Date of publication MAR-29, 2021, date of current version AUG-13, 2021.
www.computingonline.net / computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ijc.20.1.2091

Cross-Platform Mobile App Development in Industry: A Multiple Case-Study

TASNIM ZOHUD, SAMER ZEIN
Birzeit University, Bir Zeit, Ramallah and Al-Bireh, 71939, Palestine
(e-mail: t.zuhod@gmail.com, szain@birzeit.edu)

Corresponding author: Tasnim Zohud (e-mail: t.zuhod@gmail.com).

ABSTRACT Cross-platform frameworks for mobile application development are pervasive and allow developers to build applications that can operate on a spectrum of mobile platforms. With the high availability of such frameworks, it is imperative to give better understanding about the perception of industry and the challenges they face in this field. Although there are several studies in this field of research, only few research works investigate cross-platform mobile development in the industrial contexts in-depth. We conducted a qualitative research using multiple case-study approach. Interviews and focus groups methods were applied to gather information from four different software development companies in Palestine. This study aims at understanding industrial teams approach to mobile cross-platform development. In particular, it seeks to identify the challenges they face in the areas of code maintenance and software testing. Our findings cover the important aspects in industrial contexts, including the development process, as well as, how and why the cross-platform development approach is chosen. From the practitioners' perspectives, we argue that the developers' experience is the most influential factor in the development process. Further, our study investigates the different cross-platform technologies in the industry, and the dominant one from the practitioners' perspectives, where they agreed that the React-Native is the promising and dominant technology. From the challenges perspective, we found that the APIs provided by the vendors are not easy to operate by developers and considered to be a major challenge. Finally, this research investigates how testing and maintenance methodologies are applied along with their challenges.

KEYWORDS cross-platform development, mobile development, mobile apps, cross-platform in industry.

I. INTRODUCTION

RECENTLY, smartphones have become extensively utilized in different aspects of our modern life. In fact, mobile applications, apps hereinafter, are being included into critical sectors including m-banking, m-payments, m-health, and even military contexts to mention just a few [1]. Moreover, mobile app development is considered as one of the most growing IT sectors. The fierce competition in the mobile app market makes it imperative to reduce the development time of these apps and at the same time allowing them to run on large spectrum of hardware devices and platforms. Mobile app development is so different from traditional web and desktop applications and indeed has its own peculiarities such as the need to deal with different platforms. Nowadays, we have various platforms that power mobile devices such as Apple iOS, and Google Android.

Mobile apps can be developed using either native, m-site or cross-platform technologies [2]. Native approach uses programming language that is specific for certain mobile platform such as Java and Kolten for Android; and Swift for Apple devices. This means that the developed app using this approach can be run only on its particular platform. In order for a mobile app to be successful and pervasive, it is expected to run on each mobile platform. However, this can be problematic using the native approach since that each platform will require a special mobile app.

This paper has been submitted for the Open Special Issue on Green Mobile Computing and IoT Systems. Assessment, Modeling, Assurance.

The second approach is known as the m-site approach, depends on web technologies (HTML, CSS and JavaScript) and is designed to run within the boundaries browser application. Mobile web apps (m-sites) do not need to be adapted to any operating platform since they are implemented to run as web applications with the ability to be viewed on small screens [3]. Despite that, m-site apps have poor performance; the response time is highly affected by the client-server interaction, and the execution of code through the browser sandbox limits the access to all device hardware capabilities. In addition, they do not provide the native look and feel of native apps, thus making them less attractive for the end user.

