





## Research Article

# An Empirical Study of Cross-Platform Mobile Development in Industry

Andreas Biørn-Hansen <sup>1,2</sup>, Tor-Morten Grønli <sup>1</sup>,  
Gheorghita Ghinea <sup>1,2</sup> and Sahel Alounch <sup>3</sup>

<sup>1</sup>Mobile Technology Lab, Department of Technology, Kristiania University College, Oslo, Norway

<sup>2</sup>Department of Computer Science, Brunel University London, Uxbridge, UK

<sup>3</sup>Faculty of Electrical Engineering and Information Technology, German Jordanian University, Amman, Jordan

Correspondence should be addressed to Andreas Biørn-Hansen; [andreas.biorn-hansen@westerdals.no](mailto:andreas.biorn-hansen@westerdals.no)

Received 25 May 2018; Revised 29 October 2018; Accepted 22 November 2018; Published 3 January 2019

Academic Editor: Gerardo Canfora

Copyright © 2019 Andreas Biørn-Hansen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The purpose of this study is to report on the industry's perspectives and opinions on cross-platform mobile development, with an emphasis on the popularity, adoption, and arising issues related to the use of technical development frameworks and tools. We designed and conducted an online survey questionnaire, for which 101 participants were recruited from various developer-oriented online forums and websites. A total of five questions are reported in this study, of which two employed a Likert scale instrument, while three were based on multiple choice. In terms of technical frameworks, we find that PhoneGap, the Ionic Framework, and React Native were the most popular in use, both in hobby projects and in professional settings. The participants report an awareness of trade-offs when embracing cross-platform technologies and consider penalties in performance and user experience to be expected. This is also in line with what is reported in academic research. We find patterns in the reported perceived issues which match both older and newer research, thus rendering the findings a point of departure for further endeavours.

## 1. Introduction

Since the advent of the smartphone era, mobile-specific software applications, or *apps* for short, have seen a drastic increase in popularity [1], a surge correlating to the sheer amount of smartphone owners, reaching into the billions as of 2018 [2, 3]. Alongside the wide reach and general adoption of the smartphone, more than 5.6 million apps are available for download throughout the three major app marketplaces [4], specifically Apple's App Store, Google's Play Store, and Microsoft's Store. Additionally, across the app marketplaces, hundreds of billions of apps are downloaded on annual basis [1]. With revenues generated through apps estimated to stretch into the hundreds of billions USD in the run up to year 2020 [5], the development, availability, and use of apps are likely to continue and grow. In our study, we are specifically interested in the actual development of apps as conducted by practitioners in the industry and the variety of possible development approaches and technologies existing

on the market. Our motivation is grounded in suggestions from previous research (e.g., [6, 7]), and in an identified lack of quantitative perspectives from the industry in relation to mobile app development, with an emphasis on what we refer to as cross-platform development. Traditionally, mobile smartphone apps have been developed using tools designed for heterogeneous platform-specific development [8, 9]. This type of development is commonly referred to as the Native development approach [10], pointing to the use of development environments, Software Development Kits (SDKs), and programming languages native to the target mobile platform. With multiple major platforms on the market, namely, those of Apple's iOS, Google's Android, and Microsoft's Windows 10 Mobile, a consequence of the traditional development approach is the arduous efforts that must be put forth to develop an idea into software that can be deployed to all the aforementioned platforms [7], inherently requiring proficiency in numerous platform-specific tools and languages. This typically includes the Xcode environment

and Swift/Objective-C for iOS development and the Android Studio environment and Java/Kotlin for Android development [8, 11]. Nevertheless, of the more than 5.6 million apps available throughout the app stores, the absolute majority were developed using this exact development approach [12].

An alternative path for conducting app development is through the use of cross-platform mobile development frameworks, allowing for the reuse of code across multiple platforms [12]. This is in stark contrast to the heterogeneous nature of the Native development approach, in which code-sharing has yet to be identified as practical, although novel approaches for translation of Native code have recently surfaced, including the Native-2-Native tool presented by Chadha et al. [13] and an approach for code conversion proposed by El-Kassas et al. [14]. Due to the historical impracticality of code-sharing between Native platforms, cross-platform mobile technologies have enjoyed the interest of both academia and industry practitioners [15, 16], as the benefits of adopting such technologies are frequently cited to decreased time-to-market and knowledge requirements when compared to Native app development [17]. When discussing cross-platform development, it is important to note that the term does not specifically target a single product, tool, framework, or development approach. Instead, *cross-platform* is an umbrella term covering all of the aforementioned concepts. A multitude of technical frameworks, tools, and overarching conceptual development approaches belong to this term, adding to its complexity. Technical frameworks including PhoneGap, Cordova, Ionic, Titanium Appcelerator, and MoSync are frequently mentioned in the literature [18, 19]. Each of these frameworks can be categorized into an overarching development *approach* based on parameters including user interface rendering environment, access to device functionality, and code compilation technique [10]. These approaches are discussed in additional detail throughout the remainder of this paper.

While cross-platform development does involve a series of benefits, certain drawbacks are also frequently mentioned in the literature. In 2012, Facebook found that their cross-platform based mobile app did not manage to meet certain level of their requirements, due to the app's inherent and unique complexity [20, 21]. This, together with reported issues on scalability and unmet user experience expectations, triggered Facebook to abandon their cross-platform app and instead embrace traditional platform-specific Native app development [20, 21]. Their previous cross-platform based application was developed using a development approach referred to as the *Hybrid* approach, an alternative way of developing mobile apps leveraging common web technologies including HTML, CSS, and JavaScript [7]. Facebook abandoning Hybrid development left the approach in a state of suspiciousness of whether it could in fact generate reliable apps [22, 23]. More than half a decade later, the reputation and application of the Hybrid approach are still much-debated topics throughout both academia and developer communities, as presented and discussed throughout this study.

To better understand the industry's perception of cross-platform development, we conducted an online survey

questionnaire to gain insight from mobile developers. Throughout the survey, we gathered quantitative data on framework popularity, framework usage, and common issues associated with cross-platform development. Using this data, further work can focus on researching and solving real-world problems and thus can be of great value and validity beyond academics. Future research can also base technical decisions on the findings we present throughout the article. Summarizing, with this paper we aim to bring industry perspectives to the research community, both through the presentation of the state of cross-platform development as reported by practitioners, as well as to bring forth empirically backed suggestions for further research.

*Outline.* The remainder of this paper is organized as follows. In Section 2 we discuss existing research previously carried out targeting cross-platform app development, with an emphasis on user-oriented studies and the overarching topics that were part of the questionnaire. A presentation of the research method and survey design is provided through Section 3. Following that, in Section 4 we bring forth our findings from the survey then discuss them throughout Section 5 in the context of previous work. A conclusion together with a discussion on limitations and suggestions for further work is then provided in Section 6.

## 2. Previous Work

Studies on cross-platform mobile development vary in methods and focus, with a wide range of topics being actively researched. Examples of such include performance scrutiny of cross-platform frameworks and tools (e.g., [19, 24–26]), studies on user experience (e.g., [6, 27]), more descriptive and overarching studies cover topics ranging from taxonomy (e.g., [10, 28]) and evaluations of development approaches (e.g., [12, 29, 30]) to more technical comparisons (e.g., [18, 31]) and app store analysis of cross-platform apps (e.g., [22, 32, 33]). The aforementioned categories have seen arduous work and results adding to an increasing body of knowledge. Some of the earliest works produced by researchers, including Heitkötter et al.'s [7, 12] comparative studies, focus on descriptively evaluating cross-platform frameworks on a wide array of criteria. With these studies, the authors provided a foundation for much of the research that has followed in recent years. The study at hand draws from the findings presented in these articles, which assisted in the development of the questions included in our survey questionnaire.

