УДК 004.45

ЗРІВНЯННЯ РОЗРОБКИ ДЛЯ IOS ТА ANDROID В КОНТЕКСТІ СУЧАСНИХ МОБІЛЬНИХ ТЕХНОЛОГІЙ

Бєгунова В.Д.
e-mail: veronika.biehunova@nure.ua
Науковий керівник – к.т.н., ас. Кобилін І. О.
Харківський національний університет радіоелектроніки, каф. ІНФ,
м. Харків, Україна

The work compares the main differences between developing mobile applications for iOS and Android platforms. It analyzes the features of each platform's ecosystem, including the choice of programming languages, development environments, and user interface design requirements. It discusses the benefits and challenges that developers face, including issues of performance, stability, testing, and adaptation of applications to different devices. It also focuses on the features of publishing applications on the App Store and Google Play. The study allows us to better understand what factors influence the choice of approach to developing mobile applications depending on business needs and user expectations.

Мобільні додатки є невід'ємною частиною цифрової екосистеми, значною мірою визначаючи спосіб взаємодії користувачів з інформацією та сервісами. Розробка додатків для двох провідних мобільних платформ – iOS та Android – є ключовою складовою сучасної мобільної індустрії. Хоча обидві платформи надають потужні можливості для розробників, кожна з них має свої характерні риси, що впливають на вибір технологій, стратегій та підходів у розробці.

Платформа iOS була розроблена компанією Apple як частина закритої екосистеми, де жорстко контролюється взаємодія між апаратним та програмним забезпеченням. Такий підхід дозволяє Apple досягати високої продуктивності та стабільності своїх пристроїв. Розробка для iOS використовує мовою програмування Swift, що була створена компанією для підвищення безпеки, продуктивності та зручності кодування. Swift є сучасною мовою з лаконічним синтаксисом, що підвищує ефективність розробки та знижує кількість помилок, у порівнянні з традиційною Objective-C. Основним середовищем розробки є Xcode, яке забезпечує інтеграцію з іншими сервісами та інструментами Apple, що значно спрощує процес тестування, відлагодження і розгортання додатків [1].

Android, розроблений компанією Google, є відкритою платформою, що дозволяє різним виробникам мобільних пристроїв реалізовувати свої продукти на основі Android. Це відкриття надає більше гнучкості для розробників, але водночас створює певні складнощі через різноманіття пристроїв з різними характеристиками. Офіційними мовами програмування для Android є Java та Kotlin. Kotlin, як новіший мовний стандарт, офіційно підтримується Google і забезпечує покращену сумісність з Java, водночас пропонуючи

більш лаконічний синтаксис і знижуючи кількість можливих помилок у коді [2]. Для розробки під Android використовується Android Studio, яке має потужні інструменти для аналізу продуктивності, тестування на різних пристроях та емуляції.

Платформа iOS забезпечує високу продуктивність і стабільність завдяки глибокій оптимізації між програмним і апаратним забезпеченням. Завдяки обмеженій кількості моделей пристроїв (iPhone, iPad) розробники можуть створювати додатки, які будуть стабільно працювати на всіх підтримуваних пристроях [3]. Крім того, App Store має жорсткі вимоги щодо безпеки та якості додатків, що включають ретельну перевірку кожного додатку перед публікацією. Цей процес дозволяє зменшити ризики безпеки та підвищити довіру користувачів до додатків.

Android надає більше свободи вибору для розробників, однак це також може призвести до низки проблем, зокрема, через велику кількість пристроїв і версій операційної системи [4]. Фрагментація платформи ускладнює тестування додатків на різних пристроях з різними характеристиками екранів, процесорів, версіями Android, що може призвести до проблем із сумісністю та безпекою додатків.