The third approach is the cross-platform, which includes several techniques to develop mobile apps such as hybrid, cross-compiled, interpreted and other solutions [4]. Hybrid applications are developed based on Web technologies (HTML, JavaScript and CSS) but unlike the m-sites, they are not executed within the browser. Instead, they are run in a special web container with better access to device capabilities through a specifically designed APIs. Nevertheless, the use of non-native components in the interface harms the user experience, and the execution is slowed down by the load associated with the web container [5]. The other types of cross-platform solutions constitute approaches that translate the original code of the app into mobile platform native code [6]. All of the cross-platform solutions aim to achieve the same objective: building one app that operates over different mobile platforms. Recently, cross-platform technologies have become pervasive and highly adopted by the industry. With the availability of these different options to develop mobile apps, we found it imperative to investigate the industrial perspective and the challenges they face.

Our motivation in this research is highly stipulated by the findings revealed from our previous systematic mapping study, Zohud et al. [7], which concluded that there is a lack in research that investigates testing and code maintenance methods for cross-platform mobile apps.

Using interviews and the focus groups, we gathered qualitative data on approach adaption process, framework dominance, common challenges, testability and maintainability of cross-platform app.

The results show that the dominant cross-platform tool in the industry is React-Native, due to continuous support and improvement by the Facebook team. Among the challenges faced by the developers, the hardware API calls, libraries provided by the community and the communities for the cross-platform technologies are the mostly reported challenges. Regarding the testing techniques, the manual testing is the mostly applied technique.

The remainder of this paper is organized as follows: The related work is introduced in Section II. Section III demonstrated the research propositions that this study is based on. In Section IV we described the research method of this empirical study. The results are presented in Section V followed by Section VI where we discussed our findings. Section VII presents threats to validity, VIII presents future work and our work conclusions are given in Section IX.

II. RELATED WORK

Several studies were conducted to investigate the cross-platform mobile apps development. These studies are different in the focus topic and applied research methods. The majority of the previous studies are comparative studies, as they were conducted to compare between different approaches, frameworks, tools and applications [7].

The study conducted by Ciman et al. [9] is an example of comparative study with the experiment carried out to compare different cross-platform frameworks based on the energy consumption. Another comparison analysis was conducted by Willocx et al. [10] to evaluate the native and hybrid approaches based on application that had been developed with two versions each using one of the approaches. An experimental study was implemented by Vilcek et al. [11] to compare between different cross-platform development tools, which are the Android Studio, Xcode, Visual Studio, Ionic, PhoneGap and NativeScript. The study conducted by Ali et al. [12] is an additional example of the comparative studies, they mined 80K of hybrid apps from different stores to evaluate these apps and rate them according to the platform they run on.

Part of the conducted studies was applied to analyze specific cross-platform development framework according to different factors, example of such studies is the research done by Dunka et al. [13] that focused on building hybrid mobile apps with the use of Ionic framework.

The other type of the applied studies those that were implemented to propose new approaches in the field of cross-platform, some of them suggested development frameworks solutions, and others introduced evaluation frameworks and few proposed testing frameworks. The conducted study by Tung et al. [14] is an example of the research that proposed a development framework. The proposed library has a purpose of facilitating the development of cross-platform apps with acoustic sensing.

There are few studies investigated software development in the industrial contexts, a study conducted by Zein et al. [15] aimed to investigate the testing techniques that mobile developers use, and to identify the challenges they face by applying a multiple case-study with four companies in the industry of mobile development. They reported the major challenges in the industry regarding testing mobile applications and found that there is no sufficient knowledge about testing mobile apps in a way that confirms life-cycle properties. Additionally, they found that there is no clear testing approach applied in the testing phase. A study of Zarour et al. [16] investigated the software development effort estimation in the Palestinian IT sector. They designed a survey and then conducted a multiple-case study with four different companies; their findings showed that inaccurate estimation of the project causes 25% of the cost overrun. There is another research of Asfour et al. [17] that applied a qualitative method in the Palestinian industry. Their investigation focused on exploring the Agile process in developing mobile apps by conducting multiple-case study involving four mobile development companies. They concluded that not all principles of Agile are fit the mobile development process. Another study with applied qualitative methods was conducted by Ahmad et al. [18] and investigated the challenges in mobile development through online questionnaire and interviews. They reported several challenges faced by the industry such as fragmentation, user experience, testing, reuse of code and others. In a study of Biørn-Hansen et al. [8] it was compared between hybrid and interpreted approaches in terms of ease of communication between app-side and native-side, in addition to the use of native functionality such as device file storage. They conducted an online questionnaire targeted at the industry practitioners with 101 respondents. They found that the major difference between those two approaches is that hybrid approach depends on WebView component which constitutes an abstraction layer for native-side calls that passed through. While the interpreted approach bridges the native features instead of using on-device interpreters. However, there is no clear issue regarding the communication between app-side and native-side in both approaches, but hybrid one requires less coding effort and shows that it is four times faster than the interpreted approach in terms of communication between app-side and native-side.

