is studied by (Mercado et al., 2016) [S9]. This study attempted to find out if there was any relationship between the specific development tool and the user's perception which reflects the quality of the application. They mined 50 apps developed using three different hybrid approaches, and analyzed 787,228 user reviews in terms of security, performance, reliability and usability. They found that is selecting the development approach is absolutely affects the users' perception.

The study by (Dunka et al., 2017) [S13] investigated developing hybrid mobile apps using Ionic framework. They analyzed the Native, Web and Hybrid mobile development approaches before explaining how the apps can be developed using Ionic framework technologies. The technical details presented in this study can be used as a reference for building enterprise level application. The analysis study applied in (Vitos et al., 2014) [S18] aimed to address the issues for building hybrid apps using PhoneGap, and tried to provide solutions for these issues. The analysis has been done by using the PhoneGap tool to implement features of Baltic Insurance House mobile application, the development covered different functionalities. The result confirmed that several points should be considered when choosing PhoneGap as the development tool.

The study by (El-Kassas et al., 2015) [S23] aimed to provide an overview for the available cross-platform approaches. They described each approach in details which are, Cross-Compilation, Virtual Machine, Model-Driven Development, Web-Based, Component-Based, Cloud-Based and Merged approach. The study analyzed the pros and cons for them, and proposed a categorization for many approaches. (Bioren-Hansen et al., 2017) [S26] surveyed the literature and introduced a technical baseline to be followed in the future research in the cross-platform development field. They suggested some approaches and frameworks to be included in the research such as Model-Driven and Component-Based approaches. The suggestions are formed based on the traversed literature, and considered the most common discussed in the previous research, hence, researchers can avoid the deprecation.

2.4.3.3 *New Approach*

Some studies proposed new approaches to develop hybrid mobile apps, they presented the technical and architectural details for these approaches. The studies presented in this section are S8, S19, S20, S22, S24, S27, S28, S29 and S30.

(Heitkotter et al., 2013) [S8] introduced a model-driven approach md2 for developing cross-platform apps. This approach helps the developers by generating the native code automatically after only specifying the high-level design for the app. They described the general concepts of the model then demonstrated the technical details of how the model can be implemented. Md2 has been discussed according to real-world project and proved that it's helpful for app development approaches.

The approach proposed by (El-Kassas et al., 2014) [S19] is a mixed solution between some different cross-platform development approaches. They combined the advantages from those mixed approaches, and tried to reduce their drawbacks. The proposed architecture with name of Integrated Cross-Platform Mobile Development (ICPMD) supported the source code availability (e.g. iOS), and of course the ability to run it on different platforms.

An evaluation framework analysis is presented in (Ahti et al., 2016) [S20], it has been validated through an experiment in order to propose a decision making helper. The framework validation aimed also to analyze a cross-platform tool, PhoneGap, by comparing it with native development tool for Android and Windows Phone platforms. They developed an application with two versions for Android and Windows Phone using native development tool, and one hybrid version using PhoneGap. The proposed evaluation framework is used to assess these applications by comparing the native version against the hybrid version. This experiment reveals that developing the hybrid version was easier than the native ones, but it showed an average quality in appearance and usability.

(Brucker et al., 2016) [S22] presented an approach of static code analysis which is a technique to detect the data-flows in hybrid mobile applications by generating calls graphs. This approach has been evaluated by applying it on apps mined from Google Play, these apps were developed using Cordova framework and ranked as top Cordova apps in the store. The results showed the generated calls graphs for cross-platform language are highly precise. The quality depends in the used language, which is JavaScript in this case.

The study by (El-Kassas et al., 2016) [S24] extends [S19] by introducing a new code conversion approach, this approach proposed a way to find a matching for set of code patterns inside the input source code in order to produce the required source code for a specific platform. The study evaluated the efficiency of the generated apps from the new version of ICPMD compared with the apps developed using other cross-platform tools. The results showed a remarkable enhancement in the speed, memory usage, and the app size.

A Native-2-Native approach is proposed by (Byalik et al., 2015) [S27], it is a method to transform the source code of a specific platform to be compatible with another platform. This technique is applicable only for transforming the Android source code (Java) into iOS version (Swift). The code transforming done via plugin added to the Eclipse IDE, it's evaluated by applying the method on several Android/Java APIs. The results indicate that Native-2-Native is a promising tool to support cross-platform development.

(Chadha et al., 2016) [S28] applied a study that extends [S27], it enhanced the proposed technique of (Native-2-Native) by made it able to transform the native source code from iOS/Swift into the Android version (Java). They improved the Eclipse IDE plugin to detect the code in Java or Swift that access the native resources, then create a query to get a resource in web-based programming that can access the equivalent native resource in Android or iOS platforms. The enhancement has been evaluated, it's also showed a valuable development tool.

The study done by (Tung et al., 2018) [S29] proposed a library to facilitate the development of cross-platform apps with acoustic sensing. The solution library of name LibAcousticSensing (LibAS) has been evaluated in this study by developing three aps using it. These developed apps covered the main aspects of acoustic sensing such as sound fingerprinting, inter-device interaction and others. The evaluation findings showed that LibAS reduced the required lines code, hence, less effort and less development time.

The proposed framework by (Boushehrinejadmoradi et al., 2015) [S30] is to test the development frameworks used to build cross-platform apps. The study built a prototype of the suggested tool with name of X-Checker, then tried it on the Xamarine framework, it caught 47 bugs in that framework. X-Checker tool is designed based on differential testing method which has been proved that it's an effective testing method.

The proposed solutions introduced different types of new approaches, some of them proposed a development framework, others suggested an evaluation framework that can help in

evaluating the developed apps using cross-platform frameworks, and one study proposed a testing framework. A sub-categorization is showed in Table 4 according to the type of suggested framework.

Framework Type	**Studies (S)**
Development Framework	S8, S19, S24, S27, S28, S29
Evaluation Framework	S20, S22
Testing Framework	S30

Table 4: New approach sub-categories

2.4.4 *Research Accomplishments and Contribution Facets*

Sub-RQ1: What accomplishments do these studies present, and what contribution facets do they provide?

The achievements for the presented research covered several aspects, several studies compared cross-platform development approaches and tools against each other. The comparison done according to the most important factors that developers should consider when developing a hybrid app, and they also affect the user perception of the app. The comparative studies results explained the advantages and disadvantages for the compared approaches or frameworks.

Part of the included studies analyzed specific development framework, they showed technical details about how apps can be developed using these frameworks. Moreover, Pros and Cons for each framework are explained. The comparative and analysis studies aimed to be used as a reference for the developers, they can benefit from the evaluation results that studies provide. Additionally, through the technical details and recommendations that have been included in the studies.

Some other studies proposed a solution in the field of cross-platform apps, most of the solutions have been validated through an experiment or other research method. The suggestions differ from each other, the majority of them proposed a development framework, two of them proposed an evaluation framework, and only one suggested a testing framework.

The presented achievements for the mapped studies are classified according to the research approaches, which is summarized in Table 5. The research approaches have been decided based on the conducted studies. Three research approaches are identified, evaluation research, validation research, and solution research. Most of the studies were conducted using evaluation research approach (73%).

