ISSN: 2583-6749 (Online)

DTC Journal of Computational Intelligence, Vol-2, Issue-1

Approaches and Models Integral to Mobile Application Development: A Systematic Review

Apoorva Garg[1*], Dr. Anuradha[2]

[1]School of Engineering and Technology, The NorthCap University, Gurugram, India
[2]Assistant Professor, Department of Computer Science and Engineering, The NorthCap University, Gurugram, India
*apoorva20csu021@ncuindia.edu

Abstract

Mobile application development has witnessed a significant transformation over the years, with various techniques emerging to cater to the diverse needs of developers and businesses. To reduce complexity and help achieve abstract solutions, it is essential to understand the software development process and the user-centered design principles that guide the development. The main objective of this paper is to review existing approaches and models to identify how each technique affects the productivity of the process. By understanding the common challenges faced during the process of mobile app development, three primary areas of study were identified. This paper presents a decade's overview of the developments pertaining to these three fields. Based on the analysis, this paper concludes that cross-platform development is the best-suited technique for modern mobile app development. By mitigating the shortcomings of native and hybrid approaches, cross-platform empowers developers to build high-performance, feature-rich applications without compromising user experience and platform compatibility. Moreover, a model-driven approach enables developers to capture the essential aspects of an application and subsequently utilize them to generate the corresponding code, supporting rapid development. The study further underscores the significance of using cross-platform frameworks in conjunction with Agile methodologies, which results in faster feedback loops and greater customer satisfaction.

Keywords: Mobile Application Development, Model Driven Development, Cross-platform development, Agile Methodologies, Agile MDD

1. Introduction

Apps have revolutionized the way we communicate, make purchases, access information, play games, etc., and have become a fundamental part of our lives. With the surging demand for mobile apps, businesses of all sizes have been investing in app development to reach new customers and provide better experiences to existing ones. Figure 1 describes the statistics of mobile market shares in India and how they are expected to grow at an annual rate of 7.13% [1].

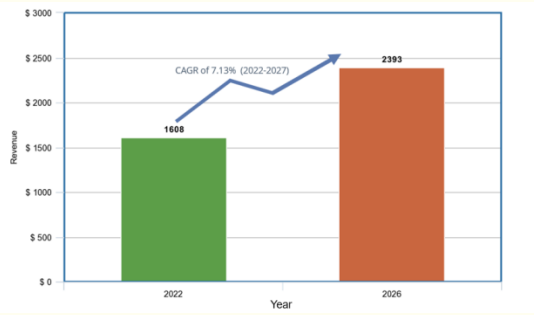

Figure 1 Increasing shares of the mobile market in India

For a field that is critical to the success of organizations and businesses in today's extremely digital world, it is essential to study the different approaches for development to gain a deeper insight into their strengths and limitations. Studying the various techniques can assist developers in identifying the most appropriate tools, frameworks, and technologies for their projects, as well as help to understand the trade-offs and challenges involved in each.

The need for this review stems from the literature study done on the challenges and obstacles of mobile app development. One of the most common challenges is the problem of software reuse i.e., the choice of mobile application implementation strategy and the heterogeneity of Mobile OS [2]. Initially, the only way to build apps was through native development. As needs changed, other approaches, namely Hybrid, Cross-Platform and Progressive Web Apps (PWA) were also introduced. But despite the obvious drawbacks: fragmentation, lack of supporting tools for testing and analysis, and frequent SDK updates [3], native solutions remain a key player in the mobile app market, and it continues to evolve as technology and user expectations change.

When designing an app, it is only logical to want to target both the major operating systems and their platforms. Native development involves using the platform-dependent supported

languages and APIs directly while the Hybrid technique involves using a combination of technologies like HTML, CSS, JavaScript, and native platforms to build an app that runs on more than one platform. The Cross-platform approach rather aims to attain the goal of "write once run everywhere" and involves using a single codebase to build a platform-independent app [4]. Aside from the direct comparisons, the UX, stability, simplicity of updating, and cost of development are just a few other variables to consider.

Other productivity factors responsible for a high-quality mobile application include adaptation to changing requirements, emphasis on interface components, and short development and deployment time [5]. It has been highlighted in [6] that because the industry necessitates so much importance to the end-user's perspectives, it enforces a swift cycle of demand and launch. As a consequence, the industry deviates from following the traditional development life cycle.