A recent survey-based study conducted by Biørn-Hansen et al. [29] is the closest study to our work, but it is a quantitative research focused only on the cross-platform development tools in terms of adaption, dominance and challenges that developers face, while we investigated the whole development process for the cross-platform app. They formed an online survey of five questions put to 101 participants, then they analyzed the results quantitatively and highlighted the commonly reported issues in the industry which are the performance, user experience and the efficiency of the development tools. However, in our study we apply a multiple case-study with interviewing developers from four development companies, then we analyzed the result using qualitative method and concluded with answers for the research questions we designed.

III. RESEARCH PROPOSITIONS

Providing propositions helps in addressing the research problem that will be studied, and the data types that will be collected become clearer. Further, it is helpful to get more focused on the data that will be collected, and increases the ability for the researcher to stay at the research scope limit [20].

Based on the above-presented studies that are already investigated in the literature, we prepared three propositions that will be studied in our research. (1) Cross-platform development is a challenging approach which is not explored deeply in the industrial contexts, (2) maintenance in cross-platform apps might not be effective, (3) automation testing in cross-platform approach might not be supported and effective as in other development approaches.

The objective of this study is to explore and investigate these propositions.

IV. RESEARCH METHOD

This research applied a qualitative approach which investigates the issues of cross-platform apps development. This type of research is known to be exploratory in nature; provides in-depth information for specific case; and gives a chance to enrich the understanding of some problem [23]. Qualitative research is intended to cover the behavior and the 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.

Fig. 1 shows the multiple-case study design approach, which is inspired by Yin [21]. We started with defining the research theory, then design the data collection procedure that will be performed on the selected cases.

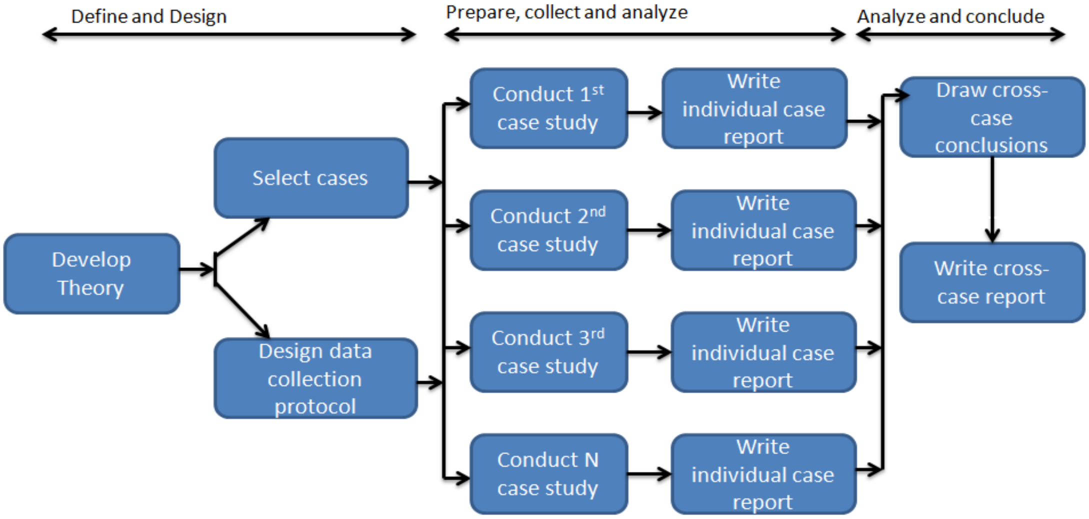
Define and Design
Prepare, collect and analyze
Analyze and conclude
Develop Theory
Select cases
Design data collection protocol
Conduct 1st case study
Conduct 2nd case study
Conduct 3rd case study
Conduct N case study
Write individual case report
Write individual case report
Write individual case report
Write individual case report
Draw cross-case conclusions
Write cross-case report

