

The Impact of Kotlin Multiplatform on Modern Android Development

Mohammad Rahim Tahiri¹ , Fardin Ahmadi² 

1. Faculty Member, Department of Software Engineering, Faculty of Computer Science, Kateb University, Kabul, Afghanistan. (Corresponding Author), Email: rahim.taheri@kateb.edu.af
2. Visiting Lecturer, Department of Software Engineering, Faculty of Computer Science, Kateb University, Kabul, Afghanistan.

Article Info

Article type:
Research Article

Article history:

Received: 21/02/2026
Received in revised form: 23/02/2026
Accepted: 12/03/2026
Available online: 19/03/2026

Keywords:

Kotlin Multiplatform, cross-platform development, code sharing, Android, Mobile Application

ABSTRACT

The rapid growth of mobile applications and the need for simultaneous presence on multiple platforms have created challenges in software development. Developing Android and iOS versions separately typically leads to increased costs, development time, and maintenance complexity. Kotlin Multiplatform, as a modern cross-platform approach, enables sharing business logic, data models, and network layers while preserving the native user interface of each platform. This study aims to investigate the impact of Kotlin Multiplatform on optimizing modern Android development in terms of reducing redundant code, increasing development speed, and improving maintainability. The research uses a qualitative, descriptive-analytical method, and data were collected by reviewing technical documents, official documentation, scientific articles, and case studies. Findings indicate that using this technology can share 40-70% of domain and data layer code across platforms. As a result, redundancy is reduced, development speed increases, and maintenance becomes simpler. However, challenges such as initial setup complexity and the need for shared technical skills among teams were also observed. Kotlin Multiplatform can play an effective role in modern Android development as an architecture-oriented approach, provided it is used in suitable projects with proper technical management.

Cite this article: Tahiri, M. & Ahmadi, F. (2026). The Impact of Kotlin Multiplatform on Modern Android Development, *Kateb Scientific-Research Journal of Technology and Engineering*, 1 (1), 179-190.



تأثیر کاتلین چندسکویی بر توسعه مدرن اندروید

محمد رحیم طاهری^۱  فردین احمدی^۲ 

۱. پوهنیار دیپارتمنت انجینیری نرم افزار، پوهنځی کمپیوتر ساینس، پوهنتون کاتب، کابل، افغانستان. (نویسنده مسئول)

ایمیل: rahim.taheri@kateb.edu.af

۲. استاد قراردادی دیپارتمنت انجینیری نرم افزار، پوهنځی کمپیوتر ساینس، پوهنتون کاتب، کابل، افغانستان.

اطلاعات مقاله

نوع مقاله:

مقاله تحقیقی

تاریخ دریافت: ۱۴۰۴/۱۲/۰۲

تاریخ ارزیابی: ۱۴۰۴/۱۲/۰۴

تاریخ پذیرش: ۱۴۰۴/۱۲/۲۱

تاریخ انتشار: ۱۴۰۴/۱۲/۲۸

واژه‌های کلیدی:

کاتلین چندسکویی، توسعه چندسکویی، اشتراک‌گذاری کد، اندروید، توسعه نرم‌افزار موبایل

چکیده

گسترش اپلیکیشن‌های موبایل و نیاز به حضور هم‌زمان در پلتفرم‌های گوناگون، چالش‌هایی در توسعه نرم‌افزار ایجاد کرده است. توسعه جداگانه نسخه‌های اندروید و iOS به‌طور معمول به افزایش هزینه، زمان و پیچیدگی نگهداری می‌انجامد. کاتلین چندسکویی (Kotlin Multiplatform) به‌عنوان رویکردی نوین، امکان اشتراک‌گذاری منطق تجاری، مدل‌های داده و لایه‌های شبکه را فراهم می‌سازد، در حالی که رابط کاربری طبیعی هر پلتفرم حفظ می‌شود. هدف این تحقیق بررسی تأثیر کاتلین چندسکویی بر بهینه‌سازی توسعه مدرن اندروید از دیدگاه کاهش کد تکراری، افزایش سرعت توسعه و بهبود نگهداری است. روش تحقیق کیفی، توصیفی تحلیلی بوده و داده‌ها از راه بررسی اسناد فنی، مستندات رسمی، مقالات علمی و مطالعات موردی گردآوری شده‌اند. یافته‌های تحقیق نشان می‌دهد که استفاده از این فناوری می‌تواند ۴۰ تا ۷۰ درصد از کد لایه دامنه و داده را میان پلتفرم‌ها به اشتراک بگذارد. در نتیجه، افزونگی کاهش یافته، سرعت توسعه افزایش و نگهداری ساده‌تر می‌شود. با این حال، چالش‌هایی مانند پیچیدگی تنظیمات اولیه و نیاز به مهارت مشترک در تیم‌ها نیز وجود دارد. کاتلین چندسکویی می‌تواند به‌عنوان یک رویکرد معماری-محور در توسعه مدرن اندروید نقش مؤثری ایفا کند، مشروط بر آن‌که در پروژه‌های مناسب و با مدیریت فنی صحیح به کار گرفته شود.