From the academic literature, we also encounter numerous notable and relatable studies drawing from and extending the seminal work presented by Heitkötter et al., including those by Mercado et al. [22], Ali and Mesbah [34], and Malavolta et al. [32, 33]. The key differentiators between these studies and our own current paper are that the aforementioned studies are (1) all of a technical nature although maintaining a link to our own research interest, and (2) they work on uncovering perceptions and opinions on apps developed using cross-platform technologies from the perspective of *end-users* through app store analyses, while we focus on the gathering and analyses of the perspectives of *developers*

through an online questionnaire. As we will further discuss throughout this article, and which is especially prominent in the results presented in Table 2, the perspectives of the end-user is in fact a factor of great importance and relevance to the participants of our survey.

Few studies on cross-platform mobile development are based on data gathered from survey questionnaires and interviews. This gap in the knowledge body is what we target with our study, with specific emphasis on industry perspectives. A study by Angulo and Ferre [27] is one of few identified research projects that lean more towards a user experience and questionnaire type study, rather than being predominantly software engineering or vehicles for discussion of more overarching topics. Yet, the study differs greatly from ours. Their focus is on testing apps developed using different frameworks on the same group of expert mobile device users to see if cross-platform development produced apps differ in respect of user experience. While the contributions from the study are highly interesting for the field of research and practice, the study targets end-users rather than developers.

A recent study by Francese et al. [35] has a focus similar to ours in that they, through an online questionnaire, investigate practitioners' relation to mobile app development and thoughts on technologies. While the authors do touch upon topics related to cross-platform app development, including querying the participants for which cross-platform frameworks they use, their predefined options differ from ours. The authors present findings of great interest and contributions to the body of knowledge. They provide valuable insight for further research, especially regarding development approach preference, and the work experience among the participants in terms of Hybrid app development. Our study differs greatly from [35]; our focus is on the participants' perception of issues related to cross-platform development, additional to broadening the question regarding cross-platform development frameworks. Whereas Francese et al. limit the predefined options to PhoneGap, Titanium, Appcelerator, and AppMobi, we extend the list of frameworks to further uncover specific patterns and technologies for discussion and further research.

We also identified a study by Puvvala et al. [36], focusing on interviewing and surveying mobile developers. With this study, the authors are interested in the creation of a model for choice of mobile platforms, backed by empiricism of the questionnaires conducted. A set of their questions share similarities with those included in our survey questionnaire, although the overall theme of their study is more towards parameters such as infrastructure costs, development environments, monetizing of apps, device fragmentation, and the availability of SDKs and code examples. Several of these parameters are also included in the previously presented Heitkötter et al.'s [7, 12] seminal work. The main similarity is a question on community support, which in the context of Puvvala et al.'s study refers to the communities surrounding iOS and Android, whereas in our study, the question on community is related to those surrounding specific technical development frameworks. We extend their work by adjusting the focal point from general mobile development to cross-platform technologies and include questions regarding

thoughts on potential constraints and issues arising when adopting such development frameworks.

Perhaps the study identified to be closest in nature to our current article is a recent interview-based empirical study by Ahmad et al. [37]. Rather than adopting an online questionnaire-based quantitative method as we do in the paper at hand, Ahmad et al. interviewed 34 practitioners on questions related to tools, user experience, reusability of code, testing, knowledge, change management, security, and fragmentation. While several of these questions partially match with those found in our questionnaire, we emphasize the technological tools and frameworks adopted by practitioners and provide quantitative data on issues related to cross-platform development (e.g., performance and developer communities, neither of which are included in Ahmad et al.'s study), in which the focus is less on the technological tools available, and more on the qualitative analyses and providing a set of challenges that build on and extend the Software Engineering Body of Knowledge. We find that their qualitative study and our quantitative study together form a good foundation for further research, both through illuminating technological options for ensuring external validity and through listing of challenges and issues reported by industry practitioners.

From our assessment of the academic literature, we notice a pattern in encountered themes and topics. The majority of identified research focuses on principles in software engineering and design science. The nature of these studies are predominantly technical and mathematical, in which evaluations of performance, user reviews, and comparisons of cross-platform frameworks are common topics. Studies similar in nature to ours target either general mobile developers or aim to gauge the opinions and perceptions of end-users. The purpose of our study, on the other hand, is to contribute with quantitative industry perspectives towards a better understanding of the issues underlying cross-platform mobile development and thus have an empirical foundation for suggesting further work in our field of research.

### 3. Research Method and Survey Design

The online survey questionnaire was launched on February 17th 2016, aiming to gather data from industry practitioners on development framework popularity and usage and on issues related to cross-platform development and the resulting apps. In this section, we present how the data gathering was conducted and from where participants were recruited, a brief introduction to the tools used, and the questions and survey instruments included, and we outline the results of a pilot test conducted prior to the launch of the survey. The raw survey data in Excel format is available online on Github, see Section "Data Availability" on page 23.

*3.1. Research Questions.* Having presented and analyzed relevant literature in the previous section, what we find is a knowledge gap mostly related to the industry adoption of cross-platform frameworks and practitioners' perceptions of issues arising in the context of cross-platform app

development. We developed two research questions targeting these overarching gaps.

The first research question targets the often-so reported *issues* related to cross-platform development (as discussed in numerous studies, e.g., [24–26, 30, 36, 38]). By asking participants to answer questions related to this research question, we can identify possible similarities between the perceptions of practitioners and scholarly researchers. Possible matches in identified issues could evidently indicate the need for additional research, hence possibilities for future work. The first research question reads as follows:

**RQ1:** Which issues regarding cross-platform mobile app development are commonly reported by the survey participants?

Our second research question target the adoption and use of technical cross-platform frameworks. With this question, our aim is twofold. Firstly, to identify discrepancies in the usage of these frameworks in the industry, versus their inclusion in academic research [16, 18]. Major discrepancies could indicate that a refresh of the technologies included and scrutinized in research is needed to stay industry relevant. This could be a practical implication of our study. Heitkötter et al. [12, p. 309] explicitly states that “keeping track with progress in [...] frameworks” is an important endeavour for future research. Secondly, to get an overview of the usage of technical frameworks in the industry. We differentiate between hobby usage and professional usage in an attempt to analyze whether the (imminent) future adoption of cross-platform frameworks in the industry might differ from the status quo. The second research question reads as follows:

**RQ2:** Which cross-platform frameworks are adopted by the survey participants, and to what degree does professional technology adoption differ from hobby usage?

**3.2. Survey Tool.** The questionnaire was developed using SurveyPlanet (SurveyPlanet: <https://www.surveyplanet.com/>), a survey tool for designing and carrying out online questionnaires. Two survey instruments were used, being those of *scoring* and *multiple choice*. The scoring instrument enabled the creation and use of a numbered Likert scale. While a Likert scale may be used with text as the option headings and numbers for the Likert items, SurveyPlanet chose to use numbers as the option headings and radio buttons for the Likert items. To support the numbered headings, an optional question *subtitle* field was available to specify the meaning behind the headings.

**3.3. Questions and Measures.** A total of five questions were included in the survey questionnaire. Two made use of the *scoring* instrument, and the remaining four used the *multiple-choice* instrument. To sufficiently adhere to standard practices for decisions on such as question wording, scale length and labelling, a document by Taylor-Powell [39] on scaling was identified and used as the foundation for the survey creation. The cross-platform development frameworks, as listed below in question #1, were chosen for inclusion in the survey due to prevalence in either or both industry and academia [16].

An “Other” option was included in the questionnaire, enabling participants to manually input an answer. We

deemed this a necessity due to the vast landscape of mobile development, thus rendering it highly possible that the predefined options were not exhaustive enough. The following questions and predefined options were included in the questionnaire:

Question #1: How familiar are you with the following frameworks?