Figure 1. Multiple-case study design [21]

The theory was developed on the basis of the literature review, mainly by our previous systematic mapping study of the work of Zohud et al. [7]. The main points on which the theory was established are the following: there is a lack in code maintenance investigation for cross-platform apps, in addition to the finding that there are not enough studies conducted on the testing techniques in this area.

The qualitative data is collected using multiple case-study method, which includes two data collection methods: (i) interviews, and (ii) focus groups. The gathered data was in a form of written notes and recorded audios.

A. DATA ANALYSIS

The variety of data collection methods help to apply data triangulation which is according to [21] facilitates the data validation using different data collection techniques over multiple data resources. The collected data will be analyzed in a way that facilitates retrieving the relevant passages. We used a thematic coding technique [21] to classify the passages according to predefined codes and generate the themes.

THEMATIC CODING ANALYSIS

It is a form of a qualitative research analysis that inspects the themes of the given data which describes the research phenomenon. The coding analysis must 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 to narrow the wide data through discovering themes, and then specifying the research questions [25]. Fig. 2 demonstrates the following thematic coding analysis process which is applied in this research:

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

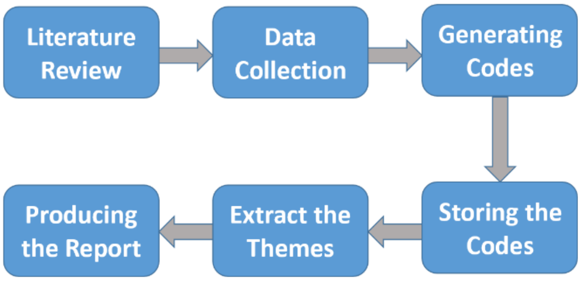

Figure 2. Thematic coding analysis steps

Table 1 shows an example of the thematic coding used in this research. Each row represents a sentence for different company of the mentioned RefID.

Table 1. Thematic coding example

RefID	SentenceId	Sentence	Code1	Code2
AT1	21	React-Native is the dominant technology	Development process	Development tool
AT3	53	Automation testing may waste our time	Testing	Challenge
AT2	35	Sometimes we need to modify the cross-platform library (build a plugin) to support some features	Challenge	
Ex1	2	The cost is an important factor for the client to choose the development approach	Development process	
Ex2	22	The plugins restrict the developers while using them and the data they provide	Challenge	
Ex3	43	We usually choose either Ionic or React-Native	Development process	Development tool
CT	4	Web developers are highly available since web technologies are old development techniques, so it is easier and takes less time/cost to go for hybrid (Ionic)	Development process	Development tool
CT	14	We do manual testing due to automation testing cost	Testing	Challenge
CT	15	Ionic has a service facilitating the maintenance process (Ionic Pro) which makes the developers able to push the change without re-publishing the app again	Maintenance	
IS	10	We use manual testing on our projects	Testing	
IS	14	Maintenance on cross-platform apps reduces the change time	Maintenance	
IS	6	We previously in one of the apps switched from Ionic2 into native because Ionic highly utilizes the device resources which slows down the app response	Challenge	

B. RESEARCH QUESTIONS

Selecting the development approach is a critical decision that industrial teams should take. Cross-platform mobile app development approach has many benefits that can make the developers willing to provide 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. As mentioned in the Introduction section (1), there are only few studies that explore the cross-platform development in the real-life.