The advancement and evolution of software development methodologies have been taking place ever since the seventies and can be traced back to the earliest days of computer programming when software was developed using ad-hoc methods and no formal processes were in place. Over the years, the models have evolved to reflect the ever-changing requirements of the industry. Agile is one of the most notable and popular methods followed during software development [7, 8] because it prioritizes flexibility and responsiveness by making the development process iterative and incremental.

The above-mentioned factors are all directly responsible for other challenges such as delivery of quality applications and uncertain requirements [9] which can be eliminated through a good model-based development. Model Driven Development (MDD) is a development paradigm that puts models in the middle of the entire development process. It addresses executability or the potential for producing executable applications from modeling efforts. In this approach, the implementation/execution is typically semi-automatically created from the provided models, allowing for improvements in software quality and productivity [10].

To summarize, by understanding the common challenges faced by developers, three areas were identified that seemed to affect the process largely. Thus, establishing the following research questions:
RQ1: Which mobile app development approach is the most productive?
RQ2: What is Model-Driven Development?
RQ3: Which is the most suited Software Engineering Process Model to use?

The rest of the manuscript is structured as follows: Section 2 covers the literature review/related works. Section 3 briefly describes the methodology of the paper. Section 4

delves into the main discussion with subsection 4.1 talking about the approaches of development – further divided into 4.1.1, 4.1.2, 4.1.3, 4.1.4, and focusing on Native, Hybrid, and Cross-platform development and their evaluation respectively. Section 4.2 briefly describes Model Driven Development. Section 4.3 delves into the software engineering process models and Section 4.3.1 into Agile methodology, Section 4.3.2 into Agile MDD specifically. Section 5 lays out a summary and draws a conclusion. Section 6 discusses the future scope.

2. Related Works

Many related works can be identified, and each discusses the identified areas of study or research questions individually. Though no academic papers were identified that talked about all of them in conjunction, that too in the context of Mobile Application Development, Table 1, 2, and 3 provide a summary of the literature study conducted.

Table 1 Literature Review of Mobile App Development Approaches

Author	Approach Discussed	Objective	Findings
Holzer, Adrian, and Jan Ondrus [11]	Native	Performs a multi-criteria evaluation of development strategies based on their overall performance.	Native approach performed best.
Huynh, Minh Q. et al. [12]	Hybrid	Investigates a potential method for creating hybrid and cross-platform applications.	Recommends Ionic framework to practitioners.
Nawrocki, Piotr, et al. [13]	Cross-platform	Compares native and cross-platform frameworks.	Flutter, a cross-platform approach offered the best performance.
Biørn-Hansen, et al. [14]	Progressive Web Apps	Performs feature and performance comparison of PWA and other approaches.	PWAs performed best in feature comparison and trade-offs are illustrated.

Table 2 Literature review of Model Driven Development

Author	Objective	Findings
Umuhoza, Eric, and Marco	Establishes a set of standards for evaluating the current model-	Survey results show a preference of native applications and code

Brambilla [15]	driven methods for developing mobile apps.	generation.
Shamsujjoha, Md, et al. [16]	Identifies key benefits, limitations and gaps of existing MDD techniques.	The analysis gives importance to architecture, domain model, and code generation.
Tufail, Hanny, et al. [17]	Identifies 11 MDD models, and 21 MDD tools and subsequently presents an analysis of them.	Concluded that OutSystems and MDA SMARTAPP are the best option for MDD of mobile applications.
Heitkötter, Henning et al. [18]	Proposes MD^2 for model-driven development of cross-platform apps.	Successfully highlights the features and applications of MD^2
Francese, Rita, et al. [19]	Proposes an MDD approach for portable applications.	The approach uses native device features.

Table 3 Literature review of Software Engineering Process Models

Author	Objective	Findings
Despa, Mihai Liviu. [20]	Discusses the characteristics, strengths, and weaknesses of each SE model.	Successfully illustrates when to use lightweight and heavyweight methodologies.
Ashishdeep, A. et al. [5]	Discusses Software process models particular to Mobile App Development	The literature survey proves that agile methodologies are the most suited.
Jabangwe, Ronald et al. [21]	Provides a substantial review of existing models used in mobile app development.	The models frequently use agile techniques or state-based principles.
Shylesh, S. [22]	Discusses the various SE models	Illustrates what scenarios each SDLC models are best in.