Research Approach	Studies (S)	# of studies
Evaluation Research	S1, S2, S3, S4, S5, S6, S7, S9, S10, S11, S12, S13, S14, S15, S16, S17, S18, S19, S21, S23, S25, S26	22
Validation Research	S20, S22, S24, S27, S28, S29, S30	7
Solution Research	S8	1

Table 5: Research approach facets

The contribution facets classification described in this section was inspired by other mapping study (Zein et al., 2016). The facets were classified according to their contribution into the following groups: evaluation, framework, review, tool, method and metrics. The majority of the studies are classified under evaluation, they represented comparative studies to assess the approaches and the tools of the cross-platform development. Framework represents a detailed technique of wide purpose and covers several research questions. Review studies surveyed the literature to provide an overview for a specific area such as development approaches (Shahrokni, 2013). A tool provides specific software for development or testing purposes. Methods explain certain goal and research question. Metrics, which we have only one metrics study in our mapping study, show a guideline to be followed in a specific area in order to achieve the goal. Figure 3 shows the distribution of contribution facets, and Table 6 presents a summary of the contribution facets and their relevant studies.

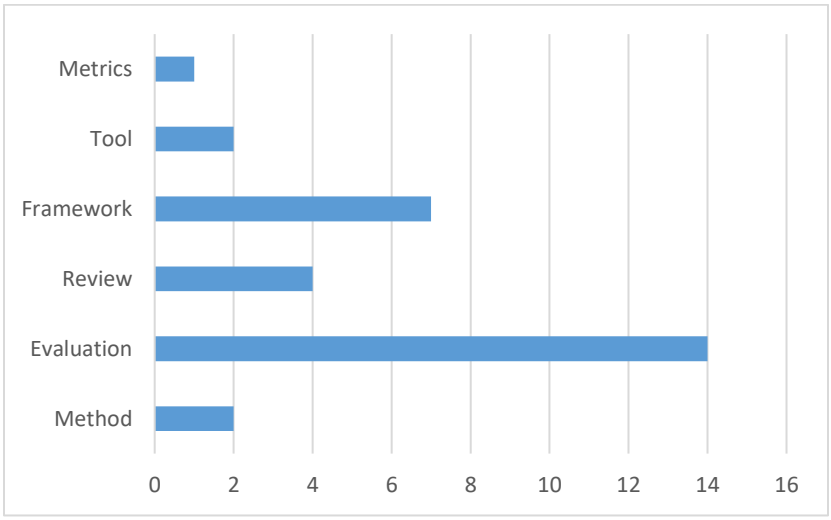

Figure 3: Distribution of contribution facets

Contribution Facet Type	Studies (S)	Description
Metrics	S26	Baseline to be followed in the future research in the cross-platform development field
Tool	S13	Ionic development tool to build hybrid apps for enterprise level
	S30	X-Checker tool to test the development frameworks used to build cross-platform apps
Framework	S8	Model-driven approach md2 for developing cross-platform apps by generating the native code automatically after only specifying the high-level design for the app
	S19	ICPMD is a development framework that supports the source code availability and the ability to run it on different platforms
	S20	Evaluation framework to assess the developed applications by comparing the native version against the hybrid version
	S24	Enhanced ICPMD, new code conversion approach to find a matching for set of code patterns inside the input source code
	S27	Native-2-Native approach to transform the source code of a specific platform to be compatible with another platform, only from Android/Java into iOS/Swift
	S28	Enhanced Native-2-Native approach by made it able to transform the native source code from iOS/Swift into the Android version (Java)
	S29	LibAcousticSensing (LibAS) a library to facilitate the development of cross-platform apps with acoustic sensing
Review	S3	Conducted a literature survey study in order to assess the trend for cross-platform development in the last few years
	S9	The study attempted to find out if there an impact of using the multi-platform development tool on the users' reviews
	S18	Analysis study aimed to address the issues for building hybrid apps using PhoneGap
	S23	Provides an overview for these cross-platform approaches, Cross-Compilation, Virtual Machine, Model-Driven Development, Web-Based, Component-Based, Cloud-Based and Merged approach
Evaluation	S2	Analyzed different development approaches that can be used to achieve the platform independence
	S4	Presented a comparison of cross-platform development approaches, web, hybrid, interpreted, cross-compiled and

		model driven approach. Discussed the desirable features for the eligible development tool
	S5	Evaluated the development approaches of cross-platform apps which are the Web apps and the apps developed by PhoneGap and Titanium
	S6	A comparative analysis that is done based on the historical review of some app types which are web, hybrid, interpreted and generated apps
	S7	Comparison study to measure the performance of the cross-platform development tools based on the response time, CPU usage, memory usage and disk usage
	S10	Analysis of cross-platform approaches which are mobile web, hybrid, interpreted and cross-compiled, it evaluated the most important features of those development approaches
	S11	The study evaluated PhoneGap, Titanium and Sencha Touch tools in terms of CPU, memory usage and power consumption
	S12	Compared hybrid apps that have been developed using different cross-platform tools
	S14	Comparative experiment by building a simple mobile app that was developed using six different IDEs (Native and Hybrid)
	S15	Assessed three cross-platform development tools which are ReactNative, Ionic and Fuse frameworks
	S16	Conducted an experiment aimed to evaluate the impact of creating app using cross-platform tool on the UX
	S17	Analysis done in order to get a better understanding about the challenges that developers face in developing hybrid apps
	S21	Comparative study to analyze the hybrid apps according to end users perception (user' reviews)
	S25	Experimental analysis to compare the existing cross-platform frameworks according to energy consumption
Method	S1	Analyzed the process of building cross-platform mobile apps then tried to define a tool to automatically fix the bugs
	S22	Presented an approach of static code analysis which is a technique to detect the data-flows in hybrid mobile applications by generating calls graphs

Table 6: Contribution facets and the mapped studies

The bubble chart below (Figure 4) presents the number of papers grouped under the main categories, against the papers categorized based on the contribution facets.