استاد: طاهری، محمد رحیم، احمدی، فردین (۱۴۰۴). تأثیر کاتلین چندسکویی بر توسعه مدرن اندروید. مجله علمی -

تحقیقی تکنالوژی و انجینیری کاتب، ۱(۱)، ۱۹۰-۱۷۹.



© نویسندگان.

ناشر: پوهنتون کاتب.

مقدمه

تحول دیجیتال در دهه اخیر باعث افزایش چشم‌گیر استفاده از اپلیکیشن‌های موبایل شده است. سازمان‌ها و شرکت‌های نرم‌افزاری برای حفظ رقابت‌پذیری ناچارند محصولات خود را به‌صورت هم‌زمان برای چندین پلتفرم عرضه کنند. در حوزه موبایل، این امر بیشتر به توسعه هم‌زمان برای اندروید و iOS اشاره دارد. رویکرد سنتی توسعه جداگانه برای هر پلتفرم، از نظر عملکرد طبیعی دارای مزیت است، اما به تکرار پیاده‌سازی منطق تجاری، الگوریتم‌ها و مدیریت داده می‌انجامد [۱]. این افزونگی پیامدهایی همچون افزایش هزینه‌های انسانی، طولانی شدن چرخه توسعه، پیچیدگی هماهنگی نسخه‌ها و دشواری در نگهداری سیستم را به همراه دارد.

در پاسخ به این چالش‌ها، رویکردهای توسعه چندسکویی شکل گرفته‌اند. برخی فریمورک‌ها مانند React Native و Flutter کوشیده‌اند با ارائه یک لایه رابط کاربری مشترک، توسعه را تسریع کنند. اما این رویکردها گاهی در پروژه‌های پیچیده با محدودیت‌هایی در عملکرد بومی و دسترسی به قابلیت‌های ویژه پلتفرم مواجه می‌شوند [۲]. کاتلین چندسکویی رویکرد متفاوتی اتخاذ کرده است. این فناوری به‌جای اشتراک کامل رابط کاربری، تمرکز خود را بر اشتراک منطق برنامه و لایه‌های داده قرار می‌دهد. به این ترتیب، توسعه‌دهندگان می‌توانند تجربه کاربری طبیعی هر پلتفرم را حفظ کنند و درعین‌حال از مزایای استفاده مجدد از کد بهره‌مند شوند. مسأله اصلی این تحقیق آن است که آیا این رویکرد می‌تواند به‌طور واقعی توسعه مدرن اندروید را بهینه سازد یا نه. منظور از توسعه مدرن اندروید، توسعه‌ای است که بر اصول معماری منظم، ماژولار بودن، تست‌پذیری، نگهداری‌پذیری و بهره‌وری تیمی استوار باشد [۳].

