

Розробка кросплатформенних мобільних додатків

¹ Вінницький національний технічний університет

Анотація

Проаналізовано сучасні методи розробки кросплатформенних мобільних додатків

Ключові слова: мобільний додаток, flutter, react native, кросплатформенність.

Abstract

Modern methods of cross-platform mobile application development are analyzed

Keywords: mobile application, flutter, react native, cross-platform.

Вступ

Мобільні додатки в сучасному світі мають особливе місце, оскільки вони стали невід’ємною частиною повсякденного життя. Вони розширюють можливості мобільних пристроїв та дозволяють виконувати різноманітні завдання та отримувати інформацію швидко та зручно.

Розробка мобільних додатків з кожним роком збільшує свою частку на ринку розробки програмного забезпечення. Згідно галузевого огляду, проведеним компанією Statista загальна кількість перших завантажень мобільних додатків у всьому світі становила 95.1 мільярда у 2017 році, і зростає до 142,6 мільярдів у 2023 році і тримається стабільно протягом останніх років. Ці цифри зростають з кожним роком, а разом з ними зростає і ринок мобільних додатків. У 2021 році вперше було завантажено понад 100 мільярдів додатків, і очікується, що до 2025 року їх кількість досягне 200 мільярдів. На рисунку 1 наведено графік поквартального завантаження мобільних додатків з 2015 по 2023 роки. Цікавим, до речі, є скачок завантажень у першому та другому кварталах 2020 року, коли розпочалась епідемія Covid-19.

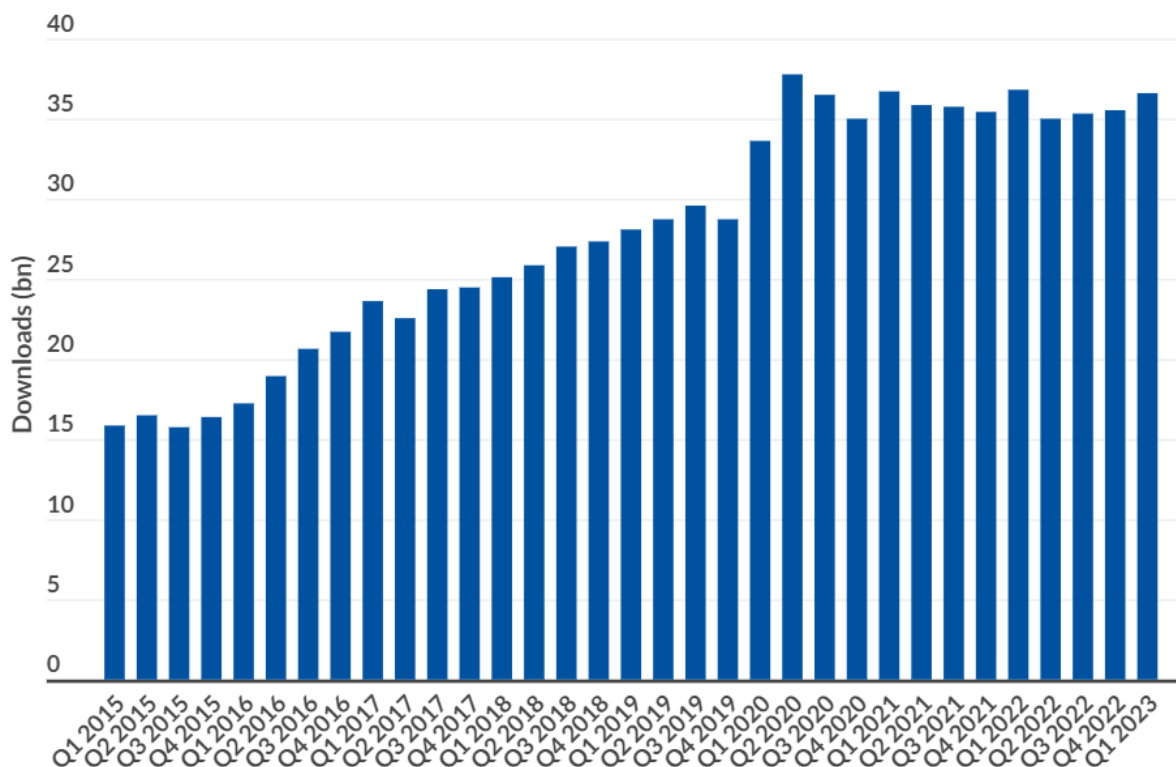


Рисунок 1 – статистика поквартального завантаження мобільних додатків

Така велика кількість користувачів не може бути знехтувана бізнесом. Тому компанії та сервіси все частіше створюють і мобільні версії своїх продуктів, щоб отримати більше і не втратити своїх активних користувачів.