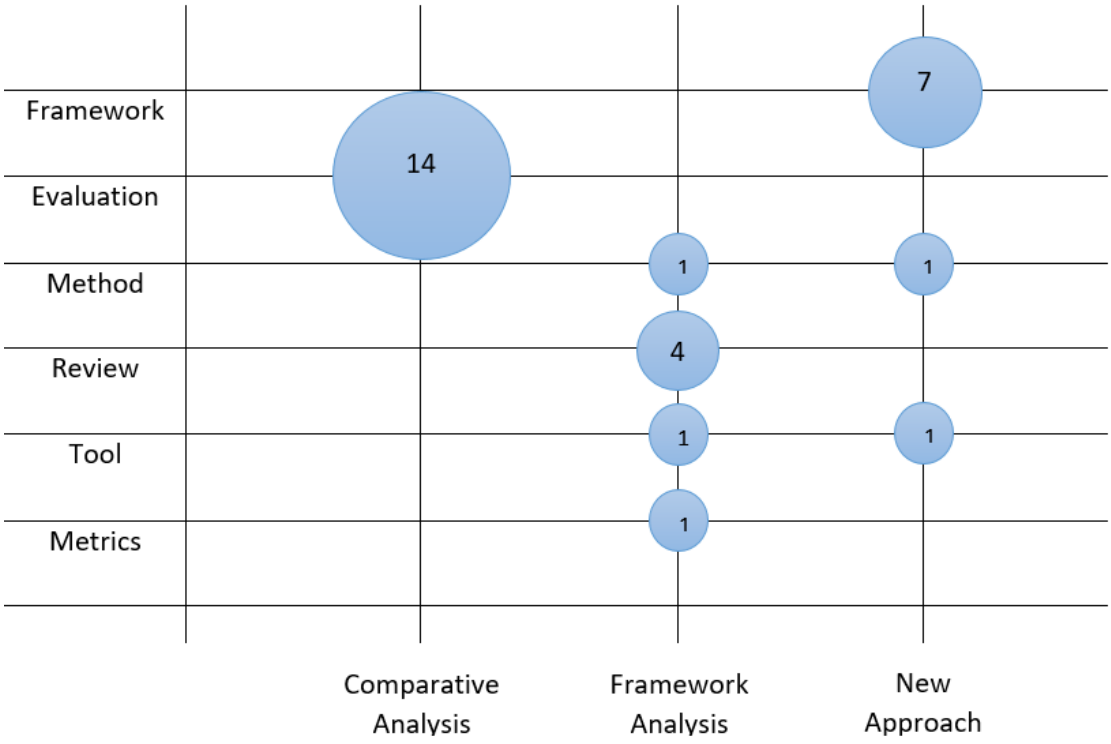

Figure 4: Contribution facets against main categories

2.4.5 *Main Challenges*

Sub-RQ2: What are the main challenges addressed by these studies?

We have presented three main categories for the covered studies in this mapping study, which are comparative analysis studies, framework analysis studies, and new approach studies. The studies under each category addressed some relevant challenges, they are summarized in Table 7.

Category	**Challenge(s) addressed**
Comparative Analysis	• Achieve UI consistency • Target the user experience of native app level • Build a maintainable app • Build an application with a satisfactory level of performance
Framework Analysis	• Create a real tool based on the existing framework to be used by the developers • Automatically integrate the generated code
New Approach	• Support multi types of applications

	• The ability to transform the code from the native source with considering the UI translation

Table 7: Addressed main challenges

2.4.6 *Main Techniques*

Sub-RQ3: What are the main research techniques applied by these studies?

Among 30 studies, 19 of them applied an experiment to come up with findings by either answering research questions or investigate hypothesis. Moreover, there are another two studies that combined the experiment with survey, which means that experimental research formed the majority of the presented studies which is around 70%. Oppositely, only one study conducted interviews in its research process. Survey is conducted in another two studies, while case study is applied in only two studies. There are 4 studies that have reviewed the literature in order to give an overview for the existing studies. Table 8 shows the summary of the studies and applied techniques.

Research Technique	**Studies (S)**	
Experiment	S1, S2, S7, S8, S9, S12, S13, S14, S19, S20, S21, S22, S24, S25, S27, S28, S29, S30	
Survey	S4, S11	
Case study	S10, S18	
Interviews	S5	
Mixed	S15, S17	Experiment + Survey
Literature Review	S4, S7, S24, S26	

Table 8: Applied techniques

2.5 *Discussion*

This systematic mapping study aims to create a classification scheme and to collect, interpret and analyze all the relevant studies in the area of cross-platform mobile apps development. We found that there is no comprehensive systematic mapping study in this area, so, conducting such a study will help in improving the knowledge and spot the light on the gaps in the field.

The mapped studies showed some research gaps that require more investigation. First, most of the studies that were categorized under new approach category (see Table 2) validate the

proposed solutions on applications that have been built specifically for the validation purpose. If these studies were based on evaluating the solutions using real and more complex applications, then the scientific value of these studies would be more convincing and comprehensive. However, there are some of these studies that used real applications mined from the app stores, but these involved apps were simple and don't represent the real-world applications. It's highly recommended to validate the proposed solutions and frameworks on more complex applications.

Secondly, the majority of the studies categorized under comparative analysis category conducted a comparison process based on the results of using native vs. hybrid mobile apps. However, these studies didn't provide a detailed information about how they distinguish between the apps types. More specifically, these studies did not explain how they were able to identify whether an app, which is downloaded from an online store, to be native or hybrid. If these studies gave the exact steps about how the differentiating process was done, then these studies would be more trusted and replicable.

Further, we found that there is a lack of studies focusing the testing challenges and solutions for cross-platform mobile apps. We found little studies that mentioned this challenge briefly, and one study [S30] that proposed a testing framework solution for the developed apps. Testing hybrid apps is an area that requires more investigation since there are several points should be considered when testing hybrid app. For instance, writing automation test cases can be easy if the app is built using only web based technologies and with little native code, one test script can run across all supported platforms. On the other hand, how the testing process will be conducted when the app is developed with totally different languages? One possible solution is that each supported platform should have its own tests scripts. Such an issue and other testing issues should be further investigated in the field of testing cross-platform apps.

Finally, little studies investigate the maintenance challenges and solutions for the cross-platform apps. Maintainability can largely affect the developers' choice when deciding the approach to develop a mobile app. Additionally, mobile apps are rapidly changing due to the fact that they should be up to date to stay competitive, which is achieved by adding new features and improve the existing ones. Therefore, we believe that researchers should contribute better in this area to create a reference for the developers as several aspects can be covered and investigated. The literature should show the developers how the cross-platform app affects the maintenance cost in terms of effort and time. Moreover, the research can cover the required skills for maintaining

hybrid apps, whether the developers need specialized skills or not. Lower cost, less maintenance duration, and minimal learning curve in the maintenance process are sufficient factors to develop apps using cross-platform approach.

2.6 *Summary*

Chapter 2 presents a review of the state of knowledge of empirical studies in the area of mobile cross-platform app development. After analyzing 295,816 studies and applying multi-step inclusion/exclusion criteria, thirty (30) studies were finally included and mapped to a classification schema.

The classification contained four categories (1) structure of the topic; (2) contribution facets; (3) applied techniques; and (4) research facets. The category of structure of the topic is defined based on the thematic analysis, and is classified into three sub-categories: comparative analysis, framework analysis, and new approach. The contribution facets group included six categories: metrics, tool, framework, review, evaluation, and method. The third category which is the techniques applied in the selected studies categorized the empirical studies based on the research methods they used, experiment, survey, case study, interviews, and mixed methods. The category of research facets classified the studies based on the research facets. It contains three research facets, validation research, evaluation research, and solution research.

Two main research gaps were identified. First, there is a lack of studies focusing on testing challenges and solutions for mobile cross-platform app development. Secondly, little studies focus on the maintenance challenges. Both challenges are only mentioned briefly in the studies without conducting an empirical research.

Chapter 3 Research Methodology

The research applied a qualitative approach. The data is collected using multiple case studies which included different research methods, interviews and focus groups. The gathered data was in a form of written notes and recorded audios.

