

# ПОРІВНЯЛЬНИЙ АНАЛІЗ НАТИВНОЇ ТА КРОСПЛАТФОРМЕННОЇ РОЗРОБКИ

Вінницький національний технічний університет

## *Анотація*

*У роботі проаналізовано переваги та недоліки кроссплатформенної та нативної розробки мобільних застосунків, а також визначено умови доцільності використання певного способу розробки та наведено аргументи щодо вибору мови новачком.*

**Ключові слова:** нативна розробка, кроссплатформенна розробка, Kotlin, Flutter, React Native, iOS.

## *Abstract*

*This paper analyzes the advantages and disadvantages of cross-platform and native mobile application development, identifies the conditions for the feasibility of using a particular development method, and provides arguments for choosing a language for a beginner.*

**Keywords:** native development, cross-platform development, Kotlin, Flutter, React Native, iOS.

## Вступ

В останні роки розробка мобільних застосунків стала невід'ємною складовою сучасного технологічного світу. Одним із ключових питань, що виникає при створенні мобільного застосунку, є вибір підходу до розробки – кроссплатформенна чи нативна. Кроссплатформенна розробка дозволяє створювати застосунки, які можуть працювати на різних платформах, таких як iOS та Android, використовуючи спільний код. З іншого боку, нативна розробка передбачає створення окремих застосунків для кожної платформи, використовуючи мови програмування та інструменти, специфічні для кожної платформи. Таким чином, метою роботи є аналіз переваг та недоліків кроссплатформенної та нативної мобільної розробки для визначення випадків, в яких доцільно використати той чи інший спосіб розробки.

## Результати дослідження

Кроссплатформенність у розробці мобільних застосунків стає все більш поширеним вибором для багатьох ІТ-компаній, особливо тих, де працює невелика кількість співробітників. Цей підхід зазвичай використовують для невеликих і середніх проектів, де термін розробки коливається від декількох місяців до року. Кроссплатформенна розробка забезпечує можливість створювати застосунки, які працюють на різних платформах, з використанням загального коду.

Однією з основних переваг кроссплатформенності є зниження витрат для клієнта та компанії. Завдяки меншій кількості розробників потрібно менше ресурсів, що призводить до зниження загальних витрат на проект [1]. Крім того, спілкуватися з однією людиною, а не з великою командою розробників, замовник вважає зручнішим, оскільки це спрощує процес комунікації та забезпечує однотайність розуміння вимог щодо функціональності та дизайну застосунку, адже навіть в рамках двох розробників потрібно узгоджувати бачення проекту.

Однак для забезпечення інтуїтивно зрозумілого інтерфейсу як для користувачів iOS, так і для користувачів Android, нативна розробка є найефективнішим варіантом. Використання рекомендацій інтерфейсу, специфічних для кожної платформи, дозволяє уникнути незручностей для користувачів і заощадити час на покращення їх досвіду.

У великих проектах, де пріоритетом є продуктивність, якість та надійність, нативна розробка виявляється найкращим вибором. Розробка для Android та iOS вимагає розробників з глибокими знаннями і досвідом відповідних мов програмування. Хоча це призводить до збільшення витрат на проект, у випадку великих проектів такі витрати зазвичай виправдовуються.

Нативна розробка також забезпечує оптимальну швидкість роботи програми за рахунок використання компільованого коду і повної апаратної підтримки пристрою. Оновлення мови програмування та апаратного функціоналу негайно стають доступними для розробників, без необхідності чекати на адаптацію у кросплатформних технологіях.

Однією з ключових переваг нативної розробки є відсутність обмежень, що дозволяє розробляти великі проекти. Нативні застосунки можуть повністю використовувати функціонал системного програмного забезпечення, такий як камера, мікрофон, геолокація, що сприяє гнучкості та ефективності реалізації функцій застосунку, а також ефективному використанню ресурсів пристрою, таких як пам'ять та батарея [2].

Простота навчання також є важливим фактором при виборі технології. Популярні технології, як от нативна розробка з використанням Kotlin для платформи Android, мають багато курсів, уроків і статей, що спрощує їх вивчення. Популярність Kotlin в Google сприяє зростанню кількості навчальних матеріалів і бібліотек, а також зростає зацікавленість інших компаній у використанні цієї технології [3].