۱. اهمیت تحقیق

اهمیت این تحقیق از آن‌جا ناشی می‌شود که توسعه جداگانه اپلیکیشن برای اندروید و iOS به‌طور معمول باعث افزایش هزینه‌های مالی، مصرف زمان بیشتر، تکرار کدها، دشواری نگهداری و ایجاد ناهماهنگی میان نسخه‌های گوناگون می‌گردد. در شرایطی که سرعت عرضه نرم‌افزار و کیفیت محصول نقش اساسی در موفقیت سازمان‌ها دارد، استفاده از روش‌هایی که بتواند بهره‌وری توسعه را افزایش دهد، یک ضرورت جدی به شمار می‌رود. در این میان، کاتلین چندسکویی (Kotlin Multiplatform) به‌عنوان یکی از فناوری‌های نوین، امکان اشتراک‌گذاری منطق تجاری، مدل‌های داده و لایه‌های شبکه را میان چند پلتفرم فراهم می‌سازد، بدون آن‌که رابط کاربری طبیعی هر سیستم‌عامل از بین برود. این ویژگی می‌تواند تعادلی مناسب میان توسعه طبیعی و توسعه چندسکویی ایجاد کند.

۲. مسأله اصلی تحقیق

مسأله اصلی این تحقیق آن است که آیا کاتلین چندسکویی به‌راستی می‌تواند توسعه مدرن اندروید را از نظر کاهش کد تکراری، افزایش سرعت توسعه، بهبود نگهداری نرم‌افزار و هماهنگی تیم‌های توسعه بهینه سازد یا نه. همچنین مشخص نیست که این فناوری در کنار مزایا، با چه چالش‌های فنی و مدیریتی همراه است.

این تحقیق با استفاده از روش کیفی و رویکرد توصیفی تحلیلی انجام شده است. داده‌ها از راه بررسی مقاله‌های علمی، مستندات رسمی، گزارش‌های فنی و مطالعات موردی پروژه‌های واقعی جمع‌آوری و تحلیل گردیده‌اند. بنابراین، هدف این تحقیق بررسی تأثیر کاتلین چندسکویی بر توسعه مدرن اندروید و ارائه دیدگاه علمی درباره مزایا، محدودیت‌ها و کاربرد عملی آن در پروژه‌های نرم‌افزاری است.

۳. پیشینه تحقیق

در سال‌های اخیر، توسعه چندسکویی موبایل به یکی از موضوعات مهم در مهندسی نرم‌افزار تبدیل شده است. تحقیق‌ها و گزارش‌های صنعتی متعددی درباره استفاده از کاتلین چندسکویی انجام شده که هر یک مزایا و محدودیت‌های این فناوری را بررسی کرده‌اند.

جت‌برینز^۱ (2024) در مستندات رسمی خود بیان می‌کند که کاتلین چندسکویی امکان اشتراک‌گذاری منطق تجاری، لایه داده و بخش‌هایی از کد را میان iOS، Android، و Web Desktop فراهم می‌سازد، در حالی که قابلیت حفظ توسعه طبیعی هر پلتفرم نیز باقی می‌ماند [۲]. این رویکرد به سازمان‌ها اجازه می‌دهد بدون قربانی کردن تجربه کاربری طبیعی، از مزایای استفاده مجدد از کد بهره‌مند شوند. در گزارش دیگری، جت‌برینز اعلام کرد که کاتلین چندسکویی به مرحله پایدار (Stable) رسیده و برای استفاده در محیط‌های تولیدی آماده است. این موضوع نشان‌دهنده بلوغ فنی این فناوری و افزایش اعتماد سازمان‌ها به استفاده از آن می‌باشد [۲].

کومار^۲ (2022) در یک مطالعه مقایسه‌ای میان Jetpack Compose، React Native، Flutter و Kotlin Multiplatform نتیجه گرفت که کاتلین چندسکویی در پروژه‌هایی که نیاز به عملکرد طبیعی، معماری ماژولار و اشتراک منطق تجاری دارند، گزینه مناسب‌تری به شمار می‌رود؛ هر چند پیچیدگی راه‌اندازی نخستین آن نسبت به برخی فریمورک‌ها بیشتر است [۴].

آنتونیز^۳ (2022) نیز در بررسی توسعه مدرن اپلیکیشن‌های موبایل اظهار می‌کند که استفاده از کاتلین چندسکویی می‌تواند هزینه توسعه و زمان عرضه محصول به بازار را کاهش دهد؛ زیرا تیم‌ها به جای توسعه دوباره منطق برنامه، از کد مشترک استفاده می‌کنند [۵]. بر پایه گزارش جت‌برینز، بسیاری از شرکت‌های بزرگ مانند Google، Philips، Forbes، و McDonald's از کاتلین چندسکویی در پروژه‌های واقعی استفاده کرده‌اند که این موضوع بیانگر قابلیت صنعتی و مقیاس‌پذیری آن است [۲].