The qualitative approach investigates the issues of cross-platform apps development. These kinds of research are exploratory in nature which provides an in-depth information for specific case and give a chance to enrich the understanding of some problem (Jackson et al., 2007). Qualitative research is intended to cover the behavior and perceptions for the targeted participants, it is successful in recognizing important factors that affect the research issue such as social norms, gender roles, religion and others. Through qualitative research, we can get a comprehensive textual description of individuals' experience in the research problem.

In this thesis, the qualitative approach is used due to its flexibility which suits the defined research problem, unlike the quantitative methods that restrict the researcher with many aspects. For example, the quantitative research design should be fixed from the beginning, while the qualitative allows to change or add into the study design. Moreover, the questions in the quantitative methods are predefined and usually they are closed-ended, also the participants' responses don't affect them, while the responses in the qualitative approach can affect the next questions, and the questions format are open-ended. Also, the gathered data through quantitative methods are numerical, which the researcher gives a numeric values to the responses, while the qualitative responses are textual in nature.

In addition to qualitative approach flexibility, it also provides a less formal relationship between the researcher and the participant, which gives the participants a chance to be more cooperative and share more details that they share in the quantitative methods. Additionally, the researcher gets an opportunity to have immediate responses from the participants.

3.1 *Case Studies*

Case study is a qualitative research method that facilitates the investigation of contemporary phenomena within their contexts (Baxter, 2008), it provides an in-depth understanding for the phenomena, also the nature of the qualitative case studies allows researchers to study the phenomena using different data sources (Yin, 2009).

The case study subject can be individual, organization, place, action, and etc. It can be conducted in both qualitative and quantitative research methods, but in our research we will conduct the case study qualitatively and our subject is organizations.

This study aimed to explore the cross-platform approach case in real-life, therefore, the research applied a design of a qualitative multiple case study. This design gives an opportunity to describe the case using a variety of industrial resources, which is done on multiple different data resources to be able to replicate the results across those resources and support the results revealed from each case. The data resources in this study represent Palestinian companies that are using cross-platform development approach.

3.2 *Data Collection*

The data has been collected through interviews and focus groups. In the beginning, a one-to-one interviews were hold with all team members which conducted as face-to-face and online meetings .These kinds of interviews give the interviewees a free space to explain the situation. Then, a focus group interviews were conducted with the whole development team, each team will was interviewed in one meeting. The discussion meeting was to gather deeper information.

The multiple research methods applied in this study is used to triangulate the collected data. Data triangulation is about verifying the findings with several sources of data, this helps in better understanding for the situation, in addition to confirm and enrich the findings obtained from each case study (Yin, 2009). Using data triangulation can improve the validity of the research study since the collected data is examined by multiple data sources. Moreover, triangulation guarantees that almost there is no bias in the data, and examines the data consistency.

3.2.1 Interviews

This study conducted a semi-structured interviews, where the author didn't strictly follow a formal questions list, but open-ended questions have been asked based on the discussion. The interviews have been conducted as face-to-face, online and in groups.

The interview is a conversation that conducted between the researcher and the interviewees in order to elicit information. It's a qualitative approach where the researcher ask the relevant questions which make him able to explore the participants thinking. Interviews flexibility offers the ability to adjust the questions according to the interviewees' reactions. Through interviews, the

researchers are able to illustrate the questions and investigate the answers. This method is preferred when it's difficult to observe the interviewees directly, also, it's used for data triangulation, so interviews questions verify the findings revealed from another data collection method. Researchers can lead the interviews meetings face-to-face, via telephone, or within groups (Kawulich, 2012).

3.2.2 Focus Group

Focus group is a qualitative research method which is about conducting interviews with group of people rather than meet them individually. This kind of data collection technique facilitates the discussion, and it's preferred to be used when there is a viewpoints diversity among the participants, since the views variety of the group can bring out information that may not have been explored in the individual interviews (Cohen et al., 2000). Moreover, it's easy to conduct since it requires less preparation than other data collection methods. Groups' discussion help in investigating the issues deeply, and the outcomes can be generated quickly. Group meeting can be an interesting activity for the members who get the opportunity to hear the others' perspectives and share their own. Through focus group meetings, we can pick potential participants for later individual interviews.

3.3 *Data Analysis*

The variety of data collection methods help in applying data triangulation which is according to (Yin, 2009) facilitates the data validation using different data collection techniques over multiple data resources. The collected data will be analyzed in a way that facilitate retrieving the relevant passages, we used a thematic coding technique (Yin, 2009) to classify the passages according to predefined codes and generate the themes.

3.3.1 Thematic Coding

It's a form of a qualitative research analysis, it inspects the themes of the given data which describes the research phenomenon. The coding analysis requires to be familiar with the data in order to extract the patterns, so it makes the researchers able to get deep insight from the gathered information. This kind of analysis is applicable for research questions and for large data set since it helps in narrowing the wide data through discovering themes, and then specifying the research

questions (Morgan el al., 1998). This research applied the thematic coding which the analysis process was as following:

- The case study material outcomes were used to extract the patterns (themes).
- Next, the extracted themes are defined with names (codes)
- Then, will be stored in order to be used in data triangulation by analyzing the whole case study data together.

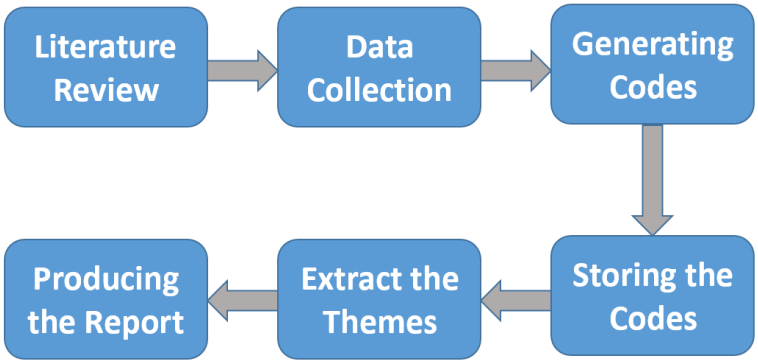

Figure 5: Thematic coding analysis steps

Chapter 4 Results

4. Results

The results section introduced the results of the case studies. This empirical study investigated the research on four different companies that are referred to them as C1, C2, C3 and C4 in order to maintain confidentiality. The first case C1 was a large outsourcing software and IT company, that employs more than 250 employees. C1 provides several software services such as mobile development, web-based solutions, quality assurance and testing and other services. C1 has one team of three members who develop cross-platform mobile applications using React-Native technologies. They provide mobile applications based on the client's requirements.

The second case C2 was an outsourcing software company that deliver customized mobile and web application solutions to organizations. C2 has two teams, each team of two members who are developing cross-platform mobile applications. One of the teams develop the apps using Ionic framework and the other is working using React-Native technologies.

The third case C3 is a company that provides technical services to clients outside the country. They are a team of three that work in building hybrid apps based on Ionic framework.

The last case C4 is a software development start-up that builds web and mobile applications for clients worldwide. They have a team of two members that working on cross-platform mobile development, they are specializes in developing hybrid apps using Ionic framework.

This section provides the results revealed from the interviews conducted with the developers.