Associated research question: RQ2

Instrument: *Likert (1-5)*

Description of instrument scale: (1) *Not at all familiar.* (2) *Slightly familiar.* (3) *Moderately familiar.* (4) *Very familiar.* (5) *Extremely familiar.*

Predefined options: (i) *Ionic Framework*, (ii) *jQuery Mobile*, (iii) *NativeScript by Telerik*, (iv) *PhoneGap*, (v) *React Native by Facebook*, (vi) *Reapp*, (vii) *Sencha Touch*, (viii) *Tabris.js*, (ix) *Titanium by Appcelerator*, (x) *Touchstone.js*, (xi) *Xamarin Forms*, (xii) *None*, (xiii) *Other*.

Question #2: How interested are you in exploring the following frameworks?

Associated research question: RQ2

Instrument: *Likert (1-5)*

Description of instrument scale: (1) *Not at all interested.* (2) *Slightly interested.* (3) *Moderately interested.* (4) *Very interested.* (5) *Extremely interested.*

Predefined options: *Same as in question # 1*

Question #3: Which of the following framework(s) are you currently using (hobby)?

Associated research question: RQ2

Instrument: *Multiple Choice*

Predefined options: *Same as in question # 1*

Question #4: Which of the following framework(s) are you currently using (professionally)?

Associated research question: RQ2

Instrument: *Multiple Choice*

Predefined options: *Same as in question # 1*

Question #5: If any, which of the following issues do you relate to cross-platform development?

Associated research question: RQ1

Instrument: *Multiple Choice*

Background for predefined options: *the list of predefined options presented below is compiled from the relevant literature discussed in Sections 2 and 3.1.*

Predefined options: (i) *overall loss in performance compared to native apps*, (ii) *Suboptimal User Experience (UX)*, (iii) *immature frameworks (too cutting edge, too much risk, etc.)*, (iv) *suboptimal options for creating good user interfaces (UI)*, (v) *immature communities (too new/low activity, etc.)*, (vi) *hard to integrate with device APIs*, (vii) *hard to test/debug*, (viii) *security issues*, (ix) *other*.

TABLE 1: Survey questionnaire results on framework familiarity, interest, and usage.

Framework ( <i>alphabetic order</i> )	Familiarity (Q#1)		Interest (Q#2)		Usage (Q#3-4)	
	Totals	Averages ( <i>max 5</i> )	Totals	Averages ( <i>max 5</i> )	N ( <i>Hobby</i> )	N ( <i>Prof.</i> )
Cordova *	-	-	-	-	2	0
Fuse by Fusetools	124	1.23	162	1.6	3	1
Intel XDK	132	1.31	147	1.46	2	0
Ionic Framework	289	2.86	281	2.78	37	17
jQuery Mobile	280	2.77	164	1.62	12	9
Meteor *	-	-	-	-	13	12
NativeScript by Telerik	159	1.57	199	1.97	5	1
PhoneGap	338	3.35	306	3.03	47	32
React Native by Facebook	302	2.99	432	4.28	46	26
Reapp	135	1.34	174	1.72	4	2
Sencha Touch	174	1.72	138	1.37	1	1
Tabris.js	112	1.11	145	1.44	1	0
Titanium by Appcelerator	173	1.71	149	1.48	3	3
Touchstone.js	123	1.22	164	1.62	2	1
Xamarin Forms	166	1.64	166	1.64	6	2
<i>None</i>	-	-	-	-	14	30

**3.4. Pilot Test.** To ensure that the survey was of a certain standard, involving understandable questions and appropriate instruments and to minimize the possibility of bias, a pilot test was conducted prior to the public release. The pilot test involved four interaction designers with experience in user testing, web design, and mobile development from industry projects. From the pilot, it was established that answering the entire survey took approximately 2 minutes from start to finish. This was deemed an acceptable length by the test panel, who reassured that the length was crucial for a survey with no tangible end-user rewards. Due to SurveyPlanet’s Likert scale-like implementation of scores, the test panel found it necessary to expand on the original description of the numbered option headings. This was introduced in the final version of the questionnaire, as further discussed in Section 3.3.

**3.5. Recruiting Survey Participants.** To ensure the possibility of a high response rate, the questionnaire link was posted to multiple websites and online groups targeting mobile developers. Specifically, these websites included Reddit, Hacker News, Crater.io, LinkedIn groups, and Facebook groups. In an attempt to avoid bias in the responses, the survey was only posted to generic online groups and forums not directly involved or targeting specific cross-platform frameworks. However, due to the nature of an online questionnaire, it is not possible to avoid that the survey link is shared and spread by participants or stakeholders onto unwanted or biased websites or online forums. Nevertheless, we can assume that some snowball sampling occurred. To the best of our knowledge, SurveyPlanet did not expose the URL of the website the respondents arrived to the survey from, making it inherently more difficult to filter out possible biased answers.

## 4. Results

**4.1. Survey Response.** A description of the survey together with a link to the online questionnaire was posted to multiple websites, forums, and online groups of relevance to mobile app development. The groups were closed Facebook and LinkedIn groups requiring a membership and approval from administrators. The websites consisted of newsboards such as Reddit, Hacker News, and Crater. Within the first 48 hours of publicly launching the survey, over 90 responses were recorded by SurveyPlanet. Due to the nature of the aforementioned websites, where posts are often ordered by date of submission and thus naturally disappear after some time depending on the website’s activity, the amount of new respondents after the initial few days declined drastically. After 11 days, the survey had reached 100 responses. In total 101 responses had been recorded by SurveyPlanet after 14 days, at which point we locked the survey and extracted the data. We find that the sample size ( $N = 101$ ) is of comparable size to similar studies, including those by Francese et al. [35] at 82 respondents, Puvvala et al. [36] at 137 respondents, and Ahmad et al. [37] at 34 respondents.

**4.2. Results: Framework Familiarity, Interest, and Usage.** The findings related to cross-platform development frameworks have been condensed and are given in Table 1. The structure of the table is as follows. The first column lists the frameworks in question. The second and third columns under “Familiarity” are dedicated to question #1 regarding the participants’ familiarity with each framework. The fourth and fifth columns under “Interest” display the answers given to question #2 regarding interest in exploring the frameworks. Columns six through seven list the participants’ votes regarding which

TABLE 2: Results from question on common issues.

Choices ( <i>ordered by N</i> )	<i>N</i>
Overall loss in performance compared to native apps	63
Suboptimal User Experience (UX)	58
Immature frameworks (too cutting edge, too much risk, etc.)	55
Suboptimal options for creating good User Interfaces (UI)	50
Immature communities (too new/low activity, etc.)	35
Hard to integrate with device API	32
Hard to test/debug	31
Security issues	13
“Dealing with platform/OS/manufacturer quirks (especially Android)”*	1
“Least common denominator feature set: your app’s features are limited to the intersection of iPhone and Android’s features”*	1

frameworks they use, in which column six is dedicated to hobby usage while column seven to professional usage.

The two most common answers from the respondents filling in the manual input option “Other” are marked with an asterisk (\*) symbol and are made up of Cordova and Meteor. We identify both frameworks to have some popularity among the survey respondents, albeit the Meteor framework does leverage Cordova as its underlying cross-platform technology, similar to the Ionic Framework and other like-minded hybrid frameworks. Cordova is in fact a derived open-source version of the PhoneGap framework, which is included in our survey questionnaire as a predefined option.