На сьогоднішній день основними актуальними операційними системами, або платформами, для мобільних пристроїв є два види: Android та iOS. Обидві платформи активно користуються популярністю серед споживачів. Згідно статистики активна частка iOS відповідає близько 22% від ринку і у Android 77%. Не зважаючи на те, що частка Android користувачів більша, частина iOS є досить значною, тому при розробці мобільних додатків для продукту зазвичай створюють дві версії мобільного додатку для кожної з платформ. Створення двох додатків значно впливає на бюджет розробки, а також час на її тестування та впровадження.

Для вирішення даної проблеми було запропоновано метод кросплатформної розробки мобільних додатків, що передбачає створення та використання однієї кодової бази для декількох платформ. Методи кросплатформної розробки значно пришвидшили та здешевили розробку мобільних додатків для бізнесу при цьому не втрачаючи значної якості розробки.

Ось декілька популярних кросплатформних фреймворків і інструментів для розробки мобільних додатків:

React Native [2] – є одним з найпопулярніших кросплатформних фреймворків для розробки мобільних додатків. Він базується на JavaScript і використовує компонентний підхід, що дозволяє швидко створювати додатки для iOS і Android з одним кодом.

Flutter [3] – це інший потужний кросплатформний фреймворк, розроблений компанією Google. Він використовує мову програмування Dart і має вбудовану бібліотеку графічного інтерфейсу. Flutter дозволяє розробникам створювати високоякісні інтерфейси користувача, які працюють на різних платформах.

Xamarin [4] – це фреймворк для кросплатформної розробки, що базується на мові програмування C#. Він дозволяє розробникам створювати додатки для iOS, Android і Windows, використовуючи спільний код.

Ionic [5] – це веб-фреймворк для розробки кросплатформних мобільних додатків з використанням HTML, CSS і JavaScript. Він використовує стандартні веб-технології для створення додатків, які можуть працювати на різних платформах.

Згідно статистики, проведеної компанією Fireart [6], доля React Native та Flutter серед кросплатформних мобільних додатків є найсуттєвішою на ринку, тому далі будуть проаналізовані обидва підходи. Доля Flutter складає 39%, а доля React Native трохи більша, і складає 42%, це близько 2 млн розробників. Проте кожного року кількість розробників Flutter збільшується, і за прогнозами через рік може обігнати React Native. Зважаючи на те, що React Native був випущений у 2015 році і завоював велику долю ринку першим, а Flutter з'явився тільки у 2018, то темпи розвитку Flutter є досить перспективними.

Головною відмінністю двох підходів є використання різних мов програмування, а саме React Native використовує мову Java Script, а Flutter – мову Dart. Використання Java Script у React Native має досить велику перевагу, оскільки кількість розробників на Java Script є значною, і перехід на React Native, наприклад, з Web розробки не займає великої кількості часу, а отже не має дефіциту розробників для даного підходу. У випадку з Dart, він потребує деякого часу адаптації, проте дана мова програмування є досить схожа на інші відомі мови програмування, такі як C#, Java, Java Script, тому її вивчення також не потребує великої кількості часу.

Обидва підходи використовують реактивне програмування, це парадигма програмування, що ставить акцент на реагуванні на зміни даних і подій у реальному часі. У реактивному програмуванні використовують потоки даних (streams) і реактивні оператори для визначення залежностей і взаємодії між елементами системи. Перевагою даної парадигми є легка обробка асинхронних подій і змін даних, зменшення складності управління станом системи, висока гнучкість та модульність, підвищена реакція та швидкість системи.