The studied cases are all outsourcing companies where the client participates sometimes in the decision of development approach selection. The development teams in these companies build the mobile applications based on the client requirements, and several factors are taken in consideration in order for the appropriate development approach to be chosen.

It's noticeable that when the client has a technical background then he can suggest the development approach to be used, but we noticed also that the main factor affects the decision is the team members experience. When they are experienced in web technologies then it's easier for them to use cross-platform technologies. This decision is also depends on additional important factors, such as the client budget. Since the native development process is more costly, which is due to the number of resources it needs and the time it takes, so when the client budget is limited then it's more suitable to develop the application using cross-platform technologies. The project

requirements may strict the development approach choice. The majority of the developers confirmed that when most of the application features require a heavy hardware resources access, then it's better to go native. This is confirmed by a developer in C3:

"*If the app requires high access on device resources (ex: GPS) then we go native since it will be complicated in cross-platform*" – Developer, C3.

By contrast, few developers stated that they take the cross-platform approach even if there are several features require device resources access. They justified that everything can be developed using the libraries provided by the community.

Additional factor that is studied when choosing the development approach is the required performance from the application. If the app is a real-time and interacting application (ex: games), then the developers use the native technologies to build it.

After choosing the development approach and when the cross-platform approach is the taken choice for developing the app. The developers have to take the decision about which cross-platform technology to use. As mentioned earlier, the companies we studied are using either React-Native or Ionic technology.

Development teams agreed that they select the cross-platform tool based on the developers experience in order to reduce the learning time. For example, one of the team members in C3 mentioned that "*developers with Angular experience usually work on Ionic framework*" – Developer, C3. Moreover, the time limitation plays a role in cross-platform tool selection since the progress time is not the same for all tools, the team leader in C1 highlighted that "*React-Native development takes time much more than Ionic*" – Team Leader, C1.

Additionally, client requirements affect this decision, we noticed that when the application needs to interact with online web service then they choose to work with Ionic, this is explained by one of the developers in C1:

"*If the application is local, then we use Ionic since we need plugins to get data from backend when developing with React-Native, while HttpClient is supported in Angular that Ionic is built on*" – Developer, C1.

However, it was noticed that React-Native is the most candidate technology when the app is required to be with high performance, one of the developers in C2 stated that "*React-Native is better in performance wise in term of rendering and memory allocation*" – Developer, C2.

Despite previous developers' statements, they all agreed that React-Native is the dominant technology in general. The team leader in C4 who is leading a team that working on Ionic confirmed that they started learning React-Native in order to support it in the future.

This study found that there are common challenges faced by most of the development teams which restrict the development process or cause an overhead for the developers. Hardware calls is a major challenge that forces the developers either to build a native module to handle it or change the whole development approach. Another common issue is about the published libraries by the community, if the one who built the library stopped supporting it then it may cause problems in the recent platform versions. One of the developers in C1 mentioned that "*The written libraries may become unsupported by the persons who wrote them, so the library could not support latest platform versions*" – Developer, C1. Plugins can cause conflict issue when several number of them are injected into the app, and they are sometimes the reason for rejecting the app publishing in the app store due to something included in these plugins that don't comply with the store policies.

The current research revealed that the cross-platform community is not mature enough, the developers face an issue with libraries documentation which is not sufficient, also the updates and bugs fixes for these libraries are slow. The team leader in C4 stated that "*The Ionic community didn't expand as they promised in early stages*" – Team leader, C4.

In general, all the developers in the cases we investigated confirmed that the cross-platform app is maintainable since the code fix or update is working smoothly on different platforms, in other words, one update for all platforms which reduces the change time. But one of the developers in C2 explained that maintainability is different for the application that is written fully using cross-platform technology from the application that has many included native code.

"*If there are many native modules injected in the cross-platform app then its maintainability is the same as native app*" – Developer, C2.

It was noticed that the dominant supported testing technique for cross-platform apps is the manual testing. Most of the studied companies don't support automation testing (except one team in C2), they explained that they avoid automation testing due to its time and resources cost. A developer in C1 stated that "*Automated tests may be written in enterprises level or in mature startups*" – Developer, C1.

The team in C2 that support the automation testing explained that they started supporting the UI automation and unit testing recently, they use [1]Appium tool for UI tests and [2]Jest for unit testing. They face some issues with Appium when handling the UI components, it behaves different when access the components on Android device from the access way on iOS device. For example, the tool can access the component on Android device even if it's in the bottom of the page without scrolling, while it cannot do this on iOS without scrolling the page.

[1] Open source test automation framework for use with native, hybrid and mobile web apps
[2] JavaScript Testing Framework with a focus on simplicity

Chapter 5 Discussion

Chapter 5 interprets the case studies results in order to provide a better understanding of the research and to answer the research questions. The findings discussion is made based on the results described in the previous section and according to the revealed themes. First, it displays the development process in the field of cross-platform development, then it presents the most used technologies in industry, after that it moved to focus on the challenges that developers face. Moreover, the used testing techniques and their limitations will be presented, in addition to the maintainability side of the cross-platform apps.

Only one research is designed to target the cross-platform development in industry, this is a quantitative study that is done by (B-Hansen et al., 2019) and based on survey results of five questions which conducted with 101 participants. B-Hansan et al. study focused on the cross-platform technology adaption and the challenges faced by the developers.

However, the current study is a qualitative research that targets the whole development process in the field of cross-platform development, which covers the process and all the factors that affect the adaption of the development approach and cross-platform technologies. Also, the challenges that commonly reported by the industry are discussed, in addition to discuss the cross-platform app maintainability and the used testing techniques in the field. We designed a multiple-case study with conducting interviews and focus groups with 4 companies in order to get a solid understanding about the development process and other related aspects in cross-platform area.

5.1 Development Process

According to industry's practices, it is apparent that the companies' decision about which development approach to choose is often taken by the developers, although the clients can sometimes force the developers to work on a specific approach, but what really matters to them is the final delivered application regardless what are the technologies used to build it.

There are several factors to determine whether cross-platform technologies are suitable or not for developing the app such as the application requirements, client budget, the time specified for the project and other factors, but this decision is often made based on the developers' experience. Cross-platform technologies are always the most candidate in this decision, since these tools and frameworks depend on web technologies which are older technologies more than the native mobile development technologies, so web developers are highly available.

Although the practitioners mentioned all the factors that affect the decision which are confirmed by the researchers in the studies such as (Raj et al., 2012), but the conducted research didn't include that the developers' experience is one of the factors that affect choosing the development approach and tool, also it's considered as one of the most influential factors. From the researchers' point of view, the application requirements is the primary selection reason.

However, the factor that may force the developers to go for native is when the application has many features need to call device resources such as GPS, Bluetooth, and etc, which is confirmed by (Raj et al., 2012). In this case, the developers have to write many native modules to handle these calls, this workaround takes time that native approach can save by providing the resources calls in an easy and direct way. On the other hand, these injected native modules can complicate the code maintenance process, because developers will have to update each module in different way to work on all platforms.