**4.3. Results: Perceived Issues.** Table 2 shows the collected responses from the multiple-choice based question “if any, which of the following issues do you relate to cross-platform development?” The options listed are issues frequently encountered and mentioned in the academic body of knowledge and are topics of manifold research projects (e.g., [19, 25, 26] on performance, [6] on user experience, and [40] on programmatic access to device APIs). The discussion following in Section 5 focuses especially on the relation between the state of research, represented by studies such as those previously mentioned, versus the opinions and perceptions of industry practitioners in regard to the findings presented in Table 2. The outcome of this discussion is of particular interest for future research, which we revisit in Section 6.

In the collected survey responses, we find two answers from those respondents answering the manual input option “Other”. In Table 2, these are wrapped in quotation marks, marked with an asterisk (\*) symbol and inserted at the end of the table.

## 5. Discussion

The core focus of the discussion that follows is to provide a better understanding of the state of research in relation to what industry practitioners deem both interesting and problematic with cross-platform mobile development. We draw findings and ideas from the academic literature base

and discuss them in the context of the previously presented survey results. By following such an approach, we are able to put forth ideas for further work that are anchored in industry needs.

**5.1. Commonly Reported Issues.** Associated research question: *Which issues regarding cross-platform mobile app development are commonly reported by the survey participants?*

The survey participants perceive performance, user experience and user interface design, risks associated technology adoption, difficulty of implementation and debugging, and security to be issues with cross-platform development. The latter, security, does, and concerning so, receive far less votes than the remaining options. These findings share similarities with those presented by Francese et al. [35] in their questionnaire-based study, wherein 17% of the participants report no security testing, 51% report manual testing, and the remainder either do security testing automatically or semiautomatically. Questioning security in the context of cross-platform frameworks is a topic covered in numerous studies, especially targeting the Android WebView component which the entire Hybrid development approach relies on. Typically, security studies report great security risks and sensitive data penetration potential in such apps, e.g., a study by Luo et al. [41] stressing what they refer to as eventual large-scale attacks on WebView-based apps. Academic efforts to minimize security risks have also been identified, specifically through the implementation of access control measures [42] and leak-detection tools [43]. We find that security, outside of WebView vulnerabilities, is more discussed in studies targeting general mobile development, i.e., not constrained to the specificity of *cross-platform development* research. A review study by Faruki et al. [44] assesses numerous techniques for introducing malware and malicious behaviour in Android apps, several of which are highly relevant for both Native and cross-platform mobile developers, including but not limited to ensuring optimal permissions schemes, and an awareness of the possibility of repackage apps. The same techniques and challenges are also stressed by Sufatrio et al. [45] in a review study similar in nature to that of Faruki et al. The state of security in mobile apps, specifically health-related software, is categorized as *alarming* in a study by Papageorgiou et al. [46], emphasizing the highly insecure implementations of sensitive data transfers over HTTP, user profiling, and more. Security is also the topic of He et al.’s [47] study on limiting attack surfaces through providing guidelines to stakeholders, including both end-users and developers, whereby the former stakeholder is asked to carefully evaluate the permissions asked for by apps upon installation, and the latter, developers, should have a clear understanding of the impact and the attack surfaces introduced with certain app permissions. From the literature, we find that apps asking the end-user for sensitive permissions may not describe the purpose or reason for requesting access to the permission-enabled data [48], rendering it difficult to know whether or not this data benefits the app-owner or a third-party more than it benefits the user [49]. The usage of in-app advertisement libraries is part of a large-scale review study conducted by Martin et al. [50], and these libraries are found to request additional app

permissions, thus having the potential of acting intrusively and maliciously. Such libraries are commonly found in free apps, according to Ruiz et al. [51], and have the potential of negatively impacting app store ratings, as well as the trust of end-users and customers [49]. In Ahmad et al.'s study on challenges in mobile development [37], in which 34 practitioners were interviewed, security was also one of the least-reported challenges. As these findings, concepts, practices, and guidelines are indeed also applicable to apps developed using cross-platform approaches, there is a clear need to stress the importance of security to practitioners, as our survey results clearly indicate. Indeed, with the introduction of the General Data Protection Regulation (*GDPR*), having security measures in place is important for both business and end-user.

As for the more frequently reported issues, topmost we find performance loss in cross-platform apps compared to apps developed using the Native approach. The various subtopics within the *performance* umbrella are frequently discussed in academic literature. Willocx et al. discuss cross-platform app performance in two separate studies [19, 26], in which they report expected penalty in performance when opting for a cross-platform development approach. However, they state that the severeness of the performance penalty does not inherently mean that the performance is unacceptable, in fact, quite the opposite. Based on their empirical performance investigation, the authors report generally acceptable performance and especially make note of high-end devices performing well, an important factor also stressed by Noei et al. [52] in their study on the impact of device attributes on user-perceived app quality. Similar findings are also reported by Ciman and Gaggi [25] in their empirical investigation of energy consumption in the context of apps developed using cross-platform frameworks. The key takeaway from their study is the importance of deep framework knowledge and through that making a decision which technical framework to build on. Similarly to the findings reported by Willocx et al., also Ciman and Gaggi report an inherent performance penalty introduced by the adoption of a cross-platform framework. However, of the frameworks listed in Table 1, Ciman and Gaggi only scrutinize PhoneGap and Titanium in addition to MoSync, whereas the latter was not included in our survey due to deprecation [53]. Willocx et al. scrutinize Xamarin and PhoneGap in their 2015 performance study [26] and proceed to include numerous additional frameworks in their 2016 study [19], e.g., the Ionic Framework, Titanium, and jQuery Mobile, all of which are listed in Table 1. In the same table, the framework popularity of the Ionic Framework, PhoneGap, and React Native are considerably higher than the rest. Research on the latter framework is rarely encountered when traversing the literature, although it has undergone some assessment in recent studies, including [40, 54], although performance was not the core focus of either studies. Nevertheless, the verdict from both studies is that React Native has its pros and cons compared to other frameworks but requires additional scrutiny on a par with that conducted by Willocx et al. and Ciman and Gaggi. From recent studies, we also make note of findings partially contradicting previous research, including Ma et al.'s [55]

study on device performance during HTTP requests in Native and Web-based applications, in which they report on scenarios in which the Web app can outperform Native apps. Thus, we note that the industry's interests do not necessarily correlate to what is presented in academic research.

For the sake of brevity, we simultaneously discuss two of the issues reported in Table 2, being those of "Suboptimal User Experience (UX)" and "suboptimal options for creating good user interfaces (UI)", as these topics are seemingly interlinked in the literature assessed. These concerns are also raised in the academic community surrounding cross-platform mobile development, and we find an insufficient body of knowledge deriving from research. To the best of our knowledge, only one longitudinal end user-oriented study has been conducted to bring forth the opinions of users in terms of the usability of cross-platform apps, being that by Angulo and Ferre [27]. Their experiment involved 37 participants each evaluating a Native and cross-platform app and providing feedback on a variety of mostly nontechnical parameters such as dependability and attractiveness of the apps. The overarching findings reported by Angulo and Ferre essentially point to the importance of choosing a cross-platform framework facilitating the development of apps that follow design and interaction guidelines put forth by the platforms. Nevertheless, they highlight how the Native approach provides a better set of tools for implementing good user interfaces and user experiences. This matches with results from Ahmad et al.'s [37] interview-based study, in which practitioners deemed user experience an issue more critical to Hybrid development than Native development. Other findings related to user experience originating from academia are typically in the context of user interfaces, e.g., Dhillon and Mahmoud's multifocused study in which they discuss PhoneGap, Titanium, Adobe Air, and MoSync and their capabilities regarding, among other topics, user experience [38]. Their discussions are drawn from an overview of framework capabilities within user experience, e.g., availability of accessibility APIs, inclusion of gesture recognition, and access to Native user interface elements. Similarly to the previous studies traversed, Dhillon and Mahmoud also stress the importance of framework choice, as they identified major differences between the frameworks scrutinized, such as PhoneGap's lack of capabilities at the time of writing (2014). The context and projected end-user of the apps should also be taken into account when deciding on app development approach, as Francese et al. report differences in usability requirements between consumer and enterprise apps [35]. This is also mentioned by Lim et al. [56], despite in the context of differences in mobile app usage between countries, in which they find that price and quality expectations vary between users located in the countries investigated. Some studies are also drawn from the experience of having implemented a cross-platform app as part of a larger study, for example, Smith et al.'s [23] study on self-reporting of alcohol consumption, in which they developed a hybrid app in which they deemed to have a medium degree of user-perceived performance, in spite of providing an overall acceptable performance. In more general mobile computing literature, Huang et al. [57] study the user-perceived latency

