ВИКОРИСТАННЯ FLUTTER У ФОРМУВАННІ КОМПЕТЕНТНОСТЕЙ З РОЗРОБКИ МОБІЛЬНИХ ЗАСТОСУНКІВ

USING THE FLUTTER FRAMEWORK TO DEVELOP MOBILE APPLICATION DEVELOPMENT SKILLS

УДК [378.091.212:004] :004.42
DOI https://doi.org/10.32782/ip/87.58

Шиян І.О.,
аспірант кафедри інформатики та кібернетики
Мелітопольського державного педагогічного університету імені Богдана Хмельницького (Запоріжжя, Україна)

Стаття присвячена дослідженню дидактичного потенціалу кросплатформеного фреймворку Flutter як інноваційного інструменту для формування комплексу професійних компетентностей у майбутніх розробників мобільних застосунків. В умовах стрімкого розвитку IT-ринку та зростаючих вимог до фахівців, традиційний підхід до навчання, що передбачає паралельне вивчення нативних технологій для Android (Kotlin/Java) та iOS (Swift), демонструє свою ресурсозатратність та низьку ефективність. Такий підхід не лише збільшує когнітивне навантаження на студентів, але й створює фінансові бар'єри, зокрема через необхідність використання дороговартісної техніки Apple для розробки під iOS. У статті розглядається перехід до єдиної кодової бази на основі Flutter як ефективна освітня та виробнича стратегія, що дозволяє оптимізувати навчальний процес та готувати більш універсальних спеціалістів.

Особлива увага приділяється глибокому аналізу унікальних архітектурних переваг Flutter, що вигідно вирізняють його серед інших кросплатформених рішень. Детально розглядається роль власного графічного двигуна Skia, який забезпечує повний контроль над кожним пікселем та гарантує подібність користувацького інтерфейсу на всіх платформах. Аналізується декларативний підхід до побудови UI, що формує у студентів сучасний спосіб мислення про інтерфейс як функцію від стану, та переваги мови програмування Dart, зокрема її строга типізація та null-safety, що сприяють виробленню навичок написання чистого та безпечного коду. Досліджується, як ключовий інструмент Flutter – Hot Reload – кардинально трансформує освітній процес, дозволяючи студентам миттєво візуалізувати результати своєї роботи, що прискорює експерименти, налагодження та поглиблює практичне розуміння принципів UI/UX.

Таким чином, Flutter не лише значно спрощує поріг входу у мобільну розробку, а й цілеспрямовано формує сучасне архітектурне мислення, зорієнтоване на кросплатформеність, реактивне програмування та ефективне управління станом(state management). Робиться висновок, що інтеграція Flutter в освітній процес є стратегічно виправданим кроком. Це дозволяє готувати гнучких та конкурентоспроможних фахівців, чиї навички є релевантними для широкого спектра сучасних платформ, включаючи веб, десктопні та вбудовані системи, що забезпечує їхню затребуваність на ринку праці в довгостроковій перспективі.

Ключові слова: *навчання мобільної розробки, освітній процес, Flutter, мобільні операційні системи, освітні технології.*

This article is devoted to researching the didactic potential of the cross-platform Flutter framework as an innovative tool for developing a set of professional competencies in future mobile application developers. In the context of the rapid development of the IT market and growing demands on specialists, the traditional approach to training, which involves the parallel study of native stacks for Android (Kotlin/Java) and iOS (Swift), is proving to be resource-intensive and inefficient. This approach not only increases the cognitive load on students, but also creates financial barriers, in particular due to the need to use expensive Apple equipment for iOS development. The article discusses the transition to a single code base based on Flutter as an effective educational and production strategy that allows optimizing the learning process and training more versatile specialists.

Particular attention is paid to an in-depth analysis of Flutter's unique architectural advantages, which set it apart from other cross-platform solutions. The role of Skia's proprietary graphics engine, which provides complete control over every pixel and ensures user interface consistency across all platforms, is examined in detail. The declarative approach to UI construction is analyzed, which shapes students' modern way of thinking about the interface as a function of state, and the advantages of the Dart programming language, in particular its strict typing and null-safety, which contribute to the development of skills in writing clean and safe code.The article explores how Flutter's key tool, Hot Reload, radically transforms the educational process, allowing students to instantly visualize the results of their work, which speeds up experimentation and debugging and deepens their practical understanding of UI/UX principles.