Apple надає розробникам чіткі рекомендації для дизайну інтерфейсу користувача в рамках Human Interface Guidelines. Ці рекомендації орієнтовані на створення простих, естетично привабливих і функціональних інтерфейсів, що зручні для користувачів. Вони акцентують увагу на принципах інтуїтивно зрозумілого користувацького досвіду, використовуючи візуальні елементи, що забезпечують логічність та ясність у використанні.

В свою чергу, Google пропонує дизайн-стратегію Material Design, що також спрямована на створення уніфікованого користувацького досвіду на всіх пристроях Android [5]. Основними акцентами Material Design є простота, адаптивність та зручність, із великою увагою до чіткої ієрархії елементів інтерфейсу. Крім того, Android вимагає більш детального підходу до адаптації інтерфейсу для різних розмірів екранів і роздільних здатностей, оскільки на платформі існує велика різноманітність пристроїв.

Важливим аспектом розробки під iOS є обмежена кількість пристроїв, що спрощує процес оптимізації додатків. Завдяки цьому розробники можуть швидше забезпечити стабільну роботу додатків на всіх підтримуваних моделях пристроїв.

Android підтримує значно ширший діапазон пристроїв, що дозволяє охопити велику кількість користувачів. Однак це також ускладнює процес тестування та оптимізації. Різні виробники можуть використовувати різні версії Android, а пристрої можуть мати різні характеристики екранів, процесорів і пам’яті. Для забезпечення коректного відображення додатків розробники використовують гнучкі макети та відносні одиниці вимірювання, що дозволяє адаптувати інтерфейси під різноманітні пристрої.

Розробка для iOS може займати менше часу завдяки обмеженій кількості моделей пристроїв, що вимагають підтримки, а також уніфікованій екосистемі. Проте суворі вимоги Apple до дизайну, функціональності та безпеки додатків можуть призвести до додаткових витрат часу на перевірку та відповідність встановленим стандартам. Для Android час розробки може бути більше через необхідність врахування великої кількості пристроїв і версій операційної системи. Водночас, відкритий характер платформи дозволяє розробникам більш гнучко реагувати на зміни та тестувати нові функції на широкому спектрі пристроїв.

Таблиця 1 – Зрівняння iOS та Android

Характеристика	iOS	Android	Коефіцієнт порівняння (iOS : Android)
Мова програмування	Swift (сучасний, швидший) / Objective-C (старий)	Kotlin (сучасний) / Java (старий)	1:1.2 (Kotlin трохи легший за Swift)
Продуктивність додатків	Вища, оскільки Apple контролює залізо та ПО	Варіюється, залежить від пристрою	iOS стабільніший)
Середній час розробки	30-40% швидше через контрольовану екосистему	Довше через фрагментацію пристроїв	iOS швидший у розробці)
Фрагментація пристроїв	~30 моделей IPhone (активних)	10 000+ моделей Android-пристроїв	Android більш фрагментований)
Частка ринку	~28% глобально	~71% глобально	Android домінує)
Процес публікації	Жорсткий контроль (2-5 днів)	Менше обмежень (від декількох годин)	iOS важче пройти модерацію)
Оновлення ОС	90% користувачів оновлюються на нові версії	Більше 50% користувачів – старі версії	iOS швидше оновлюється)
Гнучкість налаштувань	Обмежена	Висока (можна замінити багато параметрів ОС)	Android більш гнучкий)

Список використаних джерел:

1. Neuburg M. iOS 15 Programming Fundamentals with Swift: Swift, Xcode, and Cocoa Basics. O'Reilly Media, 2021.
oore K. D., Mota C., Taheri S. Kotlin Multiplatform by Tutorials: Build Native Apps Faster by Sharing Code Across Platforms. Kodeco, 2023.
arook F. UIKit Apprentice: Beginning iOS Development with Swift. Kodeco, 2024.
aase C. Androids: The Team That Built the Android Operating System. 1st ed. Kindle Edition, 2021.
5. Meike G. B., Schiefer L. Inside the Android OS: Building, Customizing, Managing and Operating Android System Services. 1st ed. Paperback, 2021.