5.2 Cross-platform Tools

There are several cross-platform tools options to develop mobile apps. Although the number of companies that use Ionic equals to those who are using [3]React-Native in this study, but based on the practitioners' sayings, the Ionic framework is currently used more than React-Native. This fact is because the Ionic framework is older than the React-Native. But, from the practitioners' point of view, the React-Native is a promising technology since it's open-source in addition to the fact that its community is rich and expanding in a satisfactory way. There is a lack of studies that investigate the React-Native since it's a new technology, but the recent research agree with the practitioners' opinion according to React-Native popularity. An experimental study (Martinez et al., 2017) mentioned that the reason for React-Native popularity is the strong participation in its community in addition to the development team provided by Facebook to evolve the framework. Hence, the potential cause of React-Native customers' growth and community expanding is Facebook's support, with Facebook recently announcing that they will invest more in the open source community of React-Native.

5.3 Challenges

Several previous research such as (Amatya et al., 2013), (Malavolta et al., 2015) and (B-Hansen et al., 2019) introduced the challenges that developers face in the cross-platform

[3] Open source mobile application framework created by Facebook. It is used to develop applications for Android, iOS and UWP

development field, these challenges include fragmentation, performance, user experience and hardware resources calls. Almost all the challenges that revealed from the current industrial research are mentioned in the earlier research, but to our knowledge, this research is the first at investigating the issue of the written libraries that provided from the community which forms a real challenge for the practitioners. The community may stop support these libraries, this forces the developers to maintain them which costs time.

5.4 Maintenance

The maintainability of cross-platform app was studies in (Martinez, 2017), the researchers concluded that the maintenance process is simplified and the cost is reduced in the cross-platform development, and this is due to the shared code that run on all the mobile platforms. This is confirmed by the industrial developers, but they also clarified that the maintenance process will be complex as native maintenance if the application has many injected native modules.

5.5 Testing

In general, the development life cycle ends with testing phase, where the testing process can be applied in several ways. Automation testing is usually used to reduce the testing time and effort, but according to the studied cases in this research, automation testing is not supported in the majority of companies in the cross-platform development field, this is due to the fact that automation testing needs more resources while the companies go forward cross-platform development in order to reduce the resources cost, so they depend on manual testing only. To our best of knowledge, testing cross-platform apps is not investigated in the literature, which makes this study the first research that discuss the testing in the industrial contexts.

Chapter 6 Conclusion

6. Conclusion

This report presented the work done in this thesis which is an industrial investigation of cross-platform mobile apps development. The main research proposition was that the cross-platform development isn't explored deeply in the industrial contexts of Palestinian IT companies, since there is a lack in studies that investigate it. This thesis is done based on the main proposition, in addition to the issues identified in the SMS, which is implemented through conducting interviews with development teams from four different companies. The conducted research is deeply explored how the development teams in the Palestinian IT companies develop mobile apps using the frameworks of the cross-platform technologies. It's concluded that the development approach and tool selection decision for developing an application depends on the developers' experience first then some other factors. Also, the testability and maintainability aspects in the development process have been investigated. In term of maintenance, the cross-platform technologies facilitate the maintenance process but it becomes complex when the app has many injected native modules. According to testing, the majority of the companies support the manual testing only.

Data Availability

The data used in this study to support the presented results have been pushed to GitHub repository as XLSX-based documents. The following link is the data repository location: **https://github.com/TasnimZuhod/thesis_data**.

References

1. Nagappan, M., & Shihab, E. (2016). Future Trends in Software Engineering Research for Mobile Apps. 2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER). doi:10.1109/saner.2016.88
2. Furlan, A. (2018, August 13). Cross Platform Mobile App Development Guide (2017). Retrieved from http://www.businessofapps.com/guide/cross-platform-mobile-app-development/
3. Latif, M., Lakhrissi, Y., Nfaoui, E., & Es-Sabi, N. (2017). Review of Mobile Cross Platform and Research Orientations. 2017 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS), doi:10.1109/wits.2017.7934674.
4. Ribeiro, A., & Rodrigues D. S., A. (2012). Survey on Cross-Platforms and Languages for Mobile Apps. 2012 Eighth International Conference on the Quality of Information and Communications Technology, doi:10.1109/quatic.2012.56.
5. S. Alamri., & A. Mustafa. (2014). Software Engineering Challenges in Multi Platform Mobile Application Development. Advanced Science Letters, vol. 20, no. 10, pp. 2115–2118., doi:10.1166/asl.2014.5671.
6. Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008). Systematic mapping studies in software engineering, 12th International Conference on Evaluation and Assessment in Software Engineering, p. 1.
7. Zein, S., Salleh, N., & Grundy, J. (2016). A Systematic Mapping Study of Mobile Application Testing Techniques.” The Journal of Systems & Software. doi:10.1016/j.jss.2016.03.065.
8. Kitchenham, B., & Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering. School of Computer Science and Mathematics, Keele University.
9. Felderer, Michael & C. Carver, Jeffrey. (2018). Guidelines for Systematic Mapping Studies in Security Engineering. Empirical Research for Software Security: Foundations and Experience.

10. Shahrokni, A., & Feldt, R. (2013). A systematic review of software robustness. Information and Software Technology 55, 1-17.
11. Martinez, M., & Lecomte, S. (2017). Towards the Quality Improvement of Cross-Platform Mobile Applications. 2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft). doi:10.1109/mobilesoft.2017.30
12. Taneja, K., & Taneja, H., & K. Bhullar, R. (2016). Cross-Platform Application Development for Smartphones: Approaches and Implications. doi:978-9-3805-4421-2.
13. Amatya, S., & Kurti, A. (2013). Cross-Platform Mobile Development Challenges and Opportunities. Springer International Publishing Switzerland. doi:10.1007/978-3-319-01466-1_21.
14. Latif, M., Lakhrissi, Y., Nfaoui, E. H., & Es-Sbai, N. (2016). Cross platform approach for mobile application development: A survey. 2016 International Conference on Information Technology for Organizations Development (IT4OD). doi:10.1109/it4od.2016.7479278
15. Heitkötter, H., Hanschke, S., & Majchrzak, T. A. (2013). Evaluating Cross-Platform Development Approaches for Mobile Applications. Lecture Notes in Business Information Processing Web Information Systems and Technologies, 120-138. doi:10.1007/978-3-642-36608-6_8
16. Xanthopoulos, S., & Xinogalos, S. (2013). A comparative analysis of cross-platform development approaches for mobile applications. Proceedings of the 6th Balkan Conference in Informatics on - BCI 13. doi:10.1145/2490257.2490292
17. Willocx, M., Vossaert, J., & Naessens, V. (2016). Comparing performance parameters of mobile app development strategies. Proceedings of the International Workshop on Mobile Software Engineering and Systems - MOBILESoft 16. doi:10.1145/2897073.2897092
18. Heitkötter, H., Majchrzak, T. A., & Kuchen, H. (2013). Cross-platform model-driven development of mobile applications with md2. Proceedings of the 28th Annual ACM Symposium on Applied Computing - SAC 13. doi:10.1145/2480362.2480464
19. Mercado, I. T., Munaiah, N., & Meneely, A. (2016). The impact of cross-platform development approaches for mobile applications from the users perspective. Proceedings

of the International Workshop on App Market Analytics - WAMA 2016. doi:10.1145/2993259.2993268