Thus, Flutter not only significantly lowers the barrier to entry into mobile development, but also purposefully shapes modern architectural thinking focused on cross-platform compatibility, reactive programming, and effective state management. The conclusion is that integrating Flutter into the educational process is a strategically justified step. It allows for the training of flexible and competitive specialists whose skills are relevant to a wide range of modern platforms, including web, desktop, and embedded systems, ensuring their long-term demand in the labor market.

Key words: *mobile development education, learning process, Flutter, mobile operating systems, educational technologies.*

Постановка проблеми. У сучасному технологічному цифровому світі мобільні застосунки є ключовим інструментом для освіти, бізнесу та повсякденного життя. Цим підтверджується існування стабільно високого попита на кваліфікованих розробників мобільних застосунків. Традиційний підхід до розробки, що передбачає створення окремих версій застосунку для iOS (з використанням Swift/Objective-C) та Android (Kotlin/Java), є ресурсозатратним, вимагає декількох команд розробників та збільшує час виходу продукту на ринок.

У відповідь на ці виклики все більшої популярності набувають кросплатформенні технології, які дозволяють створювати застосунки для обох платформ на основі єдиної кодової бази. Це підтверджує **аналіз останніх досліджень і публікацій.** У дослідженні Зарічук О. (2023) [8] порівнюються нативні та кросплатформені технології і виділяються очевидні переваги других на першими у швидкості та вартості розробки. У дослідженні Ічанської Н.В. та Улько С. І. (2020) [5] розглядаються сильні і слабкі сторони популярних кросплатформених фреймворків: Apache Cordova, React Native, Xamarin та ін. Серед таких рішень виділяється фреймворк Flutter від Google, який демонструє стрімке зростання завдяки гнучкості у створенні користувацьких інтерфейсів високій продуктивності, та широким перспективам розвитку (підтримка вебу, десктопних та вбудованих систем).

У зв'язку з цим виникає актуальна науково-педагогічна та методологічна проблема: яким чином інтеграція Flutter в освітній процес впливає на формування ключових компетентностей майбутніх розробників мобільних застосунків? Необхідно дослідити, чи дозволяє «Flutter-first» підхід не тільки прискорити процес навчання, але й сформувати у студентів сучасне архітектурне мислення, орієнтоване на кросплатформеність, та забезпечити їхню конкурентоспроможність на ринку праці в довгостроковій перспективі.

Мета полягає в тому, щоб проаналізувати технологічні переваги кросплатформної мобільної розробки над класичною нативною та надати методичні рекомендації щодо використання кросплатформеного фреймворку Flutter як ефективного інструменту для формування ключових професійних компетентностей у майбутніх розробників мобільних застосунків.

Виклад основного матеріалу дослідження. Для розуміння дидактичного потенціалу Flutter необхідно визначити його місце в системі сучасних інструментів мобільної розробки, порівнявши його з нативними та іншими кросплатформеними підходами.

1. Аналіз сучасних підходів до розробки мобільних застосунків

1.1. Нативна розробка (iOS, Android)

Нативна розробка передбачає створення програмного продукту з використанням мов програмування та інструментарію (SDK), які є створеними для конкретної операційної системи[8]. Для платформи iOS – це мови Swift або Objective-C та середовище Xcode, а для Android – Kotlin або Java та середовище Android Studio.

У нативного підходу до розробки присутні свої переваги:

– Максимальна продуктивність. Прямий доступ до апаратних ресурсів пристрою без проміжних шарів абстракції забезпечує найвищу швидкість роботи та відгуку інтерфейсу.

– Повноцінний доступ до API ОС. Розробники мають доступ до всіх нових функцій та можливостей операційної системи одразу після їх виходу

– Уніфікований користувацький досвід (UI/UX). Застосунки виглядають і функціонують саме так, як очікують користувачі операційної системи. Фреймворк нативної розробки надає доступ до стандартних елементів графічного інтерфейсу для їх швидкої імплементації.