З іншого боку, Flutter, інша технологія, що розроблена Google для крос-платформної розробки, також має потенціал. Flutter використовує мову Dart і має привабливі переваги, такі як малювання вікон на екрані смартфона і їх самостійний рендеринг та більш простий вихідний код. Однак, у порівнянні з React Native та Xamarin, Flutter є молодшою технологією, і кількість навчальних матеріалів та бібліотек для неї ще обмежена. Це може ускладнити процес навчання та розробки в цій технології, особливо для початківців.

Крос-платформність також може знизити поріг знань для входження в технологію. Наприклад, React Native, який базується на React JS, дозволяє веб-розробникам легко перейти до розробки мобільних застосунків. Крім того, платформа Expo надає інструменти, які полегшують роботу з React Native і знижують поріг входу для новачків. Також слід відзначити хорошу документацію від Facebook, яка сприяє переходу від веб-розробки до мобільної.

У підсумку, швидке зростання популярності Kotlin, завдяки активній підтримці та популяризації від Google робить його привабливим для початківців у мобільній розробці. З іншого боку, хоча Flutter має перспективи, недостатня кількість навчальних матеріалів та бібліотек може ускладнити його вивчення і використання для новачків. Крос-платформність, як у React Native, може знизити поріг знань для входження в технологію, особливо для веб-розробників, завдяки схожості з React JS і платформі Expo, що спрощує розробку та тестування застосунків.

## Висновки

На основі результатів аналізу та порівняння різних технологій розробки, можна зробити такі висновки щодо використання кросплатформної та нативної розробки.

Кросплатформна розробка підходить краще у випадках:

1. Коли необхідно розробити застосунок для кількох платформ, наприклад, якщо потрібно мати версії для Android та iOS.
2. Якщо команда розробки складається з обмеженого числа розробників, і необхідно максимізувати використання їхнього часу та ресурсів, уникнувши дублювання коду для кожної платформи.
3. Якщо потрібно швидко випустити продукт на ринок, зменшивши час розробки та тестування шляхом використання загального коду.
4. Якщо розробляється простий застосунок з основним функціоналом, який не вимагає великих обчислювальних потужностей або глибокої інтеграції з операційною системою.

Нативна розробка підходить краще у випадках:

1. Коли потрібні передові можливості та повна інтеграція з операційною системою, яка забезпечує максимальну продуктивність та швидкість роботи застосунку.
2. Якщо розробляється застосунок з високою складністю та специфічним функціоналом, який вимагає доступу до глибоких рівнів операційної системи або спеціалізованих функцій пристрою.
3. Коли в наявності є достатньо ресурсів (людських та фінансових) для розробки та підтримки окремих версій застосунку для кожної платформи.
4. Якщо мобільною розробкою починає займатися новачок.

Важливо враховувати, що кожен проект має свої унікальні вимоги та обставини, тому рішення щодо використання кроссплатформної або нативної розробки повинно базуватись на детальному аналізі потреб проекту, ресурсів команди та очікувань щодо продуктивності та швидкості роботи застосунку.

#### СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Native and Cross-Platform Development. URL: <https://kotlinlang.org/docs/native-and-cross-platform/>
2. Native vs Hybrid vs Cross-Platform: A Comprehensive Comparison. URL: <https://www.netsolutions.com/insights/native-vs-hybrid-vs-cross-platform>
3. Language Rating 2023. URL: <https://dou.ua/lenta/articles/language-rating-2023/>

**Цимбал Максим Юрійович** — студент групи 5ПІ-216, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: [iezeikiile98@gmail.com](mailto:iezeikiile98@gmail.com)

**Романюк Оксана Володимирівна** – к.т.н., доцент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: [romaniukoksanav@gmail.com](mailto:romaniukoksanav@gmail.com)

**Tsymbal Maxym Y.** — Department of Information Technology and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: [iezeikiile98@gmail.com](mailto:iezeikiile98@gmail.com)

**Romaniuk Oksana V.** – Candidate of Technical Sciences, Associate Professor of the Software Chair, Vinnytsia National Technical University, Vinnytsia, e-mail: [romaniukoksanav@gmail.com](mailto:romaniukoksanav@gmail.com)