of Android apps and find that the Android operating system itself introduces constraints on the user experience, an observation that can be critical going forward with further research on the user experience of cross-platform mobile apps. We find that the survey respondents' thoughts on user experience and possibilities of good user interfaces in cross-platform apps, as presented in Table 2, correspond with what is also reported in the literature, whereas the most important factor is seemingly the choice of framework or overarching development approach in relation to a product specification.

Our survey questionnaire also questioned the participants' perception of maturity of cross-platform tools and their associated communities. Both questions received numerous votes, respectively, 55 regarding maturity of tools and 35 regarding surrounding communities. To the best of our knowledge, these questions are typically not raised in the context of research but are seemingly, and understandably, of great importance for practitioners. One exception is Heitkötter et al.'s study on criteria for cross-platform frameworks, in which we do find "long-term feasibility" listed [12]. The rapid pace of development through the continuous advances in the state of the art not only is a challenge for the practitioners responding to our survey but also is indeed a factor we as researchers must also take into account to ensure validity and relevance. As briefly previously mentioned, the MoSync cross-platform framework is to be found in numerous research articles [16] but has received its last update in 2013, more than half a decade ago. In Table 1, MoSync is not included, and none of the respondents answering "Other" and typing their answer manually reported using the framework either. Thus, even though certain frameworks can frequently be encountered in academic research (e.g., [25, 38]), the relevance of the technologies does not necessarily match what is used in the industry. We suggest future research to be drawn from the results presented in Table 1 when deciding on the sample of technical frameworks and thus complying with industry interests.

About one-third of the respondents identified issues related to the integration and use of platform and device features, a topic that is also popularly discussed in the research literature. Examples of such features include programmatic access to, e.g., device camera, contact lists, accelerometer sensor, and GPS, among a wide array [12]. This issue is also frequently examined in the literature, although with some discrepancies in terms of findings. Whereas numerous studies report limited access to underlying platform and device APIs, others report quite the opposite. The importance of such access is often stated in descriptive studies focusing on the possibilities and drawbacks of cross-platform frameworks. Latif et al., for instance, describe access to platform and device features as a requirement for frameworks to facilitate, although they recommend such access to be direct without any abstraction layers [29]. To the best of our knowledge, native-code-generating frameworks, i.e., those of the Model-Driven Development approach, are among the only ones that can facilitate this requirement; e.g., the frameworks listed in Table 1 all rely on some intermediary abstraction layer to provide feature access. Additional to the study of Latif et al., there are also several comparative studies,

in which sets of frameworks are compared and measured on various parameters. Within this category of studies, we find for example Heitkötter et al.'s seminal paper describing a variety of criteria on which one can evaluate cross-platform frameworks [7]. Indeed, they list *access to platform-specific features* as an important criterion and describe the capabilities of PhoneGap and Titanium to well facilitate such access. However, some studies report limitations imposed by frameworks, thus making them comparably less capable for app development than the Native development approach. Examples of studies include [23, 58], stating that feature access is *limited*, a statement also found in [33], describing the (lack of) feature access as a technical shortcoming. However, all three studies may describe the state of the art at the point of publishing, which, due to the rapid pace of development in the field, may no longer be representable. In [40], Biørn-Hansen and Ghinea investigate through artefact development how features are exposed to the cross-platform development environments. While they state that they did not identify any functionality that could not be exposed, thus used in cross-platform apps, the study is limited in comprehensiveness, and they stress the need of more exhaustive studies verifying their findings. We deem the necessity of more research as inherent to sufficiently provide concluding proof on feature accessibility in cross-platform apps and prove that the 32 participants (ref. Table 1) voting for "hard to integrate with device API" are in sync with the state of research.

Lastly to be discussed, the "hard to test/debug" option received 31 votes, only one vote less than the previously discussed issue. Alas, in the literature we rarely encounter mentions of testing and debugging of cross-platform apps. They are both listed as subcriteria in Heitkötter et al.'s [7] comparative study; however, neither are discussed in the context of the frameworks at hand, specifically PhoneGap and Titanium. The same also goes for Latif et al.'s [29] survey-based study on cross-platform development, wherein framework-facilitation of debugging and that of testing are mentioned as important factors, but neither are discussed to any degree. Majchrzak et al.'s study on innovative cross-platform frameworks [18] briefly reports their experience of debugging React Native, the Ionic Framework, and Fuse, although their findings are based on experiences gained through the implementation of numerous artifacts, lacking broader empirical substance.

**5.2. Technology Adoption.** Associated research question: *Which cross-platform frameworks are adopted by the survey participants, and to what degree does professional technology adoption differ from hobby usage?*

In terms of framework interest, React Native scores the highest average score, 4.28 out of 5. This finding matches well with results from other surveys, such as the 2017 State of JavaScript online questionnaire [15], wherein React Native is by far the framework most of the survey participants are interested in learning more about. To the best of our knowledge, React Native is infrequently mentioned in academic research [16], most likely due to the novelty of the framework. Although being reported the "most interesting" framework in our survey results, it is not the most used one in terms of neither hobby nor professional settings, as displayed in



Table 1. In both aforementioned categories, PhoneGap scores the highest. This is also in line with studies on cross-platform framework usage in deployed Android applications [59]. PhoneGap is also the reported second-most interesting framework (3.03 of 5), while being the technology most of the respondents are familiar with (3.35 of 5). The high level of familiarity with PhoneGap reported by respondents may be due to the framework's history, as it was one of the first frameworks of its kind in allowing developers to wrap web-based code (HTML, CSS, and JavaScript) into a native app container, rendering the web code to look and function as if it was written using native technologies [60]. PhoneGap later became a product of Adobe, while the core of PhoneGap, allowing for the wrapping and execution of native code through JavaScript interfaces, was given to the Apache Foundation, open-sourced, and renamed Cordova [61]. Both PhoneGap and Cordova still exist as distinct products, which may potentially have confused survey participants unfamiliar with the aforementioned historical anecdote, as Cordova is the foundation of numerous projects (e.g., the Ionic Framework, Framework7, and OnsenUI), while PhoneGap is its own product.

Our survey results indicate that frameworks often encountered in academic research, e.g., jQuery Mobile, Sencha Touch, Titanium, and Xamarin Forms [16], are not the same frameworks as those practitioners choose and adopt, being those of PhoneGap, the Ionic Framework, and React Native. From industry outlets, we find mentions and recommendations for adopting React Native over Xamarin [62], and the State of JavaScript questionnaire [15] lists Xamarin in their "Other" category, with the less popular frameworks. As displayed in Table 1, PhoneGap's usage is high among the survey participants, while also being frequently encountered in both academic [16] and industry contexts. Thus, further research may include a variety of frameworks listed in Table 1, although decisions on which frameworks to include can now be made with an empirical grounding in our survey results.