Попри всі очевидні переваги, такий підхід має значну кількість недоліків:

– Висока вартість розробки та часові затрати. Необхідність підтримувати дві окремі кодові бази та, часто, дві окремі команди розробників. Це значно збільшує бюджет та терміни реалізації проєкту.

– Складність синхронізації. Впровадження нових функцій потребує подвійних зусиль та ускладнює їх одночасний реліз на обох платформах.

1.2. Огляд та порівняння кросплатформених фреймворків

Кросплатформені фреймворки були створені для вирішення проблеми розробки під кілька мобільних платформ. Окрім вищеназваного Flutter, ще можна виділити React Native та .NET MAUI (раніше Xamarin).

React Native – фреймворк, розроблений компанією Meta, для створення мобільних застосунків за допомогою JavaScript та React. Його ключова особливість полягає у використанні «мосту» (bridge), який транслює JavaScript-код у нативні UI-компоненти. Це забезпечує нативний вигляд застосунку, але може призводити до втрат продуктивності застосунку при побудові складних графічних інтерфейсів [2].

.NET MAUI – рішення від Microsoft, що дозволяє розробникам на C# та .NET створювати застосунки для iOS, Android, Windows та macOS. Як і React Native, він трансформує спільний код у нативні інтерфейси відповідних платформ. .NET MAUI є нащадком Xamarin.Forms, який тепер можна використовувати для розробки не тільки під iOS та Android, а і macOS і Windows. Цей фреймворк надає можливість реалізувати логіку програми та макет інтерфейсу користувача в одній базі коду[7].

Flutter – це фреймворк для розробки програмного забезпечення з відкритим кодом, який був створений компанією Google. Він може використовуватися для розробки кросплатформених застосунків на основі спільної кодової бази для веб-сайтів, Android, iOS, Linux, macOS та Windows. Вперше представлений у 2015 році, Flutter був випущений у травні 2017 року. Flutter займає унікальну позицію у світі кросплатформених інструментів. Він не використовує нативні компоненти і не покладається на «міст». Натомість Flutter використовує власний високопродуктивний графічний двигун Skia для рендерингу UI на екрані. Це дозволяє досягти повної уніфікації інтерфейсу та високої продуктивності на всіх платформах.

1.3. Flutter як стратегічний вибір

Вибір Flutter в освітньому процесі обґрунтовується низкою його фундаментальних переваг, які безпосередньо впливають на формування професійних навичок.

Архітектура: віджети та двигун Skia. Основний принцип Flutter – «все є віджет». Інтерфейс будується з ієрархії віджетів (кнопки, текстові поля, відступи, анімації), що реалізує декларативний підхід до UI. Студенти вчаться мислити не послідовністю імперативних команд, а описом кінцевого стану інтерфейсу. Завдяки двигуну Skia, який використовується в Google Chrome та Android, інтерфейс виглядає однаково на будь-якому пристрої, а продуктивність анімацій сягає 60–120 кадрів на секунду [6].

Мова програмування Dart. Flutter використовує мову Dart, яка також була розроблена Google. Вона має кілька переваг у освітніх цілях: типізація та null-safety допомагають уникнути поширених помилок на етапі написання коду; синтаксис є зрозумілим для тих, хто знайомий з Java, C# або JavaScript. Ключовою особливістю Dart є підтримка двох типів компіляції: JIT (Just-In-Time) для процесу розробки та AOT (Ahead-Of-Time) для скомпільованих версій застосунків [3].

Ключові переваги для навчання та розробки:

1. Єдина кодова база: Студенти фокусуються на вивченні однієї мови та одного фреймворку для створення повноцінних застосунків для обох головних платформ.

2. Висока продуктивність: AOT-компіляція в нативний код забезпечує швидкість роботи.

3. Виразний та гнучкий UI: Контроль над кожним пікселем дозволяє створювати унікальні та брендовані інтерфейси без обмежень нативних платформ.