با وجود این، برخی مطالعات و تجربه‌های صنعتی نشان می‌دهد که چالش‌هایی مانند مدیریت وابستگی‌ها، زمان Build، تنظیمات Gradle و نیاز به مهارت مشترک میان تیم‌های iOS و Android همچنان وجود دارد.

در مجموع، پیشینه تحقیق نشان می‌دهد که بیشتر تحقیق‌ها بر مزایای اشتراک کد، کاهش هزینه و حفظ عملکرد طبیعی تأکید دارند؛ اما هنوز نیاز به بررسی عمیق تأثیر این فناوری بر توسعه مدرن اندروید از

1. JetBrains

2. Kumar

3. Antunes

دیدگاه نگهداری پذیری، بهره‌وری تیمی و انسجام معماری وجود دارد. تحقیق حاضر می‌کوشد این خلأ را تا اندازه‌ای پوشش دهد.

۴. روش تحقیق

۴_۱. گونه تحقیق

این تحقیق از گونه کیفی و با رویکرد توصیفی تحلیلی انجام شده است. هدف تحقیق، تحلیل مفهومی و ارزیابی تأثیر یک فناوری بر فرایند توسعه نرم‌افزار است، نه اندازه‌گیری تجربی کمی.

۴_۲. روش جمع‌آوری داده‌ها

داده‌ها از منابع زیر گردآوری شده‌اند:

- مستندات رسمی و راهنماهای توسعه؛
 - مقاله‌های علمی مرتبط با توسعه چندسکوپی؛
 - مطالعات موردی شرکت‌هایی که از KMP استفاده کرده‌اند؛
 - گزارش‌های فنی و تجربه‌های صنعتی منتشر شده.
- داده‌های این تحقیق به صورت کتابخانه‌ای و اسنادی گردآوری شده‌اند. برای افزایش اعتبار علمی تحقیق، از منابع معتبر علمی، مستندات رسمی و گزارش‌های صنعتی مرتبط با کاتلین چندسکوپی استفاده گردید. مهم‌ترین منابع مورد استفاده عبارت‌اند از:
۱. مستندات رسمی جت‌برینز به‌عنوان توسعه‌دهنده اصلی کاتلین چندسکوپی، برای بررسی معماری، قابلیت‌ها، شیوه اشتراک‌گذاری کد و وضعیت پایداری فناوری [۲].
 ۲. مستندات (Google) Android Developers برای تحلیل میزان پشتیبانی اندروید از KMP و نقش آن در توسعه مدرن اندروید [۶].
 ۳. مطالعه کومار (2022) جهت مقایسه کاتلین چندسکوپی با Flutter، React Native و Jetpack Compose از نظر عملکرد، توسعه‌پذیری و معماری [۴].
 ۴. تحقیق آنتونیز (2022) برای بررسی تأثیر KMP بر سرعت توسعه، کاهش هزینه و بهبود فرایند تولید نرم‌افزارهای موبایل [۵].
 ۵. مطالعه گاسپیل^۱ (2024) جهت تحلیل کاربرد KMP در پروژه‌های واقعی و نقش آن در حفظ تجربه کاربری بومی [۷].
 ۶. گزارش اسپرینگر^۲ (2024) برای بررسی میزان اشتراک‌گذاری کد و ارزیابی چارچوب‌های Code Sharing در توسعه موبایل [۸].

1. Guşpiel

2. Springer

۷. نمونه‌های صنعتی منتشرشده توسط شرکت‌هایی مانند McDonald's، Philips، Netflix و Forbes که از کاتلین چندسکویی در پروژه‌های واقعی استفاده کرده‌اند، به‌منظور تحلیل قابلیت عملیاتی این فناوری [۲].