3. Methodology

In this study, a manual search was carried out within the listed digital databases and libraries: ScienceDirect (Elsevier), ACM Digital Libra, IEEE Xplore, and Springer. Google Scholar was also used to perform simple searches across an array of publishing formats and disciplines. Table 4 summarizes the number of search results from the digital databases pertaining to the areas of study and their relevant keywords.

Table 4 Search results from the digital libraries.

S.No	Search String	Number of results			
		IEEE	ScienceDirect	Springer	ACM
1.	"Mobile" AND "application" AND "development"	7, 732	2, 135	2, 840	3, 155
2.	"Native" AND "app" AND "development"	111	1, 257	2, 922	2, 907
3.	"Hybrid" AND "app" AND "development"	98	1, 376	2, 919	2, 219
4.	"Cross-platform" AND "app" AND "development"	88	1, 633	4, 087	3, 615
5.	"Model" AND "driven" AND "development"	14, 781	6, 036	7, 605	3, 742
6.	"Agile" AND "Methodologies"	1, 224	2, 837	2, 421	1, 397
7.	"Agile" AND "MDD"	16	145	462	1, 220

To ensure the retrieval of the most relevant papers, search strings with even more adaptations were used. The papers were filtered out based on their relevance and the year in which they were published. The 42 filtered academic papers present a decade's review of the developments important to the area of study. Papers reviewed that are older than that are present to discuss the relevant state-of-the-art literature.

4. Discussion and Analysis

4.1 Approaches to Mobile Application Development

4.1.1 Native App Development

Native Mobile App Development involves the use of those tools, languages, and mobile software development kits (SDKs) that are exclusive to a given mobile OS. Google's Android, Windows and Apple's iOS are the prime mobile OS, with Blackberry OS growing more obsolete each year. Table 5 describes the tools used for the respective operating systems.

Table 5 Tools used for Native Development for each OS

OS	Programming Language	Deliverable File	Development Environment

Android	JAVA, Kotlin	.apk	Android Studio, Android SDK, IntelliJ, Eclipse
iOS	Objective-C / Swift / Ruby	.ipa	XCode
Windows	Visual C#, C++	.xap	Visual Studio, Windows SDK
Blackberry	JAVA	.bar	Eclipse (BlackBerry Plug-in)

This approach results in apps that are optimized for the target platform, providing a smooth and seamless user experience by taking advantage of the platform's distinct aspects and features. Applications created using Native Development support every UI and interaction offered by the relevant operating environment and have unhindered access to device hardware [23]. With common operations like multi-touch, quicker graphic APIs, built-in components, simplicity of usage, and fluid animation, they offer the best overall experience. Many of these contribute to the requirement for complex computational techniques and highly interactive reporting [24]. A user can interact with a smartphone using intricate UI motions thanks to the multi-touch functionality.

Besides offering benefits in a wide variety of domains, native applications call for massive investments in terms of time and money. According to SpdLoad, individually, the price of iOS app development ranges from $75,000 to $500,000 based on the intricacy of each idea and its complexity. While the cost of an Android application costs another $70,000 to $500,000 [25]. This highlights a major drawback of the native approach, which is the lack of portability. This results in increased costs in each phase of the software engineering lifecycle, from writing and testing to maintenance.

4.1.2 Hybrid App Development

Since the disadvantages of Native development were almost on par with its advantages, the need for a less complex and inexpensive alternative started to rise. In 2009, Apache Cordova was released and was subsequently rebranded as PhoneGap by Adobe Systems, only to be changed back to "Apache Cordova" [26]. Cordova offers "a container for running the code inside a native", device-specific wrapper. As a sort of stand-in for the interface, Cordova offers its own WebView for the program to execute.

This hybrid approach enables the abstraction of application models and improves user experience across numerous devices [27]. It acts as a middle ground between Native apps and web applications. Effectively it also minimizes time and development costs as different platforms do not need different applications. Hybrid platform applications are also comparatively maintainable as long as the developer selects the correct framework (examples include JQuery mobile, Ionic, etc.) as it is easier to maintain web app technologies like JavaScript, HTML, and CSS than a native platform application.