4. Швидкий цикл розробки (Hot Reload): Завдяки JIT-компіляції, зміни в коді відображаються на емуляторі чи реальному пристрої за долі секунди без перезапуску програми та втрати її поточного стану. Ця функція кардинально прискорює процес навчання, дозволяючи миттєво бачити результат, експериментувати та виправляти помилки [4].

2. Компетентності сучасного розробника мобільних застосунків

Для аналізу впливу Flutter на освітній процес необхідно визначити цільову модель компетентностей, тобто набір знань та умінь, що вимагаються від сучасного мобільного розробника. Їх можна умовно поділити на технічні(hard skills) та «м'які» (soft skills) компетентності.

2.1. Технічні компетентності

Це фундаментальні навички, що безпосередньо пов'язані з процесом розробки програмного забезпечення.

1. Проєктування UI/UX. Не лише створення інтерфейсу, а й розуміння принципів побудови зручних, інтуїтивних та естетично привабливих користувацьких досвідів.

2. Управління станом застосунка. Вміння організовувати потік даних, керувати станом інтерфейсу та бізнес-логіки з використанням архітектурних патернів(BLoC, Provider, Riverpod).

3. Робота з мережевими запитами(API). Взаємодія з сервером для отримання, відправлення та обробки даних, як правило, у форматі JSON через REST API.

4. Зберігання та обробка даних. Робота з локальними базами даних (напр., SQLite, Hive) для кешування та офлайн-доступу, а також інтеграція з хмарними сервісами (напр., Firebase).

5. Тестування та налагодження коду. Вміння писати юніт-тести для логіки, віджет-тести для UI-компонентів та інтеграційні тести для перевірки роботи всього застосунку.

6. Публікація застосунків. Розуміння процесів підготовки та завантаження застосунків у магазини App Store та Google Play.

2.2. Soft skills та архітектурні компетентності

Ці навички визначають здатність розробника мислити системно, працювати в команді та постійно розвиватися.

1. Алгоритмічне та архітектурне мислення. Здатність проєктувати масштабовану, гнучку та підтримувану архітектуру застосунку.

2. Навички вирішення проблем (Problem-solving). Вміння аналізувати складні завдання, декомпозувати їх та знаходити оптимальні технічні рішення.

3. Здатність до самостійного навчання. IT-індустрія стрімко розвивається, тому вміння швидко опановувати нові технології та підходи до розробки є критично важливим.

4. Розуміння принципів кросплатформної логіки. Розробник має писати код, який буде ефективно працювати та виглядати доречно на різних платформах, враховуючи їхні особливості.

3. Роль Flutter у формуванні компетентностей

Особливості Flutter дозволяють не просто вивчати мобільну розробку, а цілеспрямовано розвивати кожну з вищезазначених компетентностей.

3.1. Формування компетентностей з UI/UX через декларативний підхід

Декларативна парадигма Flutter, де інтерфейс є функцією від стану, та інструмент Hot Reload створюють ідеальні умови для експериментів. Студенти можуть миттєво бачити результат змін у коді, що значно прискорює ітерації розробки та заохочує до пошуку кращих візуальних рішень. Це сприяє глибокому засвоєнню принципів UI/UX, оскільки теоретичні знання одразу підкріплюються практикою.

3.2. Розвиток кросплатформеного мислення

Робота з єдиною кодовою базою змушує студентів мислити абстрактно, створюючи універсальну бізнес-логіку та адаптивні інтерфейси. При цьому Flutter не ізолює від специфіки платформ.

Завдяки механізму Platform Channels, студенти можуть навчитися писати платформо-специфічний код (на Swift/Kotlin) та інтегрувати його у Flutter-застосунок, що формує глибоке розуміння того, як працює взаємодія між спільним та нативним кодом

3.3. Перспективність Flutter як мотиваційний фактор

Згідно з дослідженнями ринку, Flutter стабільно входить до числа найбільш затребуваних кросплатформених технологій[1].

Усвідомлення того, що отримані навички є актуальними та відкривають широкі кар'єрні перспективи, значно підвищує мотивацію студентів. Розширення екосистеми Flutter на веб, десктопні та вбудовані системи робить знання ще більш універсальними, готуючи фахівців, здатних створювати рішення для будь-якої платформи.