20. Delia, L., Galdamez, N., Thomas, P., Corbalan, L., & Pesado, P. (2015). Multi-Platform Mobile Application Development Analysis. doi:978-1-4673-6630-4/15.
21. Dalmasso, I., Datta, S. K., Bonnet, C., & Nikaein, N. (2013). Survey, comparison and evaluation of cross platform mobile application development tools. 2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC). doi:10.1109/iwcmc.2013.6583580
22. Ali, M., & Mesbah, A. (2016). Mining and characterizing hybrid apps. Proceedings of the International Workshop on App Market Analytics - WAMA 2016. doi:10.1145/2993259.2993263
23. Dunka, Bakwa & Emmanuel, Edim & Oyerinde, Dantala. (2017). HYBRID MOBILE APPLICATION BASED ON IONIC FRAMEWORK TECHNOLOGIES. International Journal of Recent Advances in Multidisciplinary Research. 04. 3121-3130.
24. Vilcek, T., & Jakopec, T. (2017). Comparative analysis of tools for development of native and hybrid mobile applications. 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). doi:10.23919/mipro.2017.7973662
25. Majchrzak, T., & Grønli, T. (2017). Comprehensive Analysis of Innovative Cross-Platform App Development Frameworks. Proceedings of the 50th Hawaii International Conference on System Sciences (2017). doi:10.24251/hicss.2017.745
26. Angulo, E., & Ferre, X. (2014). A Case Study on Cross-Platform Development Frameworks for Mobile Applications and UX. Proceedings of the XV International Conference on Human Computer Interaction - Interacción 14. doi:10.1145/2662253.2662280
27. Ali, M., Joorabchi, M. E., & Mesbah, A. (2017). Same App, Different App Stores: A Comparative Study. 2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft). doi:10.1109/mobilesoft.2017.3
28. VITOLS, Gatis, SMITS, I., & ZACEPIN, A. (2014). Issues of Hybrid Mobile Application Development with PhoneGap: a Case Study of Insurance Mobile Application. International Baltic Conference.

29. El-Kassas, W. S., Abdullah, B. A., Yousef, A. H., & Wahba, A. (2014). ICPMD: Integrated cross-platform mobile development solution. 2014 9th International Conference on Computer Engineering & Systems (ICCES). doi:10.1109/icces.2014.7030977
30. Ahti, V., Hyrynsalmi, S., & Nevalainen, O. (2016). An Evaluation Framework for Cross-Platform Mobile App Development Tools. Proceedings of the 17th International Conference on Computer Systems and Technologies 2016 - CompSysTech 16. doi:10.1145/2983468.2983484
31. Malavolta, I., Ruberto, S., Soru, T., & Terragni, V. (2015). End Users Perception of Hybrid Mobile Apps in the Google Play Store. 2015 IEEE International Conference on Mobile Services. doi:10.1109/mobserv.2015.14
32. Brucker, A. D., & Herzberg, M. (2016). On the Static Analysis of Hybrid Mobile Apps. Lecture Notes in Computer Science Engineering Secure Software and Systems, 72-88. doi:10.1007/978-3-319-30806-7_5
33. El-Kassas, W. S., Abdullah, B. A., Yousef, A. H., & Wahba, A. M. (2015). Taxonomy of Cross-Platform Mobile Applications Development Approaches. Ain Shams Engineering Journal, 8(2), 163-190. doi:10.1016/j.asej.2015.08.004
34. El-Kassas, W. S., Abdullah, B. A., Yousef, A. H., & Wahba, A. M. (2016). Enhanced Code Conversion Approach for the Integrated Cross-Platform Mobile Development (ICPMD). IEEE Transactions on Software Engineering, 42(11), 1036-1053. doi:10.1109/tse.2016.2543223
35. Ciman, M., & Gaggi, O. (2017). An empirical analysis of energy consumption of cross-platform frameworks for mobile development. Pervasive and Mobile Computing, 39, 214-230. doi:10.1016/j.pmcj.2016.10.004
36. Biørn-Hansen, Andreas & Grønli, Tor-Morten & Ghinea, Gheorghita. (2017). Baseline Requirements for Comparative Research on Cross-Platform Mobile Development.
37. Byalik, A., Chadha, S., & Tilevich, E. (2015). Native-2-native: Automated cross-platform code synthesis from web-based programming resources. ACM SIGPLAN Notices, 51(3), 99-108. doi:10.1145/2936314.2814210
38. Chadha, S., Byalik, A., Tilevich, E., & Rozovskaya, A. (2017). Facilitating the development of cross-platform software via automated code synthesis from web-based

programming resources. Computer Languages, Systems & Structures, 48, 3-19. doi:10.1016/j.cl.2016.08.005

39. Tung, Y., Bui, D., & Shin, K. G. (2018). Cross-Platform Support for Rapid Development of Mobile Acoustic Sensing Applications. Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services - MobiSys 18. doi:10.1145/3210240.3210312
40. Boushehrinejadmoradi, N., Ganapathy, V., Nagarakatte, S., & Iftode, L. (2015). Testing Cross-Platform Mobile App Development Frameworks (T). 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE). doi:10.1109/ase.2015.21
41. Jackson, R., Drummond, D., & Camara, S. (2007). What Is Qualitative Research? Qualitative Research Reports in Communication. doi:10.1080/17459430701617879
42. Baxter, P., & Jack, S. (2008). Qualitative case study methodology: Study design and implementation for novice researchers. The Qualitative Report, 13(4), 544-559. Retrieved from http://www.nova.edu/ssss/QR/QR13-4/baxter.pdf
43. Yin, R. K. (2009). Case study research: Design and methods. Los Angeles: Sage.
44. Kawulich, B. (2012). Collecting data through observation.
45. Cohen, L., Manion, L. & Morrison, K. (2000) Research Methods in Education 5th Edition. London: RoutledgeFalmer.
46. Morgan, D.L. & Kreuger, R.A. (1998) The Focus Group Kit. Thousand Oaks, Ca.
47. Guest, G., MacQueen, K. M., & Namey, E. E. (2012). Applied thematic analysis. Thousand Oaks, CA: Sage.
48. B-Hansen, A., Gronli, T., Ghinea, G., & Alouneh, S. (2019). An Empirical Study of Cross-Platform Mobile Development in Industry. Hindawi Wireless Communications and Mobile Computing. doi:https://doi.org/10.1155/2019/5743892.
49. Raj, C. R., & Tolety, S. B. (2012). A study on approaches to build cross-platform mobile applications and criteria to select appropriate approach. 2012 Annual IEEE India Conference (INDICON). doi:10.1109/indcon.2012.6420693
50. Nakazawa, C. (2019, March 1). React Native Open Source Update March 2019. Retrieved from https://facebook.github.io/react-native/blog/2019/03/01/react-native-open-source-update.