Hybrid apps however offer a limited user interface: their design doesn't have a native feel. As a result, the UI is not as seamless as expected. Since Web View is employed and thus prevents the devices' full capacity from being utilized, possibilities like 3D are also constrained [27]. Hybrid applications may also require the user to click a certain button twice or more before receiving a response, which possibly could leave end users dissatisfied [28]. Furthermore, hybrid applications could end up being slower because they need to use a native container to run.

4.1.3 Cross-Platform Mobile App Development

The need for cross-platform solutions arose with the emerging need for apps that could run on more than one platform with only one codebase. When questioned about which framework people use professionally, React Native, Ionic Framework, Xamarin, and Flutter were some frameworks popularly named [29]. Rapidly growing frameworks like Flutter fall under the "Widget Based Approaches" that define the structural elements that make up the application. Instead of relying on the native components of the device, it uses its own rendering engine to construct its own components dynamically by translating the application code (such as JS or Dart) into native Code. Particularly in Flutter, a feature called AoT (or Ahead of Time) Compilation compiles the Dart code into Native code [30].

The cross-platform approach may suffer from lags with slower CPU render time, so the performance is often degraded as compared to its Native counterparts [31]. The main advantage includes the massive cross-cutting of development expenses as future updates and bug/security fixes can also be rolled out once for all affected platforms.

4.1.4 Evaluation

According to Microsoft, when comparing Hybrid and Cross-platform, cross-platform solutions are rather preferred. These frameworks continuously grow in robustness, showcasing increased consistency and often coming impressively close to the native user experience. For instance, many retailers employ this strategy to power fully functional cross-platform e-commerce apps because it leverages web-based programming [32]. Yet, native apps are frequently chosen when it comes to performance-related factors, such as access to platform APIs, simplicity of update, UI rendering, and better UX [31].

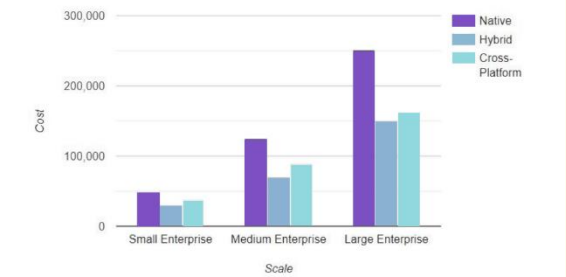

Figure 3 Comparison of App Development Costs

Figure 4. describes the comparison of the average development costs of the three discussed approaches. The native approach costs the highest as it requires a separate team for each OS. Though hybrid apps are the least expensive, they can sometimes have performance issues, and it can be challenging to create a consistent UI across platforms. Cross-platform mobile applications have come a long way in narrowing the gap with their native counterparts and may not be 100% identical to native apps, but they have come remarkably close. This makes them a compelling choice for many businesses and developers due to the advantages they offer in terms of cost-effectiveness, faster development, and code reusability across multiple platforms.

4.2 Model-Driven Development (MDD)

App development can be complex, depending on the scope, features, and requirements of the project. The constantly evolving technology and market demands also add complexity to the development process. MDD approaches enable code synthesis through a model transformation process, which makes the process of designing an app more productive. MDD techniques help in building an app more efficiently, as they "enable code synthesis through a model transformation process" [16]. The technique is used to specify the desired system so the source code can be automatically produced. MDD was created to increase abstraction and automate more of the code generation process.

Mendix, a low-code platform for application development describes MDD as a "methodology that simplifies complex app building through the process of abstraction and automation" [33]. The analysis also revealed that the three aspects of productivity, scalability, and dependability can profit from these methodologies and that the most important goals in MDD-based app development are "architecture, domain model, and code generation" [16].

Model Driven Architecture (MDA) is an approach based on and according to the Object Management Group's standard, which defines a method for creating software systems using models as the primary artifacts. A detailed comparative analysis of 21 identified MDD tools and 8 modeling languages is done in [17] where MDD tools are analyzed on predefined criteria and OutSystems and MDA SMARTAPP are deduced as the best option for MDD of mobile apps. Section 4.3.2 of the paper further delves into the integration of MDD and Agile.