Based on the research problem identified above, we formulated three main research questions. The first question aims to explore the process that industrial companies use to develop the cross-platform apps, and to spot the light on the most reported issues in developing these apps.

RQ1: How do industrial teams develop cross-platform mobile apps in term of development process and tools used? And what are the challenges they face?

The second question targets the maintainability aspect of the cross-platform app from the practitioners' point of view.

RQ2: To what extent do cross-platform frameworks support code maintenance?

The third question aims to investigate the testing area for the cross-platform apps in the practice. It also aims to examine the efficiency of testing techniques in detecting bugs of the cross-platform apps.

RQ3: What are the testing techniques supported by cross-platform frameworks? And how effective are they?

V. RESULTS

The results section introduced the findings of the case studies. This empirical study investigated the data on four different companies that are referred to as C1, C2, C3 and C4 in order to maintain confidentiality. The first case C1 was 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 delivers 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 develops 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 work on cross-platform mobile development, they specialize in developing hybrid apps using Ionic framework. Table 2 summarized the teams' sizes.

Table 2. Teams' sizes

Case study ID	Team size
C1	3
C2	4
C3	3
C4	2

This section provides the results received 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 to choose the appropriate development approach.

We noticed 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 which affects the decision is the team members' experience. When they are experienced in web technologies then it is easier for them to use cross-platform technologies. This decision also depends on additional important factors such as the client budget, since the native development process is more costly due to the number of resources it needs and the time it takes, so when the client budget is limited then it is more suitable to develop the application using cross-platform technologies. The team leader in C4 explained this:

"*We choose the development approach based on our experience, but the client budget can restrict us, since low budget suits cross-platform*" – Team leader, C4.

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

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

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 then the developers use the native technologies to build it. This is highlighted by a developer in C4:

"*Cross-platform doesn't suit apps with very high performance such as games*" – Developer, C4.

After choosing the development approach and when the cross-platform approach is taken for developing the app, the developers have to take the decision about which cross-platform technology to use.

Development teams often 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.

We also 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 working on Ionic confirmed that they started learning React-Native in order to support it in the future.

We noticed that there are common challenges that most of the development teams faced with 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 numbers 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 not comply with the store policies.

We noticed 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 fixed for these libraries are slow. The team leader in C4 stated that "*The Ionic community did not 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 codes.

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

We noticed that the dominant supported testing technique for cross-platform apps is the manual testing. Most of the studied companies do not 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 supports the automation testing explained that they have started supporting the UI automation and unit testing recently and they use Appium tool for UI tests and Jest for unit testing. They face some issues with Appium when handling the UI components, it behaves in different way when accesses 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 is in the bottom of the page without scrolling, while it cannot do this on iOS without scrolling the page.

VI. DISCUSSION

This section interprets the case studies results in order to provide a better understanding of the research and to answer the research questions. This is an exploratory study, which explores the areas that are not well investigated in the literature, and aims to motivate for future studies in this field.

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 moves on 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 Biørn-Hansen et al. [29] and based on survey results of five questions, which conducted with 101 participants. Biørn-Hansen et al. study focused on the cross-platform technology adaption and the challenges the developers faced with.

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 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.

A. 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 preferable in this decision, since these tools and frameworks depend on web technologies which are more outdated technologies than the native mobile development technologies, so web developers are highly requested.

Although the practitioners mentioned all the factors that affect the decision which are confirmed by the researchers in the studies such as [26], but the conducted research did not highlight 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 are the primary selection reason.

However, the factor that may force the developers to go for native is that the application has many features which require for device resources such as GPS, Bluetooth, and etc., which is confirmed by [26]. In this case, the developers have to write many native modules to handle these calls, this workaround takes time that native approach can help to provide 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.