3.4. Вплив мови Dart на формування фундаментальних навичок програмування

Dart є сучасною об'єктно-орієнтованою мовою зі строгою типізацією та підтримкою null safety. Вивчення Dart закладає міцний фундамент з принципів ООП, роботи з типами даних та асинхронного програмування (async/await), що є універсальними навичками для будь-якого розробника.

4. Практичні рекомендації щодо впровадження Flutter в освітній процес

Для успішної інтеграції Flutter у навчальні програми доцільно дотримуватися наступних рекомендацій:

– Структура навчального курсу. Курс може бути побудований за принципом «від простого до складного»:

1. Основи мови Dart.
2. Концепція віджетів та побудова UI.
3. Основи управління станом (StatefulWidget, Provider).
4. Навігація між екранами.
5. Робота з мережевими запитами та API.
6. Зберігання даних (локальні та хмарні бази).
7. Поглиблені архітектурні патерни (BLoC/Riverpod).
8. Виконання фінального комплексного проєкту.

Типові навчальні проєкти. Слід пропонувати проєкти різної складності: від «клону» інтерфейсу відомого застосунку для відпрацювання навичок UI до повноцінного клієнт-серверного застосунку (напр., список новин, простий месенджер).

– Використання офіційних ресурсів. Google надає велику кількість якісних безкоштовних навчальних матеріалів, таких як офіційна документація, codelabs(покрокові практичні завдання) та відеокурси, які варто активно використовувати в освітньому процесі.

Висновки. Проведений аналіз доводить, що фреймворк Flutter є не лише сучасною технологією, а й потужним дидактичним інструментом. Його архітектурні особливості та екосистема цілеспрямовано сприяють формуванню повного спектра професійних компетентностей, що вимагаються сучасним ринком праці – від технічних навичок до архітектурного мислення. Інтеграція Flutter в освітній процес дозволяє інтенсифікувати навчання, підвищити мотивацію студентів та підготувати конкурентоспроможних фахівців, здатних ефективно вирішувати бізнес-завдання у сфері мобільної розробки. Подальші дослідження можуть бути спрямовані на розробку деталізованих методик оцінювання сформованих компетентностей та порівняльний аналіз ефективності навчання Flutter з іншими технологіями.

БІБЛІОГРАФІЧНИЙ СПИСОК:

1. 2025 Stack Overflow Developer Survey. Stack Overflow Insights – Developer Hiring, Marketing, and User Research. URL: https://survey.stackoverflow.co/2025/ (дата звернення: 25.09.2025).

2. Core Components and Native Components · React Native. React Native · Learn once, write anywhere. URL: https://reactnative.dev/docs/intro-react-native-components (дата звернення: 25.09.2025).

3. Dart documentation. Dart programming language. URL: https://dart.dev/docs (дата звернення: 25.09.2025).

4. Hot reload. Docs | Flutter. URL: https://docs.flutter.dev/tools/hot-reload (дата звернення: 25.09.2025).

5. Ichanska N., Ulko S. Основні аспекти створення мобільних додатків та вибір інструментів їх розробки. *Системи управління, навігації та зв'язку.* Збірник наукових праць. 2020. Т. 1, № 59. С. 74–78. URL: https://doi.org/10.26906/sunz.2020.1.074 (дата звернення: 25.09.2025).

6. Skia. Skia. URL: https://skia.org (date of access: 25.09.2025).

7. What is .NET MAUI? – .NET MAUI. Microsoft Learn: Build skills that open doors in your career. URL: https://learn.microsoft.com/uk-ua/dotnet/maui/what-is-maui?view=net-maui-9.0 (дата звернення: 25.09.2025).

8. Zarichuk O. Comparative analysis of frameworks for mobile application development: Native, hybrid, or cross-platform solutions. *Вісник Черкаського державного технологічного університету.* 2023. Vol. 28, no. 4. P. 19–27. URL: https://doi.org/10.62660/2306-4412.4.2023.19-27 (ддата звернення: 25.09.2025).

Дата першого надходження рукопису до видання: 29.09.2025
Дата прийнятого до друку рукопису після рецензування: 31.10.2025
Дата публікації: 20.11.2025