4.3 Software Engineering Process Models

Software Engineering (SE) models provide a framework that outlines and details the steps and tasks involved in the development process of software. Particular to mobile app development, SE models provide a structured and systematic approach to design and development. A development plan must be robust enough to consider the end-user expectations, market conditions, and other important drivers. To keep up with the market's development and the level of intense competition, it must be aggressive and adaptable at the same time [34].

Due to the limitations of traditional models like Waterfall, Agile Model has gained popularity in recent years. Figure 5. present the statistics from the 16Th annual State of Agile Report, 2022 [35]. Evidently, Agile was highly preferred by the respondents. The Agile Model is iterative and emphasizes collaboration and adaptability, which makes it well-suited for complex software development projects.

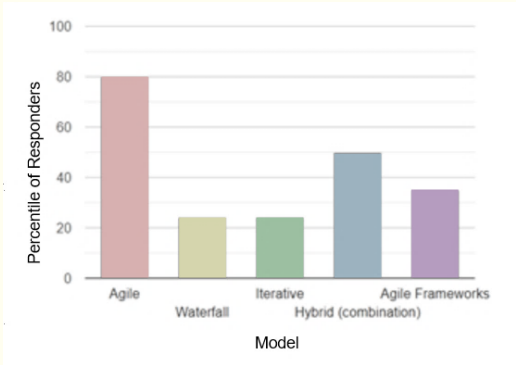

Figure 4 Comparison of Software Engineering Process Models used.

4.3.1 Agile Methodologies

The Agile approach is often suggested as a perfect fit for mobile applications as findings [36] propose that these methods help in delivering improved quality and speed. The “iterative and

incremental paradigm" of the development process serves as the foundation for this methodology. Agile methodology provides flexibility to mobile app development teams to adapt to changing requirements and market conditions. With this approach, developers can quickly pivot and make changes to the app, ensuring it meets the changing demands of users and the market. It also emphasizes shorter development cycles, allowing mobile app development teams to deliver working software early and frequently. This faster time-to-market helps businesses launch their mobile applications sooner, gain a competitive advantage, and generate revenue more quickly.

Some existing agile software development methods are XP, Scrum, Kanban, DSDM, AUP, ASD, FDD, AM, and Lean [36]. Figure 6. presents the statistics from the 16Th annual State of Agile Report, 2022 [35] and depicts the popularity rank of each Agile methodology.

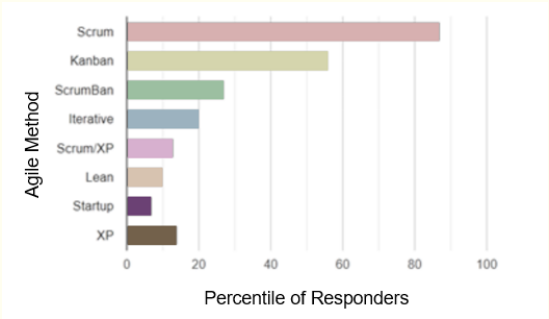

Figure 5 Comparison of Agile Methodologies Used.

Lean methodology is a process improvement approach that focuses on minimizing waste and maximizing value. Integrating Agile and Lean can help organizations streamline their software development processes, reduce waste, and improve efficiency. Both methodologies emphasize continuous improvement, collaboration, and customer value, making them highly complementary.

Scrum is also an Agile framework that emphasizes teamwork, collaboration, and iterative development. Integrating Agile and Scrum can help organizations manage complex projects and ensure a more efficient development process. Scrum provides a set of guidelines for implementing Agile, such as sprint planning, daily stand-ups, and retrospectives. Scrum, which focuses on project management, for instance, has inherent advantages in the creation of mobile applications because to its high level of user contact, disciplined and constrained scope, and shortened time to market [36]. While creating mobile applications, mobile businesses and even lone developers use a Scrum-like methodology [37].

Table 6 Agile processes and techniques.

