

## ПОРІВНЯННЯ НАТИВНОЇ РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ З УНІФІКОВАНОЮ РОЗРОБКОЮ ЗА ДОПОМОГОЮ REACT NATIVE

Залазаєв А. М., Кириченко І. В.

Харківський національний університет радіоелектроніки, Харків, Україна

На сьогодні найпоширенішою мовою програмування у світі є JavaScript [1], його розповсюдженням ми зобов'язані швидкому росту інтернет технологій, дуже велика кількість розробників, що не використовують Javascript як основну мову програмування зустрічаються з ним під час розробки ПЗ на рівні інтерфейсу користувача. Підхід з використанням уніфікованої розробки має позбавити від необхідності створення додатків для iOS та Android окремо та надасть розробникам знайомий функціонал з використанням Javascript.

**Метою доповіді** є дослідження зручності та швидкості створення ПЗ для мобільних телефонів на базі Android (з використанням Java) та iOS (з використанням Swift) у порівнянні з універсальною розробкою на React Native.

В доповіді наводяться результати вимірювань розмірів вихідного коду та швидкодії застосунку в залежності від обраного методу написання коду. Результати порівняння вихідного коду показали, що підхід React Native потребує майже у 2,5 рази менше коду через те, що кодова база тепер спільна і більше не потрібно писати один і той самий код для різних платформ, також було визначено, що Android платформа з використанням Java містить дуже багато зайвого коду та є громіздким у порівнянні з сучасним React Native. Основні вказівки та концепції React Native мають сильний вплив на розмір вихідного коду проекту, компонентна архітектура дозволяє не копіювати один і той самий код, а пере використовувати його. Окрім цього React Native надає асинхронність яка дуже погано реалізована у Java та Swift у звичному розробникам форматах fetch-синтаксису та async/await та не призводить до безлічі callback та thread, що роблять код не придатним до читання. Також було визначено, що написання користувацького інтерфейсу у декларативному стилі Android та iOS порушує паттерн проектування SOLID Single responsibility у більшості випадків та призводить до створення додатків, що складно підтримувати, React Native надає змогу використовувати CSS та розділяти інтерфейс від логіки. Результати порівняння швидкодії застосунків дозволили зробити висновок, що з використанням React Native застосунки споживають більшу кількість пам'яті, через використання браузера для відображення, але надають можливість швидко та гнучко створювати універсальні додатки для обох платформ. Можливість використання вже існуючої кодової бази та Typescript, у сумі з наведеними вище перевагами роблять React Native серйозним конкурентом на ринку мобільних додатків.

### Список літератури

1. The Most Popular JavaScript Frameworks: <https://www.makeuseof.com/most-popular-javascript-frameworks/> (дата звернення: 02.04.2022).