داده‌های گردآوری شده پس از دسته‌بندی موضوعی، بر پایه شاخص‌هایی چون کاهش کد تکراری، سرعت توسعه، نگهداری‌پذیری، هماهنگی تیمی و حفظ تجربه کاربری بومی مورد تحلیل قرار گرفتند.

۳_۴. معیارهای تحلیل

برای تحلیل تأثیر فناوری، معیارهای زیر در نظر گرفته شد:

برای ارزیابی تأثیر کاتلین چندسکویی بر توسعه مدرن اندروید، معیارهای تحلیل بر پایه اصول مهندسی نرم‌افزار و شاخص‌های رایج ارزیابی کیفیت سیستم‌های نرم‌افزاری انتخاب گردید. در این تحقیق، چارچوب مفهومی معیارها با استناد به مدل کیفیت نرم‌افزار ISO/IEC 25010 و همچنین مطالعات مربوط به توسعه چندسکویی تنظیم شده است [۹، ۴].

مدل ISO/IEC 25010 یکی از معتبرترین استانداردهای بین‌المللی برای سنجش کیفیت نرم‌افزار بوده و شاخص‌هایی مانند قابلیت نگهداری، کارایی، سازگاری، قابلیت استفاده مجدد و عملکرد را مورد توجه قرار می‌دهد. بر این پایه، معیارهای زیر در تحقیق حاضر مورد استفاده قرار گرفت:

معیار	توضیح
میزان کاهش کد تکراری	سنجش قابلیت استفاده مجدد از کد و کاهش افزونگی در پروژه‌های چندسکویی
سرعت توسعه	بررسی تأثیر اشتراک‌گذاری منطق برنامه بر کاهش زمان پیاده‌سازی قابلیت‌های مشترک
هماهنگی تیم‌ها	تحلیل میزان همکاری و انسجام میان تیم‌های iOS و Android در استفاده از منبع مشترک کد
سهولت نگهداری	ارزیابی آسانی اصلاح خطاها، توسعه نسخه‌های بعدی و مدیریت تغییرات
کیفیت تجربه کاربری طبیعی	بررسی حفظ طراحی و عملکرد طبیعی رابط کاربری در هر پلتفرم
پیچیدگی معماری	سنجش دشواری راه‌اندازی، ساختار پروژه، مدیریت وابستگی‌ها و نیازمندی‌های فنی

۴_۴. روش تحلیل داده‌ها

داده‌های کیفی به‌صورت تحلیل محتوایی دسته‌بندی شدند. نخست مفاهیم کلیدی استخراج گردید، سپس مزایا و چالش‌ها طبقه‌بندی شده و در پایان با مبانی نظری مهندسی نرم‌افزار مقایسه شدند.

۵. یافته‌های تحقیق

۵_۱. کاهش کد تکراری

تحلیل مطالعات موردی نشان داد که بخش عمده‌ای از منطق تجاری، مدل‌های داده، مدیریت شبکه و اعتبارسنجی‌ها قابل اشتراک‌گذاری هستند. در برخی پروژه‌ها، میان ۴۰ تا ۷۰ درصد از کد لایه دامنه و داده میان پلتفرم‌ها مشترک بوده است [۶]. این امر موجب کاهش افزونگی، یکپارچگی منطق و جلوگیری از ناسازگاری عملکردی گردیده است [۸].

۵_۲. افزایش سرعت توسعه

اشتراک منطق برنامه سبب می‌شود تغییرات در یک نقطه اعمال گردد و نیاز به پیاده‌سازی دوباره کاهش یابد. در پروژه‌های مورد بررسی، زمان توسعه ویژگی‌های مشترک به‌طور محسوسی کاهش یافته است. همچنین هماهنگی میان تیم‌های اندروید و iOS بهبود یافته، زیرا هر دو از یک منبع منطقی مشترک استفاده می‌کنند [۵].

۵_۳. بهبود نگهداری

نگهداری سیستم‌هایی که دارای منطق مشترک هستند ساده‌تر است، زیرا خطاها در یک نقطه اصلاح می‌شوند. همچنین تست‌پذیری افزایش می‌یابد، زیرا لایه مشترک می‌تواند به‌صورت مستقل تست شود [۴].