Author	Practice	Context	Key Features	Identified Gaps
Scharff, C., Verma, R.; [38]	SCRUM	Assumes activity-centeredness of apps	Uses iterations of fixed duration and backlogs are forwarded.	Must be created in a timely manner by small teams.
Alsaqqa, S., Abdel-Nabi, H. & Sawalha, S., [39]	XP	Suitable for object-oriented project, consists of 6 phases.	Small iterations with few quick reactions	Lacks support for distributed teams
Ozkan, Necmettin, et al [40]	Kanban	Methodology independent	More efficient in distributed context	Less efficient predictability and overview of projects.
Rodríguez, P., Mäntylä, M. et al [41]	Lean	Prioritizes delivery of critical features	Use of cross-functional teams and waste elimination	Involving management in development tasks is difficult
Al-Ratrout, Seiren, et al. [42]	AM	Agile version of Model Driven Development	Rapid, continuous delivery	Lack of documentation
Abdelaziz, A. A., El-Tahir, Y., & Osman, R. [43]	ASD	Works in a non-production environment	Development of safety systems	Potential issues with applications of a larger scale
Tirumala, S. S., Shahid Ali, and Anjan Babu [44]	FDD	Operates in feature groups	Workable and independent features for every iteration	Lack of ability to deliver on time
Zafar, Iqra, Aiman Khan Nazir et al [45]	DSDM	Project-oriented methodology	Effective cost and time and risk management	A major concern on “fitness for use”

4.3.2 Agile MDD

Since MDD can be looked upon as a "heavyweight process", developers frequently reassess its guidelines and tend to make it more lightweight by introducing agility to the process. In a study [46] done to understand how exactly the integration of Agile methods and Model Driven Development plays out, UML turned out to be the most used modeling approach and Scrum to be the most used agile practice in amalgamation with MDD. Figure 7 briefly describes the Agile MDD Process [46].

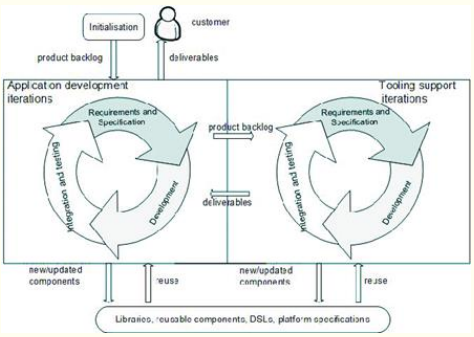

Figure 6 Agile MDD Process

Benefits of integrating Agile and MDD [46, 47]:

1) Shorter turnaround time
2) Reduction in errors
3) Code modularization
4) Evolution of legacy systems

Some challenges [46, 47]:

1) Lack of model management
2) Steep learning curve
3) Unautomated testing phase
4) Lack of verification.

5. Conclusion and Review Remarks

With the extensive use of mobile devices and the rising need for mobile apps across numerous industries, this field has experienced fast growth in recent years. As a result, mobile app development has become a crucial area of research and study for software developers, designers, and tech professionals. This paper presented some prominent techniques and methods involved in the process and highlighted each of their strengths and limitations. As per the literature review, cross-platform development seems to be the most suited technique for modern app development. Low code/no code technologies like model-

driven development can also evidently benefit the process' performance largely. For more user-centered design principles, agile methodologies have proved time and again to be the most excellent strategy for any form of software development. After study and analysis, it is concluded that agile – in integration with its types, should be used in concatenation with a hybrid/cross-platform approach to increase the productivity of developing mobile apps and to gain maximum advantage. With the constant emergence of new technologies, it is getting difficult to decide which methods and techniques to follow but with a clear goal, one can decide what methodologies to use without hassle.

6. Future Scope

At this stage, our analysis only considers the broad and overarching factors which contribute to the performance of the process. It is important to note that our examination does not take into account specific applications with their distinctive infrastructural and socioeconomic considerations, leaving ample room for further research in the future. Another additional noteworthy area of exploration pertains to Progressive Web Apps (PWAs) and their dynamic interplay with contemporary mobile applications, particularly considering the integration of newer technologies such as 5G, Internet of Things, and Artificial Intelligence. The intricate relationships between these innovative elements have the potential to significantly influence and shape the development process and research conducted in these areas.

7. Conflict of Interest

There is no conflict of interest in this work.

References

[1] https://www.statista.com/outlook/dmo/app/india

[2] El-Kassas WS, Abdullah BA, Yousef AH, Wahba AM. Taxonomy of cross-platform mobile applications development approaches. Ain Shams Engineering Journal. 2017 Jun 1;8(2):163-90.