B. CROSS-PLATFORM TOOLS

There are several cross-platform tools options to develop mobile apps, although the number of companies that use Ionic is maybe larger than the number of those who are using React-Native, which 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 its 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 is a new technology, but the recent research agrees with the practitioners' opinion as for React-Native popularity. An experimental study [19] 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 [27].

C. CHALLENGES

Several previous researches such as [27-29] introduced the challenges that developers face with in the cross-platform development field. These challenges include fragmentation, performance, user experience and hardware resources calls. Almost all the challenges that were revealed in 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 provided by 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.

D. MAINTENANCE

The maintainability of cross-platform app was studied in [30], 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 runs 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.

E. 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 by the majority of companies in the cross-platform development field, this is due to the fact that automation testing needs more resources in order to build a useful and reliable regression that can be performed after each app release. When the companies go forward cross-platform development in order to reduce the resources cost, 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 discusses the testing in the industrial contexts.

VII. THREATS TO VALIDITY

Threats to validity affect the research quality, and ensuring the case study reliability and validity offers high degree of confidence of the research findings [31]. We applied validity and reliability mechanisms in this study, we used three criteria to guarantee high level of case studies accuracy, and those criteria are construct validity, external validity, and reliability [15].

- Construct validity: having the correct measurement for the investigated concepts;
- External validity: identifying the study context for which the results can be generalized; and
- Reliability: shows that the data collection process can be replicated with the same results.

Table 3 shows the strategies inspired by Yin [21] that we applied to ensure the validity of the case study. The internal validity is excluded since this study is exploratory in nature.

Table 3. Strategies to maintain case study validity

Test	Tactic	Strategy
Construct validity	- Use multiple sources of evidence - Maintain chain of evidence	- Gathering data from interviews and focus groups - During thematic coding procedure, numbers were assigned to sentences and linked to their source documents.
External validity	- Use of theory	- Theory was established based on the literature review
Reliability	- Use case study design - Create case study database	- A case study is designed and used - Case transcripts, reflections, and spread sheets were created

Our study investigated four software development companies. We believe that investigating more companies can yield into more accurate results.

VIII. FUTURE WORK

As a future investigation, we suggest considering some additional aspects in the future research in order to achieve better results. There are plenty of improvements on the findings that can be done when considering the software features and team sizes in the future conducting studies.

IX. CONCLUSION

This research presented an industrial investigation of cross-platform mobile apps development. The main research proposition was that the cross-platform development is not explored deeply in the industrial contexts, since there is a lack of studies that investigate it. Our study is done based on the main proposition, which is implemented through conducting interviews with development teams from four different companies. The conducted research deeply explores how the development teams in the companies develop mobile apps using the frameworks of the cross-platform technologies. Also, the testability and maintainability aspects in the development process are investigated.

In this study, we investigated some research gaps identified in the literature and compared them with the practitioners' perceptions. We noticed that the main factor for adapting certain approach or framework is the developers experience in the team, also, the app requirements is another influential factor in this decision. We found that the most popular technology in the area of cross-platform from the industrial point of view is React-Native. Further, we identified different common reported challenges that developers usually face with such as the device resources access, the plugins provided by the community and the limited support from the communities of the cross-platform technologies. Moreover, testing process is often based on manual techniques in most of the companies.

DATA AVAILABILITY

The data used in this study to support the presented results have been pushed to GitHub repository. The following link is the data repository location: https://github.com/TasnimZuhod/industry_percpectives_interviews_and_analysis.

References

[1] M. Nagappan, E. Shihab, "Future trends in software engineering research for mobile apps," *Proceedings of the IEEE 23rd International Conference on Software Analysis, Evolution and Reengineering*, 2016, pp. 21–32. https://doi.org/10.1109/SANER.2016.88.