In an attempt to understand where practitioners, hence the industry, could possibly be moving towards, we differentiated between *hobby* usage and *professional* usage in our question on framework usage. We find that the Ionic Framework, PhoneGap, and React Native all see significantly more adoption in the hobby-sphere than in professional settings. Although less significantly, Xamarin, NativeScript, and jQuery Mobile also enjoy more usage in hobby projects than in professional contexts. Especially for the three former frameworks, the results could indicate that practitioners are moving towards React Native and the Ionic Framework. We know from previous research that PhoneGap is already widely used in the industry and in deployed apps [59]. However, we have identified fewer academic indications of React Native and the Ionic Framework following the footsteps of PhoneGap. Results from follow-up surveys can help to validate this indication of rising popularity and adoption. A limitation to our study in regard to this question is evident that practitioners may test the feasibility of certain frameworks prior to adopting them into professional projects. Thus, the results may be skewed as a result of this. Nevertheless, the results from this question may act as an indication of coming

changes to the cross-platform landscape and help researchers in deciding on technologies to include in implementation-oriented research.

Another interesting note on technology adoption is the lack of any Model-Driven Development (MDD) frameworks. None of the participants manually enter framework name through the "Other" option mentioned MDD. Similar findings are also reported in other academic works, e.g., Majchrzak et al. [18] commenting on the soundness of MDD designs in the context of the seeming lack of industry adoption. This is further investigated by Gorschek et al. [63], who through their 3785-respondents survey find that the majority of their participants rarely if ever use models. Within mobile development, one reason for the lack of industry adoption could be that the MDD community lacks a standardized domain-specific or modelling language for conducting mobile development, according to Umuhzo and Brambilla [64] in their survey study on model-driven based cross-platform development frameworks. Another reason may be that developers adopting MDD-based frameworks are not active in the online communities in which we announced the survey questionnaire thus were not given a chance to voice their opinions. From an academic perspective, research on and development of MDD-based frameworks are topics frequently discussed in the context of cross-platform mobile development. Studies conducted by Heitkötter et al. [17, 65] on the design and evaluation of the MD2 framework have led to several subsequent studies, e.g., those of Majchrzak et al. [66] discussing MDD in the context of business practicality, Usman et al.'s [67] deeply technical study contributing to the development of product-lines using MDD, and Rieger and Kuchen's [68] proposal of a graphical approach to MDD development of mobile apps. Nevertheless, the popularity of Model-Driven Development in academia is not reflected in the survey questionnaire results presented in Section 4.

## 6. Conclusion

The main objective of the study reported in this paper has been to report the perspectives and thoughts of industry practitioners on technologies related to and the possibilities and constraints of cross-platform mobile development tools. From the existing academic literature, we find that the majority of studies on cross-platform development are of a technical and mathematical nature, experimenting, comparing, and describing technical development frameworks and mining and analysing end-user app reviews. For the study at hand, we conducted an online survey questionnaire, recruiting participants from various online forums and communities related to the development of mobile apps. With 101 respondents, we are able to provide perspectives originating from the industry. The majority of the survey participants reported issues that are also commonly discussed in the academic literature. Issues related to performance, user experience, and the maturity of technical frameworks are those most prominently reported by the survey participants. In particular, performance and user experience are topics frequently discussed in research, while the maturity of frameworks, or the reported lack thereof, is a topic rarely

encountered. Discussing our findings in relation to related research, we find that all the aforementioned topics are in need of additional scrutiny and more conclusive results, as studies often contradict each other, rendering research-backed decision making difficult. On a more worrisome note, few participants perceived security in cross-platform apps to be an issue. As discussed in Section 5, issues related to security indubitably exist and should be accounted for in the creation of cross-platform apps.

In terms of cross-platform framework usage and popularity, survey participants were by far most interested in exploring React Native, PhoneGap, and the Ionic Framework, in that particular order. When questioned which framework they use professionally, PhoneGap is the highest ranked, followed by React Native and the Ionic Framework. The same order is also true for hobby usage, although far more of the participants are developing apps as a hobby opposed to doing it professionally. In the research literature, we find that less-used frameworks are often included for scrutiny in surveys, namely, MoSync and Titanium; however, while the former is deprecated, the latter is used by a handful of the survey participants.

*6.1. Limitations and Suggestions for Further Research.* We acknowledge that our study has limitations in terms of the depth of our questionnaire, as well as the sample nature. While our sample size  $N = 101$  cannot be claimed to be representative of the overall population, the sample is of a similar size to related work, as discussed in Section 4.1. We also acknowledge that our questionnaire is not an in-depth surveying of the participants, rather a point of departure for better understanding the perceptions of practitioners working in the industry. More in-depth follow-up studies can hopefully be drawn from our results for use in the creation survey questionnaires and interviews. One particular suggestion for such studies could be the inclusion of metaquestions allowing for cross-referencing of job position and skill level against framework usage and perceived issues arising with the adoption of cross-platform frameworks.

Continuing with suggestions for further research, the results put forth and discussed throughout this article may all individually and jointly be significant and extensive topics for future work. To ensure external and industry validity in upcoming studies on cross-platform development, researchers may look at Table 1, providing insights as to which frameworks are commonly adopted by practitioners and which frameworks may increase in popularity and usage in the coming years. Concrete topics for further research may be drawn from Table 2, in which the survey participants reported that especially performance, UI and UX penalties are to be expected in cross-platform apps and that the pace of development of new technologies render technical frameworks immature and less attractive to use in projects.

We suggest that future survey questionnaire-based or interview-based researches of similar nature also focus on metaperspectives including work responsibilities, job position, and self-perceived skills. Such information would allow for cross-referencing of technical and personal questionnaire answers, strengthening our understanding of developers'

perceptions accordingly. Following this trail, we also urge fellow researchers interested in building on our results to focus on understanding in-depth perspectives of the commonly perceived issues presented in Table 2.

## Data Availability

The XLSX-based data used to support the findings presented in this study have been deposited in the Github repository located at the following URL: <https://github.com/mobiletechlab/perspectives-industry-survey>.

## Disclosure

This research was conducted as part of the employment of the authors at Kristiania University College, Brunel University London, and the German Jordanian University.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## References

- [1] App Annie, "App annie market forecast 2017," 2017, <https://www.statista.com/statistics/271644/worldwide-free-and-paid-mobile-app-store-downloads/>.
- [2] eMarketer, "Number of smartphone users worldwide 2014-2020," 2016, <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>.
- [3] V. J. Reddi, H. Yoon, and A. Knies, "Two Billion Devices and Counting," *IEEE Micro*, vol. 38, no. 1, pp. 6–21, 2018.
- [4] D. Loesche, "Infographic: The biggest app stores," 2018, <https://www.statista.com/chart/12455/number-of-apps-available-in-leading-app-stores/>.
- [5] App Annie, "Mobile app revenues 2015-2020," 2016, <https://www.statista.com/statistics/269025/worldwide-mobile-app-revenue-forecast/>.
- [6] M. Ciman, O. Gaggi, and N. Gonzo, "Cross-platform mobile development," in *Proceedings of the the 29th Annual ACM Symposium*, pp. 757–759, Gyeongju, Republic of Korea, March 2014.
- [7] H. Heitkötter, S. Hanschke, and T. A. Majchrzak, "Evaluating cross-platform development approaches for mobile applications," in *Web Information Systems and Technologies: 8th International Conference, WEBIST 2012, Porto, Portugal, April 18–21, 2012, Revised Selected Papers*, vol. 140 of *Lecture Notes in Business Information Processing*, pp. 120–138, Springer, Berlin, Germany, 2013.
- [8] T.-M. Grønli, J. Hansen, G. Ghinea, and M. Younas, "Mobile application platform heterogeneity: android vs windows phone vs iOS vs Firefox OS," in *Proceedings of the 28th IEEE International Conference on Advanced Information Networking and Applications (AINA '14)*, pp. 635–641, May 2014.
- [9] C. Escoffier and P. Lalanda, "Managing the Heterogeneity and Dynamism in Hybrid Mobile Applications," in *Proceedings of the 2015 IEEE International Conference on Services Computing (SCC)*, pp. 74–81, New York City, NY, USA, June 2015.