[3] Akinola SO, Adelakun-Adeyemo O. Issues in Native Mobile Application Programming. InProceedings of the first Ibadan ACM International Conference on Computer Science Research and Innovations 2015.

[4] Alsaid MA, Ahmed TM, Jan S, Khan FQ, Khattak AU. A Comparative Analysis of Mobile Application Development Approaches: Mobile Application Development Approaches. Proceedings of the Pakistan Academy of Sciences: a. Physical and computational sciences. 2021 Aug 31;58(1):35-45.

[5] Ashishdeep A, Bhatia J, Varma K. Software process models for mobile application development: A review. Computer Science and Electronic Journal. 2016;7(1):150-3.
[6] Kaleel SB, Harishankar S. Applying agile methodology in mobile software engineering: Android application development and its challenges. Computer Science Technical Reports. 2013 Jan;11.
[7] Flora HK, Chande SV. A review and anaysis on mobile application development processes using agile methodlogies. International Journal of Research in Computer Science. 2013 Jul 1;3(4):9.
[8] Kaur A, Kaur K. Suitability of existing software development life cycle (sdlc) in context of mobile application development life cycle (madlc). International Journal of Computer Applications. 2015 Apr;116(19):1-6.
[9] Flora HK, Wang X, Chande SV. An investigation into mobile application development processes: Challenges and best practices. International Journal of Modern Education and Computer Science. 2014;6(6).
[10] Brambilla M, Cabot J, Wimmer M. Model-driven software engineering in practice. Morgan & Claypool Publishers; 2017 Mar 30.
[11] Holzer A, Ondrus J. Mobile app development: Native or web?. InProc. Workshop eBus.(WeB) 2012.
[12] Huynh MQ, Ghimire P, Truong D. Hybrid app approach: could it mark the end of native app domination?. Issues in Informing Science and Information Technology. 2017 Apr 23;14:049-65.
[13] Nawrocki P, Wrona K, Marczak M, Sniezynski B. A comparison of native and cross-platform frameworks for mobile applications. Computer. 2021 Mar 15;54(3):18-27.
[14] Biørn-Hansen A, Majchrzak TA, Grønli TM. Progressive web apps for the unified development of mobile applications. InWeb Information Systems and Technologies: 13th International Conference, WEBIST 2017, Porto, Portugal, April 25–27, 2017, Revised Selected Papers 13 2018 (pp. 64-86). Springer International Publishing.
[15] Umuhoza E, Brambilla M. Model driven development approaches for mobile applications: A survey. InInternational Conference on Mobile Web and Information Systems 2016 Aug 11 (pp. 93-107). Cham: Springer International Publishing.
[16] Shamsujjoha M, Grundy J, Li L, Khalajzadeh H, Lu Q. Developing mobile applications via model driven development: a systematic literature review. Information and Software Technology. 2021 Dec 1;140:106693.
[17] Tufail H, Azam F, Anwar MW, Qasim I. Model-driven development of mobile applications: A systematic literature review. In2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) 2018 Nov 1 (pp. 1165-1171). IEEE.
[18] Heitkötter H, Majchrzak TA, Kuchen H. Cross-platform model-driven development of mobile applications with md2. InProceedings of the 28th Annual ACM Symposium on Applied Computing 2013 Mar 18 (pp. 526-533).
[19] Francese R, Risi M, Scanniello G, Tortora G. Model-driven development for multi-platform mobile applications. InProduct-Focused Software Process Improvement: 16th International Conference, PROFES 2015, Bolzano, Italy, December 2-4, 2015, Proceedings 16 2015 (pp. 61-67). Springer International Publishing.
[20] Despa ML. Comparative study on software development methodologies. Database Systems Journal. 2014 Jul 1;5(3).