۵_۴. حفظ تجربه کاربری بومی

برخلاف برخی فریمورک‌های UI-محور، در این رویکرد رابط کاربری به‌صورت بومی توسعه می‌یابد. بنابراین، عملکرد و طراحی منطبق با استانداردهای هر پلتفرم حفظ می‌شود [۷].

۶. چالش‌های تحقیق

با وجود مزایای متعدد کاتلین چندسکویی، تحلیل منابع علمی، مستندات رسمی و مطالعات موردی نشان می‌دهد که استفاده از این فناوری با برخی چالش‌های فنی و مدیریتی نیز همراه است. مهم‌ترین چالش‌های شناسایی شده عبارت‌اند از:

۶_۱. پیچیدگی تنظیم نخستین پروژه

یکی از مهم‌ترین موانع در آغاز استفاده از کاتلین چندسکویی، تنظیم ساختار نخستین پروژه، پیکربندی Gradle، تعریف ماژول‌های مشترک و اتصال آن با پروژه‌های Android و iOS می‌باشد. بسیاری از توسعه‌دهندگان گزارش کرده‌اند که راه‌اندازی نخستین نسبت به پروژه‌های بومی پیچیده‌تر است [۲]. این موضوع به‌ویژه برای تیم‌هایی که تجربه قبلی در معماری چندماژوله ندارند، زمان‌بر می‌باشد.

۲-۶. نیاز به دانش معماری ماژولار

استفاده موفق از کاتلین چندسکوپی مستلزم طراحی درست معماری نرم افزار است. در صورتی که منطق برنامه، لایه داده و وابستگی‌ها به صورت منظم تفکیک نشده باشند، اشتراک گذاری کد دشوار خواهد شد. بنابراین، تیم توسعه نیازمند آشنایی با مفاهیمی چون Clean Architecture، Separation of Concerns، Modularization می‌باشد [۴].

۳-۶. مدیریت وابستگی‌های چندسکوپی

در برخی پروژه‌ها، همه کتابخانه‌ها از کاتلین چندسکوپی پشتیبانی کامل ندارند. این مسئله سبب می‌شود توسعه‌دهندگان برای برخی قابلیت‌ها از پیاده‌سازی اختصاصی هر پلتفرم استفاده کنند. همچنین هماهنگی نسخه کتابخانه‌ها میان Android و iOS گاهی موجب افزایش پیچیدگی پروژه می‌شود [۸].

۴-۶. منحنی یادگیری برای تیم‌ها

پیاده‌سازی KMP نیازمند همکاری مشترک میان توسعه‌دهندگان Android و iOS است. از آنجا که بسیاری از اعضای تیم ممکن است تنها در یک پلتفرم تخصص داشته باشند، یادگیری مفاهیم مشترک، زبان Kotlin، ساختار پروژه و نحوه ارتباط میان پلتفرم‌ها نیازمند زمان و آموزش می‌باشد [۵].

۵-۶. جمع‌بندی چالش‌ها

بر پایه تحلیل انجام شده، چالش‌های یاد شده بیشتر در مراحل نخستین پذیرش فناوری مشاهده می‌شوند و با افزایش تجربه تیم، استفاده از ابزارهای مناسب و طراحی معماری درست تا اندازه زیادی قابل مدیریت هستند. بنابراین، این چالش‌ها مانع مطلق استفاده از کاتلین چندسکوپی نیستند، بلکه عامل‌های قابل کنترل در مسیر پیاده‌سازی به شمار می‌روند.

۷. تحلیل مقایسه‌ای شاخص‌های توسعه میان پروژه‌های سنتی و پروژه‌های مبتنی بر KMP

به منظور ارزیابی عملی تأثیر کاتلین چندسکوپی، داده‌های منتشر شده از ۶ مطالعه موردی و گزارش صنعتی شامل [2] JetBrains (2024)، [4] Kumar (2022)، [5] Antunes (2022)، Springer، [8] (2024)، [7] Guśpiel (2024) و [6] Android Developer Reports (2024) مورد بررسی قرار گرفت. شاخص‌های مشترک استخراج شده شامل میزان کد تکراری، زمان توسعه و ویژگی جدید، نرخ خطا و زمان رفع خطا بودند. سپس مقادیر گزارش شده نرمال سازی و میانگین گیری گردید. نتایج مقایسه در جدول شماره (۱) ارائه شده است:

جدول ۱- میانگین شاخص‌ها

شاخص	پروژه جداگانه	پروژه KMP	درصد بهبود
کد تکراری	۶۲٪	۳۲٪	۴۸٪ کاهش
زمان توسعه ویژگی (روز)	۱۴.۲	۱۰.۱	۲۹٪ کاهش
نرخ خطا	۱۸٪	۱۳.۸٪	۲۳٪ کاهش
زمان رفع خطا (ساعت)	۹.۲	۶	۳۵٪ کاهش

منبع: محاسبه تحقیقی بر پایه داده‌های استخراج شده از: Antunes, Kumar (2022), JetBrains (2024), Guśpiel (2024), Springer (2024), (2022)

مطابق جدول (۱)، پروژه‌های مبتنی بر کاتلین چندسکویی در همه شاخص‌ها عملکرد بهتری نسبت به توسعه جداگانه داشته‌اند. بیشترین بهبود مربوط به کاهش کد تکراری (۴۸٪) بوده که نشان‌دهنده مزیت اصلی اشتراک‌گذاری منطق برنامه است. همچنین کاهش زمان توسعه ویژگی‌ها (۲۹٪) نشان می‌دهد که استفاده مجدد از کد می‌تواند سرعت عرضه محصول را افزایش دهد. در بخش کیفیت، نرخ خطا ۲۳٪ کاهش یافته و زمان رفع خطا نیز ۳۵٪ کمتر شده است؛ زیرا خطاهای موجود در لایه مشترک تنها یک بار اصلاح می‌گردند. این یافته‌ها بیانگر آن است که KMP افزون بر بهره‌وری، در نگهداری نرم‌افزار نیز تأثیر مثبت دارد.

۸. بحث و بررسی

یافته‌های تحقیق حاضر نشان داد که کاتلین چندسکویی می‌تواند در پروژه‌های متوسط و بزرگ، تأثیر مثبت بر توسعه مدرن اندروید داشته باشد. این نتیجه با یافته‌های [2] Antunes, Kumar (2022) [4] و [5] (2022) همسو است؛ زیرا آن‌ها نیز اشتراک‌گذاری منطق برنامه و کاهش هزینه توسعه را از این مزایای اصلی این فناوری دانسته‌اند. با این حال، تفاوت تحقیق حاضر با مطالعات پیشین در آن است که این تحقیق، KMP را تنها یک ابزار اشتراک‌کد ندانسته، بلکه آن را از دیدگاه معماری نرم‌افزار، نگهداری‌پذیری و هماهنگی تیمی نیز بررسی کرده است. بسیاری از تحقیق‌های قبلی بیشتر بر سرعت توسعه یا مقایسه با React و Flutter Native تمرکز داشته‌اند؛ در حالی که تحقیق حاضر نقش KMP را در توسعه مدرن اندروید تحلیل کرده است.

۱_۸ چرا کاهش افزونگی به وجود می‌آید؟

کاهش افزونگی زمانی رخ می‌دهد که منطق تجاری، مدل‌های داده، اعتبارسنجی و لایه شبکه تنها یک بار نوشته شده و در Android و iOS به صورت مشترک استفاده گردد. در روش سنتی، این بخش‌ها برای هر پلتفرم جداگانه توسعه می‌یابد؛ اما در KMP یک منبع مشترک وجود دارد. به همین دلیل، تکرار کد کاهش یافته و اصل (Don't Repeat Yourself) (DRY) تحقق می‌یابد.

۲_۸ چرا سرعت توسعه افزایش می‌یابد؟

زمانی که قابلیت جدیدی اضافه می‌شود، در پروژه‌های سنتی باید دو بار پیاده‌سازی گردد؛ اما در KMP بخش مشترک یک بار توسعه می‌شود. بنابراین زمان تولید ویژگی‌های جدید، تست منطق برنامه و هماهنگی میان تیم‌ها کاهش می‌یابد. یافته‌های جدول مقایسه‌ای نیز این موضوع را تأیید کرد که زمان توسعه ویژگی حدود ۲۹٪ کاهش یافته است.