Appendix

LIST OF INCLUDED STUDIES

The references listed below are the studies that included in this mapping study which started with letter "S":

S1	Martinez, M., & Lecomte, S. (2017). Towards the Quality Improvement of Cross-Platform Mobile Applications. 2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft). doi:10.1109/mobilesoft.2017.30
S2	Taneja, Kavita, et al. "Cross-Platform Application Development for Smartphones: Approaches and Implications." 2016, doi:978-9-3805-4421-2.
S3	Amatya, Suyesh, and Arianit Kurti. "Cross-Platform Mobile Development Challenges and Opportunities." Springer International Publishing Switzerland, 2013, doi:10.1007/978-3-319-01466-1_21.
S4	Latif, M., Lakhrissi, Y., Nfaoui, E. H., & Es-Sbai, N. (2016). Cross platform approach for mobile application development: A survey. 2016 International Conference on Information Technology for Organizations Development (IT4OD). doi:10.1109/it4od.2016.7479278
S5	Heitkötter, H., Hanschke, S., & Majchrzak, T. A. (2013). Evaluating Cross-Platform Development Approaches for Mobile Applications. Lecture Notes in Business Information Processing Web Information Systems and Technologies, 120-138. doi:10.1007/978-3-642-36608-6_8
S6	Xanthopoulos, S., & Xinogalos, S. (2013). A comparative analysis of cross-platform development approaches for mobile applications. Proceedings of the 6th Balkan Conference in Informatics on - BCI 13. doi:10.1145/2490257.2490292
S7	Willocx, M., Vossaert, J., & Naessens, V. (2016). Comparing performance parameters of mobile app development strategies. Proceedings of the International Workshop on Mobile Software Engineering and Systems - MOBILESoft 16. doi:10.1145/2897073.2897092
S8	Heitkötter, H., Majchrzak, T. A., & Kuchen, H. (2013). Cross-platform model-driven development of mobile applications with md2. Proceedings of the 28th Annual ACM Symposium on Applied Computing - SAC 13. doi:10.1145/2480362.2480464
S9	Mercado, I. T., Munaiah, N., & Meneely, A. (2016). The impact of cross-platform development approaches for mobile applications from the users perspective. Proceedings of the International Workshop on App Market Analytics - WAMA 2016. doi:10.1145/2993259.2993268
S10	Delia, Lisandro, et al. "Multi-Platform Mobile Application Development Analysis." 2015, doi:978-1-4673-6630-4/15.
S11	Dalmasso, I., Datta, S. K., Bonnet, C., & Nikaein, N. (2013). Survey, comparison and evaluation of cross platform mobile application development tools. 2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC). doi:10.1109/iwcmc.2013.6583580

S12	Ali, M., & Mesbah, A. (2016). Mining and characterizing hybrid apps. Proceedings of the International Workshop on App Market Analytics - WAMA 2016. doi:10.1145/2993259.2993263
S13	Dunka, Bakwa & Emmanuel, Edim & Oyerinde, Dantala. (2017). HYBRID MOBILE APPLICATION BASED ON IONIC FRAMEWORK TECHNOLOGIES. International Journal of Recent Advances in Multidisciplinary Research. 04. 3121-3130.
S14	Vilcek, T., & Jakopec, T. (2017). Comparative analysis of tools for development of native and hybrid mobile applications. 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). doi:10.23919/mipro.2017.7973662
S15	Majchrzak, T., & Grønli, T. (2017). Comprehensive Analysis of Innovative Cross-Platform App Development Frameworks. Proceedings of the 50th Hawaii International Conference on System Sciences (2017). doi:10.24251/hicss.2017.745
S16	Angulo, E., & Ferre, X. (2014). A Case Study on Cross-Platform Development Frameworks for Mobile Applications and UX. Proceedings of the XV International Conference on Human Computer Interaction - Interacción 14. doi:10.1145/2662253.2662280
S17	Ali, M., Joorabchi, M. E., & Mesbah, A. (2017). Same App, Different App Stores: A Comparative Study. 2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft). doi:10.1109/mobilesoft.2017.3
S18	VITOLS, Gatis, et al. "Issues of Hybrid Mobile Application Development with PhoneGap: a Case Study of Insurance Mobile Application." International Baltic Conference, 2014.
S19	El-Kassas, W. S., Abdullah, B. A., Yousef, A. H., & Wahba, A. (2014). ICPMD: Integrated cross-platform mobile development solution. 2014 9th International Conference on Computer Engineering & Systems (ICCES). doi:10.1109/icces.2014.7030977
S20	Ahti, V., Hyrynsalmi, S., & Nevalainen, O. (2016). An Evaluation Framework for Cross-Platform Mobile App Development Tools. Proceedings of the 17th International Conference on Computer Systems and Technologies 2016 - CompSysTech 16. doi:10.1145/2983468.2983484
S21	Malavolta, I., Ruberto, S., Soru, T., & Terragni, V. (2015). End Users Perception of Hybrid Mobile Apps in the Google Play Store. 2015 IEEE International Conference on Mobile Services. doi:10.1109/mobserv.2015.14
S22	Brucker, A. D., & Herzberg, M. (2016). On the Static Analysis of Hybrid Mobile Apps. Lecture Notes in Computer Science Engineering Secure Software and Systems, 72-88. doi:10.1007/978-3-319-30806-7_5
S23	El-Kassas, W. S., Abdullah, B. A., Yousef, A. H., & Wahba, A. M. (2015). Taxonomy of Cross-Platform Mobile Applications Development Approaches. Ain Shams Engineering Journal, 8(2), 163-190. doi:10.1016/j.asej.2015.08.004
S24	El-Kassas, W. S., Abdullah, B. A., Yousef, A. H., & Wahba, A. M. (2016). Enhanced Code Conversion Approach for the Integrated Cross-Platform Mobile Development (ICPMD). IEEE Transactions on Software Engineering, 42(11), 1036-1053. doi:10.1109/tse.2016.2543223

S25	Ciman, M., & Gaggi, O. (2017). An empirical analysis of energy consumption of cross-platform frameworks for mobile development. Pervasive and Mobile Computing, 39, 214-230. doi:10.1016/j.pmcj.2016.10.004
S26	Biørn-Hansen, Andreas & Grønli, Tor-Morten & Ghinea, Gheorghita. (2017). Baseline Requirements for Comparative Research on Cross-Platform Mobile Development.
S27	Byalik, A., Chadha, S., & Tilevich, E. (2015). Native-2-native: Automated cross-platform code synthesis from web-based programming resources. ACM SIGPLAN Notices, 51(3), 99-108. doi:10.1145/2936314.2814210
S28	Chadha, S., Byalik, A., Tilevich, E., & Rozovskaya, A. (2017). Facilitating the development of cross-platform software via automated code synthesis from web-based programming resources. Computer Languages, Systems & Structures, 48, 3-19. doi:10.1016/j.cl.2016.08.005
S29	Tung, Y., Bui, D., & Shin, K. G. (2018). Cross-Platform Support for Rapid Development of Mobile Acoustic Sensing Applications. Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services - MobiSys 18. doi:10.1145/3210240.3210312
S30	Boushehrinejadmoradi, N., Ganapathy, V., Nagarakatte, S., & Iftode, L. (2015). Testing Cross-Platform Mobile App Development Frameworks (T). 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE). doi:10.1109/ase.2015.21