- [10] W. S. El-Kassas, B. A. Abdullah, A. H. Yousef, and A. M. Wahba, "Taxonomy of Cross-Platform Mobile Applications Development Approaches," *Ain Shams Engineering Journal*, vol. 8, no. 2, pp. 163–190, 2017.
- [11] M. Shafirov, "Kotlin on android. now official," 2017, <https://blog.jetbrains.com/kotlin/2017/05/kotlin-on-android-now-official/>.
- [12] H. Heitkötter, S. Hanschke, and T. A. Majchrzak, "Comparing cross-platform development approaches for mobile applications," in *Proceedings of the 8th International Conference on Web Information Systems and Technologies, WEBIST 2012*, pp. 299–311, Portugal, April 2012.
- [13] S. Chadha, A. Byalik, E. Tilevich, and A. Rozovskaya, "Facilitating the development of cross-platform software via automated code synthesis from web-based programming resources," *Computer Languages, Systems & Structures*, vol. 48, pp. 3–19, 2017.
- [14] W. S. El-Kassas, B. A. Abdullah, A. H. Yousef, and A. M. Wahba, "Enhanced Code Conversion Approach for the Integrated Cross-Platform Mobile Development (ICPMD)," *IEEE Transactions on Software Engineering*, vol. 42, no. 11, pp. 1036–1053, 2016.
- [15] S. Greif, R. Benitte, and M. Rambeau, "Mobile & desktop frameworks," 2017, <https://stateofjs.com/2017/mobile/results>.
- [16] A. Biørn-Hansen, T.-M. Grønli, and G. Ghinea, "Baseline requirements for comparative research on Cross-Platform mobile development: A literature survey," in *Proceedings of the 30th Norwegian Informatics Conference*, 2017.
- [17] H. Heitkötter, T. A. Majchrzak, and H. Kuchen, "Cross-platform model-driven development of mobile applications with md2," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC '13*, pp. 526–533, Coimbra, Portugal, March 2013.
- [18] T. Majchrzak, A. Biørn-Hansen, and T. Grønli, "Comprehensive Analysis of Innovative Cross-Platform App Development Frameworks," in *Proceedings of the Hawaii International Conference on System Sciences*, pp. 6162–6171, 2017.
- [19] M. Willocx, J. Vossaert, and V. Naessens, "Comparing performance parameters of mobile app development strategies," in *Proceedings of the the International Workshop*, pp. 38–47, Austin, Texas, May 2016.
- [20] J. Dann, "Under the hood: Rebuilding facebook for iOS," 2012, <https://www.facebook.com/notes/facebook-engineering/under-the-hood-rebuilding-facebook-for-ios/10151036091753920/>.
- [21] F. Qixing DU, "Under the hood: Rebuilding facebook for android," 2012, <https://www.facebook.com/notes/facebook-engineering/under-the-hood-rebuilding-facebook-for-android/10151189598933920/>.
- [22] I. T. Mercado, N. Munaiah, and A. Meneely, "The impact of cross-platform development approaches for mobile applications from the user's perspective," in *Proceedings of the the International Workshop*, pp. 43–49, Seattle, WA, USA, November 2016.
- [23] A. Smith, K. de Salas, I. Lewis, and B. Schüz, "Developing smartphone apps for behavioural studies: The AlcoRisk app case study," *Journal of Biomedical Informatics*, vol. 72, pp. 108–119, 2017.
- [24] L. Corral, A. Sillitti, and G. Succi, "Mobile Multiplatform Development: An Experiment for Performance Analysis," *Procedia Computer Science*, vol. 10, pp. 736–743, 2012.
- [25] M. Ciman and O. Gaggi, "An empirical analysis of energy consumption of cross-platform frameworks for mobile development," *Pervasive and Mobile Computing*, vol. 39, pp. 214–230, 2017.
- [26] M. Willocx, J. Vossaert, and V. Naessens, "A Quantitative Assessment of Performance in Mobile App Development Tools," in *Proceedings of the 3rd IEEE International Conference on Mobile Services, MS 2015*, pp. 454–461, USA, July 2015.
- [27] E. Angulo and X. Ferre, "A Case Study on Cross-Platform Development Frameworks for Mobile Applications and UX," in *Proceedings of the the XV International Conference*, pp. 1–8, Puerto de la Cruz, Tenerife, Spain, September 2014.
- [28] C. Rieger and T. A. Majchrzak, "Conquering the mobile device jungle: Towards a taxonomy for app-enabled devices," in *Proceedings of the 13th International Conference on Web Information Systems and Technologies, WEBIST 2017*, pp. 332–339, Portugal, April 2017.
- [29] M. Latif, Y. Lakhrissi, E. H. Nfaoui, and N. Es-Sbai, "Cross platform approach for mobile application development: A survey," in *Proceedings of the International Conference on Information Technology for Organizations Development, IT4OD 2016*, Morocco, April 2016.
- [30] L. Delia, N. Galdamez, P. Thomas, L. Corbalan, and P. Pesado, "Multi-platform mobile application development analysis," in *Proceedings of the 9th IEEE International Conference on Research Challenges in Information Science, IEEE RCIS 2015*, pp. 181–186, Greece, May 2015.
- [31] J. Perchat, M. Desertot, and S. Lecomte, "Component based framework to create mobile cross-platform applications," in *Proceedings of the 4th International Conference on Ambient Systems, Networks and Technologies, ANT 2013 and the 3rd International Conference on Sustainable Energy Information Technology, SEIT 2013*, pp. 1004–1011, Canada, June 2013.
- [32] I. Malavolta, S. Ruberto, T. Soru, and V. Terragni, "End Users' Perception of Hybrid Mobile Apps in the Google Play Store," in *Proceedings of the 3rd IEEE International Conference on Mobile Services, MS 2015*, pp. 25–32, USA, July 2015.
- [33] I. Malavolta, S. Ruberto, T. Soru, and V. Terragni, "Hybrid Mobile Apps in the Google Play Store: An Exploratory Investigation," in *Proceedings of the 2nd ACM International Conference on Mobile Software Engineering and Systems, MOBILESoft 2015*, pp. 56–59, Italy, May 2015.
- [34] M. Ali and A. Mesbah, "Mining and characterizing hybrid apps," in *Proceedings of the 1st International Workshop on App Market Analytics, WAMA 2016*, pp. 50–56, USA.
- [35] R. Francese, C. Gravino, M. Risi, G. Scanniello, and G. Tortora, "Mobile App Development and Management: Results from a Qualitative Investigation," in *Proceedings of the 4th IEEE/ACM International Conference on Mobile Software Engineering and Systems, MOBILESoft 2017*, pp. 133–143, Argentina, May 2017.
- [36] A. Puvvala, A. Dutta, R. Roy, and P. Seetharaman, "Mobile application developers' platform choice model," in *Proceedings of the 49th Annual Hawaii International Conference on System Sciences, HICSS 2016*, pp. 5721–5730, USA, January 2016.
- [37] A. Ahmad, K. Li, C. Feng, S. M. Asim, A. Yousif, and S. Ge, "An Empirical Study of Investigating Mobile Applications Development Challenges," *IEEE Access*, pp. 17711–17728, 2018.
- [38] S. Dhillon and Q. H. Mahmoud, "An evaluation framework for cross-platform mobile application development tools," *Software: Practice and Experience*, vol. 45, no. 10, pp. 1331–1357, 2015.
- [39] E. Taylor-Powell, "Questionnaire design: Asking questions with a purpose," 1998, <https://learningstore.uwex.edu/Assets/pdfs/G3658-02.pdf>.
- [40] A. Biørn-Hansen and G. Ghinea, "Bridging the Gap: Investigating Device-Feature Exposure in Cross-Platform Development,"