۳_۸ چرا نگهداری پذیری بهتر می‌شود؟

در معماری مشترک، خطاهای موجود در منطق برنامه تنها در یک نقطه اصلاح می‌شوند. همچنین تست واحد (Unit Test) برای لایه مشترک ساده‌تر انجام می‌شود. این مسأله سبب کاهش هزینه نگهداری و تسریع در رفع خطاها می‌گردد. یافته‌های تحقیق حاضر نیز کاهش ۳۵٪ در زمان رفع خطا را نشان داد.

۴_۸ کیفیت تجربه کاربری چگونه حفظ می‌شود؟

برخلاف Flutter یا React Native که به طور معمول لایه رابط کاربری مشترک دارند، در KMP رابط کاربری هر پلتفرم به صورت بومی توسعه می‌یابد. بنابراین استانداردهای طراحی Android و iOS حفظ شده و کارایی سیستم کاهش نمی‌یابد. این ویژگی یکی از تفاوت‌های اساسی KMP با برخی فناوری‌های دیگر است.

۵_۸ چرا در برخی پروژه‌ها مناسب نیست؟

یافته‌ها نشان داد که در پروژه‌های کوچک یا تیم‌های کم تجربه، پیچیدگی راه‌اندازی نخستین، ساختار Gradle و نیاز به دانش معماری ممکن است مزایای KMP را کاهش دهد. بنابراین، استفاده از آن بیشتر برای پروژه‌های متوسط، بزرگ و بلندمدت توصیه می‌شود.

۹. جمع‌بندی و نتیجه‌گیری

بر پایه تحلیل انجام شده، کاتلین چندسکویی را می‌توان یک رویکرد مؤثر برای توسعه مدرن اندروید دانست که قادر است از راه اشتراک‌گذاری منطق برنامه، میزان کد تکراری را کاهش دهد، سرعت توسعه را افزایش دهد و نگهداری نرم‌افزار را ساده‌تر سازد.

نتایج تحقیق حاضر با مطالعات قبلی همسو است، اما نشان می‌دهد که ارزش اصلی KMP تنها در استفاده مجدد از این کد نیست، بلکه در ایجاد انسجام معماری، هماهنگی تیمی و حفظ تجربه کاربری بومی نیز می‌باشد.

به‌طور کلی نتیجه گرفته می‌شود که:

- KMP برای پروژه‌های متوسط و بزرگ بیشترین کارایی را دارد.
 - کاهش افزونگی نتیجه مستقیم اشتراک منطق تجاری و داده‌ها است.
 - افزایش سرعت توسعه ناشی از این حذف پیاده‌سازی تکراری می‌باشد.
 - نگهداری بهتر نتیجه تمرکز اصلاحات در یک منبع مشترک است.
 - استفاده موفق از این فناوری وابسته به مهارت تیم و معماری مناسب پروژه می‌باشد.
- بنابراین، کاتلین چندسکویی جایگزین کامل توسعه بومی نیست، بلکه یک راهکار مکمل و معماری‌محور برای توسعه چندسکویی مدرن به شمار می‌رود.

منابع و مرجعها

1. Kotlin Foundation. (2023). Kotlin Foundation. kotlin.org.
2. JetBrains. (2024). Multiplatform Documentation. JetBrains Official Website.
3. Oechsle, M. (2024). Cross-Platform Mobile Development.
4. Kumar. (2022). Evaluating Modern Android Frameworks: A Comparative Study of Flutter, Kotlin Multiplatform, Jetpack Compose, and React Native.
5. Antunes. (2022). Leveraging Kotlin Multiplatform for Modern Mobile App Development.
6. Google Developer. (2024). Android Support for Kotlin Multiplatform (KMP) to Share Business Logic Across Mobile, Web, Server, and Desktop.
7. Guśpiel. (2024). Mobile App Development Using Kotlin Multiplatform.
8. Springer. (2024). Evaluating Code-Sharing Frameworks in Mobile Development.
9. ISO. (2011). ISO/IEC 25010: Systems and Software Quality Models.