[2] R. Nunkesser, "Beyond Web/Native/Hybrid: A new taxonomy for mobile app development," *Proceedings of the IEEE IEEE/ACM 5th International Conference on Mobile Software Engineering and Systems (MOBILESoft)*, 2018, pp. 214-218. https://doi.org/10.1145/3197231.3197260.

[3] A. Charland, B. Leroux, "Mobile application development: Web vs. Native," *Queue*, vol. 9, issue 4, article no. 20, 2011. https://doi.org/10.1145/1966989.1968203.

[4] L. Delia, N. Galdamez, P. Thomas, L. Corbalan, P. Pesado, "Multi-platform mobile application development analysis," *Proceedings of the IEEE IEEE 9th International Conference on Research Challenges in Information Science (RCIS)*, 2015, pp. 181–186. https://doi.org/10.1109/RCIS.2015.7128878.

[5] G. Vitols, I. Smits, A. Zacepin, "Issues of hybrid mobile application development with PhoneGap: A case study of insurance mobile application," *Proceedings of the IEEE International Baltic Conference*, 2014, pp. 215–220.

[6] M. Latif, Y. Lakhrissi, E. Nfaoui, N. Es-Sabi, "Review of mobile cross platform and research orientations," *Proceedings of the IEEE International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, 2017, pp. 1-4. https://doi.org/10.1109/WITS.2017.7934674.

[7] T. Zohud, S. Zein, "A systematic mapping study of cross-platform mobile apps," *Journal of Computer Science*, vol. 15, issue 4, pp. 519-536, 2019. https://doi.org/10.3844/jcssp.2019.519.536.

[8] A. Biørn-Hansen, G. Ghinea, "Bridging the gap: Investigating device-feature exposure in cross-platform development," *Proceedings of the 51st Hawaii International Conference on System Sciences*, 2018, pp. 5717–5724. https://doi.org/10.24251/HICSS.2018.716.

[9] M. Ciman, O. Gaggi, "An empirical analysis of energy consumption of cross-platform frameworks for mobile development," *Pervasive and Mobile Computing*, vol. 39, pp. 214-230, 2016. https://doi.org/10.1016/j.pmcj.2016.10.004.

[10] M. Willocx, J. Vossaert, V. Naessens, "Comparing performance parameters of mobile app development strategies," *Proceedings of the International Workshop on Mobile Software Engineering and Systems – MOBILESoft'16*, 2016, pp. 38–47. https://doi.org/10.1145/2897073.2897092.

[11] T. Vilcek, T. Jakopec, "Comparative analysis of tools for development of native and hybrid mobile applications," *Proceedings of the IEEE 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2017, pp. 1516-1521. https://doi.org/10.23919/MIPRO.2017.7973662.

[12] M. Ali, M. E. Joorabchi, A. Mesbah, "Same app, different app stores: A comparative study," *Proceedings of the IEEE IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft)*, 2017, pp. 79–90. https://doi.org/10.1109/MOBILESoft.2017.3.

[13] B. Dunka, E. Emmanuel, D. Oyerinde, "Hybrid mobile application based on ionic framework technologies," *International Journal of Recent Advances in Multidisciplinary Research*, vol. 4, pp. 3121-3130, 2017.

[14] Y. Tung, D. Bui, K. G. Shin, "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*, 2018, pp. 455–467. https://doi.org/10.1145/3210240.3210312.

[15] S. Zein, N. Salleh, J. Grundy, "Mobile application testing in industrial contexts: An exploratory multiple case-study," *Proceedings of the IEEE International Conference on Intelligent Software Methodologies, Tools and Techniques (SOMET'2015)*, 2015, pp. 30–41. https://doi.org/10.1007/978-3-319-22689-7_3.

[16] A. Zarour, S. Zein, "Software development estimation techniques in industrial contexts: An exploratory multiple case-study," *International Journal of Technology in Education and Science (IJTES)*, vol. 3, issue 2, pp. 72-84, 2019.