- in *Proceedings of the Hawaii International Conference on System Sciences*, pp. 5717–5724, 2017.
- [41] T. Luo, H. Hao, W. Du, Y. Wang, and H. Yin, “Attacks on WebView in the Android system,” in *Proceedings of the the 27th Annual Computer Security Applications Conference*, pp. 343–352, Orlando, Fla, USA, December 2011.
  - [42] J. Yu and T. Yamauchi, “Access control to prevent attacks exploiting vulnerabilities of WebView in android OS,” in *Proceedings of the 15th IEEE International Conference on High Performance Computing and Communications, HPCC 2013 and 11th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, EUC 2013*, pp. 1628–1633, China, November 2013.
  - [43] D. Sun, C. Guo, D. Zhu, and W. Feng, “Secure HybridApp: A detection method on the risk of privacy leakage in HTML5 hybrid applications based on dynamic taint tracking,” in *Proceedings of the 2nd IEEE International Conference on Computer and Communications, ICC3 2016*, pp. 2771–2775, China, October 2016.
  - [44] P. Faruki, A. Bharmal, V. Laxmi et al., “Android security: A survey of issues, malware penetration, and defenses,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 998–1022, 2015.
  - [45] Sufatrio, D. J. J. Tan, T.-W. Chua, and V. L. L. Thing, “Securing android: a survey, taxonomy, and challenges,” *ACM Computing Surveys*, vol. 47, no. 4, article 58, 2015.
  - [46] A. Papageorgiou, M. Strigkos, E. Politou, E. Alepis, A. Solanas, and C. Patsakis, “Security and Privacy Analysis of Mobile Health Applications: The Alarming State of Practice,” *IEEE Access*, vol. 6, pp. 9390–9403, 2018.
  - [47] D. He, S. Chan, and M. Guizani, “Mobile application security: Malware threats and defenses,” *IEEE Wireless Communications Magazine*, vol. 22, no. 1, pp. 138–144, 2015.
  - [48] H. Wang, Y. Li, Y. Guo, Y. Agarwal, and J. I. Hong, “Understanding the Purpose of Permission Use in Mobile Apps,” *ACM Transactions on Information and System Security*, vol. 35, no. 4, pp. 1–40, 2017.
  - [49] S. Chitkara, N. Gothoskar, S. Harish, J. I. Hong, and Y. Agarwal, “Does this App Really Need My Location?” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 3, pp. 1–22, 2017.
  - [50] W. Martin, F. Sarro, Y. Jia, Y. Zhang, and M. Harman, “A survey of app store analysis for software engineering,” *IEEE Transactions on Software Engineering*, vol. 43, no. 9, pp. 817–847, 2017.
  - [51] I. J. Mojica Ruiz, M. Nagappan, B. Adams, T. Berger, S. Dienst, and A. E. Hassan, “Impact of ad libraries on ratings of android mobile apps,” *IEEE Software*, vol. 31, no. 6, pp. 86–92, 2014.
  - [52] E. Noei, M. D. Syer, Y. Zou, A. E. Hassan, and I. Keivanloo, “A study of the relation of mobile device attributes with the user-perceived quality of Android apps,” *Empirical Software Engineering*, vol. 22, no. 6, pp. 3088–3116, 2017.
  - [53] MoSync, 2013, <https://github.com/MoSync/MoSync>.
  - [54] A. Biørn-Hansen, T. A. Majchrzak, and T.-M. Grønli, “Progressive web apps: The possible web-native unifier for mobile development,” in *Proceedings of the 13th International Conference on Web Information Systems and Technologies, SCITEPRESS*, T. A. Majchrzak, P. Traverso, K.-H. Krempels, and V. Monfort, Eds., pp. 344–351, 2017.
  - [55] Y. Ma, X. Liu, Y. Liu, Y. Liu, and G. Huang, “A Tale of Two Fashions: An Empirical Study on the Performance of Native Apps and Web Apps on Android,” *IEEE Transactions on Mobile Computing*, vol. 17, no. 5, pp. 990–1003, 2017.
  - [56] S. L. Lim, P. J. Bentley, N. Kanakam, F. Ishikawa, and S. Honiden, “Investigating country differences in mobile app user behavior and challenges for software engineering,” *IEEE Transactions on Software Engineering*, vol. 41, no. 1, pp. 40–64, 2015.
  - [57] G. Huang, M. Xu, F. X. Lin et al., “ShuffleDog: Characterizing and Adapting User-Perceived Latency of Android Apps,” *IEEE Transactions on Mobile Computing*, vol. 16, no. 10, pp. 2913–2926, 2017.
  - [58] S. Xanthopoulos and S. Xinogalos, “A comparative analysis of cross-platform development approaches for mobile applications,” in *Proceedings of the 6th Balkan Conference in Informatics (BCI '13)*, pp. 213–220, Thessaloniki, Greece, September 2013.
  - [59] N. Viennot, E. Garcia, and J. Nieh, “A measurement study of google play,” in *Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '14)*, pp. 221–233, ACM, 2014.
  - [60] L. Davis, “PhoneGap: People’s choice winner at web 2.0 expo launch pad - ReadWrite,” 2009, [https://readwrite.com/2009/04/02/phone\\_gap\\_todays\\_peoples\\_choice\\_winner\\_at\\_launch\\_p/](https://readwrite.com/2009/04/02/phone_gap_todays_peoples_choice_winner_at_launch_p/).
  - [61] B. LeRoux, “PhoneGap, cordova, and what’s in a name?” 2012, <https://phonegap.com/blog/2012/03/19/phonegap-cordova-and-whate28099s-in-a-name/>.
  - [62] “Bekk Consulting, Tools,” 2018, <https://radar.bekk.no/tech2018/verktoy/xamarin>.
  - [63] T. Gorschek, E. Tempero, and L. Angelis, “On the use of software design models in software development practice: An empirical investigation,” *The Journal of Systems and Software*, vol. 95, pp. 176–193, 2014.
  - [64] E. Umuhzo and M. Brambilla, “Model driven development approaches for mobile applications: a survey,” in *Mobile Web and Intelligent Information Systems*, vol. 9847 of *Lecture Notes in Computer Science*, pp. 93–107, Springer International Publishing, Cham, Germany, 2016.
  - [65] H. Heitkötter and T. A. Majchrzak, “Cross-Platform development of business apps with MD2,” in *Design Science at the Intersection of Physical and Virtual Design*, Lecture Notes in Computer Science, pp. 405–411, Springer, Berlin, Germany, 2013.
  - [66] T. A. Majchrzak, J. Ernsting, and H. Kuchen, “Achieving business practicability of model-driven cross-platform apps,” *Open Journal of Information Systems*, vol. 2, no. 2, pp. 3–14, 2015.
  - [67] M. Usman, M. Z. Iqbal, and M. U. Khan, “A product-line model-driven engineering approach for generating feature-based mobile applications,” *The Journal of Systems and Software*, vol. 123, pp. 1–32, 2017.
  - [68] C. Rieger and H. Kuchen, “A process-oriented modeling approach for graphical development of mobile business apps,” *Computer Languages, Systems and Structures*, vol. 53, pp. 43–58, 2018.



**Hindawi**

Submit your manuscripts at  
[www.hindawi.com](http://www.hindawi.com)