Проте, що стосується швидкодії системи, то використання Dart у Flutter є суттєвою перевагою над Java Script у React Native. Dart дозволяє компілювати код швидше, крім цього Flutter відображає анімацію у швидкості 60 кадрів в секунду. Як правило, всі програми на Flutter за замовчуванням мають високу продуктивність. В основному це досягається за допомогою особливості Dart коду, який компілюється у нативну мову програмування для кожної платформи. Так, наприклад, для Android код

написаний на Dart буде скомпільовано у Kotlin, а для iOS у Swift. Таким чином швидкість роботи додатку буде майже на рівні якби додаток було написано нативним методом.

Важливою також для мов програмування є підтримка спільноти. З цим у обох підходів немає жодних проблем. Обидва підходи мають низку готових відкритих бібліотек, а також велику спільноту у GitHub та на інших платформах, що дозволяє пришвидшувати розробку та пошук вирішення проблем. Метою Flutter Community є надання інформації та ресурсів для всього, що стосується Flutter. Незалежно від рівня навичок, кожен розробник може знайти інформацію на каналі YouTube, який містить тижневі робочі години та хакатони Flutter.

Однією з переваг Flutter над React Native є підтримка шести платформ для розробки. Flutter не обмежений лише розробкою для Android та iOS, він також підтримує WEB розробку, а також Windows, MacOS та Linux. Це дає перспективу використання однієї кодової бази для всіх перелічених платформ, що неодмінно є досить потужним фактором для вибору Flutter як підходу до розробки.

Головною перевагою Flutter над React Native є можливість взаємодії з нативним кодом, написаним на мові Swift, Kotlin у випадку iOS та Android. Це означає, що Flutter має змогу використовувати нативні бібліотеки, методи та функції без необхідності створення аналогів на мові Dart. Так, наприклад, можна підключити бібліотеку ffmpeg для здійснення операцій з відеоматеріалами.

Висновки

Як ми бачимо зараз, Flutter і React Native мають багато спільного. Перш за все, обидва вони є фреймворками з відкритим кодом, якими може користуватися кожен. Обидва вони розроблені провідними компаніями: Google створив Flutter, а React Native — Facebook. React Native деякий час був гарячою темою, особливо у світі мобільної розробки. Не дивно – він штурмом захопив світ технологій, пропонуючи спосіб розробки мобільних додатків як для iOS, так і для Android одночасно. Він був успішно прийнятий сотнями компаній у всьому світі, включаючи Uber, Microsoft і Facebook, і використовується багатьма іншими. Однак перш ніж обрати React Native, дуже важливо зрозуміти, як це працює, і визначити, чи краще це підходить для вашого проекту. Flutter легше використовувати, оскільки він більш стійкий до оновлень системи. Це означає, що коли iOS або Android оновлять ОС, програма залишиться незмінною. Навпаки, React Native залежить від нативних елементів, тому після випуску оновлення можуть з'явитися деякі проблеми в програмі запуску. Наразі Flutter підтримує Android 4.1+ або iOS 8+, тоді як React Native підтримує Android 4.1+ та iOS 10+. Найбільша різниця між React Native і Flutter полягає в тому, що у Flutter є плагіни, розроблені командою Google, як-от геолокація та відображення. Загалом, Flutter і React Native — чудовий вибір, щоб заощадити час і гроші на розробці мобільних додатків.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Statista [Електронний ресурс] – режим доступу: <https://www.businessofapps.com/data/app-statistics>.
2. Flutter [Електронний ресурс] – режим доступу: <https://flutter.dev>.
3. React Native [Електронний ресурс] – режим доступу: <https://reactnative.dev>.
4. Xamarin [Електронний ресурс] – режим доступу: <https://visualstudio.microsoft.com/xamarin>.
5. Ionic [Електронний ресурс] – режим доступу: <https://ionicframework.com>.
6. Fireart, React native vs Flutter [Електронний ресурс] – режим доступу: <https://fireart.studio/blog/flutter-vs-react-native-what-app-developers-should-know-about-cross-platform-mobile-development/>

Прус Богдан Вікторович — аспірант кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, bohdan.prus.vntu@gmail.com

Prus Bohdan Viktorovych — PhD student of Soft Ware Design Department, Vinnytsia National Technical University, Vinnytsia, bohdan.prus.vntu@gmail.com