[17] A. Asfour, S. Zain, N. Salleh, J. Grundy, "Exploring agile mobile app development in industrial contexts: A qualitative study," *International Journal of Technology in Education and Science (IJTES)*, vol. 3, issue 1, pp. 29-46, 2019.

[18] A. Ahmad, K. Li, C. Feng, S. Asim, A. Yousif, S. Ge, "An empirical study of investigating mobile applications development challenges," *IEEE Access*, vol. 6, pp. 17711-17728, 2018. https://doi.org/10.1109/ACCESS.2018.2818724.

[19] R. Jackson, D. Drummond, S. Camara, "What is qualitative research?," *Qualitative Research Reports in Communication*," vol. 8, issue 1, pp. 21-28, 2007. https://doi.org/10.1080/17459430701617879.

[20] P. Baxter, S. Jack, "Qualitative case study methodology: Study design and implementation for novice researchers," *The Qualitative Report*,

vol. 13, issue 4, pp. 544-559, 2008. [Online]. Available at: http://www.nova.edu/ssss/QR/QR13-4/baxter.pdf.

[21] R. K. Yin, *Case Study Research: Design and Methods*, Los Angeles: Sage, 2009, 219 p.

[22] J. Gustafsson, "Single case studies vs. multiple case studies: A comparative study," 2017, [Online]. Available at: http://www.diva-portal.org/smash/get/diva2:1064378/FULLTEXT01.pdf.

[23] C. Wagner, B. Kawulich, M. Garner, *Collecting Data through Observation, in Doing Social Research: A Global Context*, Berkshire: McGraw Hill, 2012, 20 p.

[24] L. Cohen, L. Manion, K. Morrison, *Research Methods in Education*, 5th Edition, London: Routledge Falmer, 2000, 462 p.

[25] D.L. Morgan, R.A. Kreuger, *The Focus Group Kit* (6 vols.). Thousand Oaks, Sage, 1998, 692 p.

[26] C. R. Raj, S. B. Tolety, "A study on approaches to build cross-platform mobile applications and criteria to select appropriate approach," *Proceedings of the IEEE Annual IEEE India*, 2012, pp. 625–629.

[27] C. Nakazawa, *React Native Open Source*, 2019, [Online]. Available at: https://facebook.github.io/react-native/blog/2019/03/01/react-native-open-source-update.

[28] I. Malavolta, S. Ruberto, T. Soru, V. Terragni, "End users perception of hybrid mobile apps in the Google play store," *Proceedings of the IEEE International Conference on Mobile Services*, 2015, pp. 25-32. https://doi.org/10.1109/MobServ.2015.14.

[29] A. Biørn-Hansen, T. Gronli, G. Ghinea, S. Alouneh, "An empirical study of cross-platform mobile development in industry," *Hindawi Wireless Communications and Mobile Computing*, vol. 2019, Article ID 5743892, 12 pages, 2019. DOI: https://doi.org/10.1155/2019/5743892.

[30] M. Martinez, S. Lecomte, "Towards the quality improvement of cross-platform mobile applications," *Proceedings of the IEEE IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft)*, 2017, pp. 184–188. https://doi.org/10.1109/MOBILESoft.2017.30.

[31] P. Runeson, M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, vol. 14, issue 2, pp. 131-164, 2009. https://doi.org/10.1007/s10664-008-9102-8.

__TASNIM ZOHUD,__ Master, Software Engineering, Birzeit University, Palestine. Sr. QA Automation Engineer. Area of scientific interests: software testing, software development, systematic literature reviews.

__SAMER ZEIN,__ received his Master's Degree in Software Engineering from Northumbria University, United Kingdom, and his Doctoral Degree in CS from IIUM, Malaysia. He is currently an Assistant Professor at Department of CS, Birzeit University. He has authored several research articles published in reputed peer-reviewed journals and conferences. His area of interests includes empirical software engineering, mobile app development and testing, model-based software engineering, and systematic literature reviews.