[21] Jabangwe R, Edison H, Duc AN. Software engineering process models for mobile app development: A systematic literature review. Journal of Systems and Software. 2018 Nov 1;145:98-111.
[22] Shylesh S. A study of software development life cycle process models. InNational Conference on Reinventing Opportunities in Management, IT, and Social Sciences 2017 Apr (pp. 534-541).
[23] Jobe W. Native Apps vs. Mobile Web Apps. International Journal of Interactive Mobile Technologies. 2013 Oct 1;7(4).
[24] De Andrade PR, Albuquerque AB, Frota OF, Silveira RV, da Silva FA. Cross platform app: a comparative study. arXiv preprint arXiv:1503.03511. 2015 Mar 11.
[25] https://spdload.com/blog/app-development-cost/
[26] Offermann, S., and V. Rios. Adobe Announces Agreement to Acquire Nitobi, Creator of PhoneGap.
[27] Enihe R, Joshua J. Hybrid mobile application development: A better alternative to native. Global Scientific Journal. 2020 May;8(5).
[28] Khandeparkar A, Gupta R, Sindhya B. An introduction to hybrid platform mobile application development. International Journal of Computer Applications. 2015 Jan 1;118(15).
[29] Biørn-Hansen A, Grønli TM, Ghinea G, Alouneh S. An empirical study of cross-platform mobile development in industry. Wireless Communications and Mobile Computing. 2019 Jan 3;2019.
[30] Dagne L. Flutter for cross-platform App and SDK development.
[31] Shah K, Sinha H, Mishra P. Analysis of cross-platform mobile app development tools. In2019 IEEE 5th International Conference for Convergence in Technology (I2CT) 2019 Mar 29 (pp. 1-7). IEEE.
[32] https://powerapps.microsoft.com/en-us/native-vs-cross-platform-apps/
[33] https://www.mendix.com/model-driven-development/
[34] Di Bella E, Sillitti A, Succi G. A multivariate classification of open source developers. Information Sciences. 2013 Feb 1;221:72-83.
[35] https://info.digital.ai/rs/981-LQX-968/images/AR-SA-2022-16th-Annual-State-Of-Agile-Report.pdf
[36] Flora HK. Adopting an agile approach for the development of mobile applications.
[37] Alyani N, Shirzad S. Learning to innovate in distributed mobile application development: Learning episodes from Tehran and London. In2011 Federated Conference on Computer Science and Information Systems (FedCSIS) 2011 Sep 18 (pp. 497-504). IEEE.
[38] Scharff C, Verma R. Scrum to support mobile application development projects in a just-in-time learning context. InProceedings of the 2010 icse workshop on cooperative and human aspects of software engineering 2010 May 2 (pp. 25-31).
[39] Al-Saqqa S, Sawalha S, AbdelNabi H. Agile software development: Methodologies and trends. International Journal of Interactive Mobile Technologies. 2020 Nov 1;14(11).
[40] Ozkan N, Bal S, Erdogan TG, Gök MŞ. Scrum, Kanban or a Mix of Both? A Systematic Literature Review. In2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS) 2022 Sep 4 (pp. 883-893). IEEE.

[41] Rodríguez P, Mäntylä M, Oivo M, Lwakatare LE, Seppänen P, Kuvaja P. Advances in using agile and lean processes for software development. InAdvances in computers 2019 Jan 1 (Vol. 113, pp. 135-224). Elsevier.

[42] Al-Ratrout S, Tarawneh OH, Altarawneh MH, Altarawneh MY. Mobile application development methodologies adopted in Omani market: a comparative study. International Journal of Software Engineering & Applications (IJSEA). 2019 Mar 23;10(2).

[43] Abdelaziz AA, El-Tahir Y, Osman R. Adaptive Software Development for developing safety critical software. In2015 International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering (ICCNEEE) 2015 Sep 7 (pp. 41-46). IEEE.

[44] Tirumala SS, Ali S, Babu A. A hybrid agile model using SCRUM and feature driven development. International Journal of Computer Applications. 2016 Dec;156(5):1-5.

[45] Zafar I, Nazir AK, Abbas M. The impact of agile methodology (DSDM) on software project management. In Circulation in Computer Science: International Conference on Engineering, Computing & Information Technology (ICECIT 2017 Oct) (pp. 1-6).

[46] Alfraihi H, Lano K. The integration of agile development and model driven development-a systematic literature review. InInternational Conference on Model-Driven Engineering and Software Development 2017 Feb 19 (Vol. 2, pp. 451-458). SciTePress.

[47] Kulkarni V, Barat S, Ramteerthkar U. Early experience with agile methodology in a model-driven approach. InModel Driven Engineering Languages and Systems: 14th International Conference, MODELS 2011, Wellington, New Zealand, October 16-21, 2011. Proceedings 14 2011 (pp. 578-590). Springer Berlin Heidelberg.