

УДК 004

КРОССПЛАТФОРМЕННАЯ РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ: ПРЕИМУЩЕСТВА, НЕДОСТАТКИ И ПЕРСПЕКТИВЫ

Ткачёв Е.В.

магистрант,

ФГБОУ ВО «Калужский государственный университет

им. К.Э. Циолковского»

Калуга, Россия

Белаш В.Ю.

к.пед.н., доцент,

ФГБОУ ВО «Калужский государственный университет

им. К.Э. Циолковского»

Калуга, Россия

Аннотация: В данной статье рассматриваются основные преимущества и недостатки кроссплатформенной разработки мобильных приложений. Проанализированы два ключевых подхода: гибридные решения на основе веб-технологий и нативные кроссплатформенные фреймворки. Особое внимание уделено таким решениям, как Flutter, React Native и Ionic. На основе анализа рынка кроссплатформенных технологий сделан вывод о текущих тенденциях и перспективах развития данного направления.

Ключевые слова: кроссплатформенная разработка, мобильные приложения, Flutter, React Native, Ionic, мобильная разработка.

***CROSS-PLATFORM MOBILE APPLICATION DEVELOPMENT:
ADVANTAGES, DISADVANTAGES AND PROSPECTS***

Tkachev E.V.

Master's student,

Kaluga State University named after K. E. Tsiolkovsky

Kaluga, Russia

Belash V.Yu.

Ph.D., Associate Professor,

Kaluga State University named after K. E. Tsiolkovsky

Kaluga, Russia

Annotation. This article discusses the main advantages and disadvantages of cross-platform mobile application development. Two key approaches are analyzed: hybrid solutions based on web technologies and native cross-platform frameworks. Special attention is paid to solutions such as Flutter, React Native and Ionic. Based on the analysis of the cross-platform technology market, a conclusion is drawn about the current trends and prospects for the development of this area.

Keywords: cross-platform development, mobile applications, Flutter, React Native, Ionic, mobile development.

С момента появления мобильных приложений разработчики сталкиваются с задачей создания программных продуктов, которые должны одинаково хорошо работать на разных операционных системах (Android, iOS). Традиционный (нативный) подход предполагает разработку отдельных версий приложений для каждой из платформ, что является очень затратным занятием, поскольку фактически требуется разрабатывать два разных приложения с уникальной кодовой базой. Такой метод значительно увеличивает время и бюджет на разработку, внедрение новых функций и поддержку.

Для оптимизации разработки разработчики прибегают к кроссплатформенным технологиям разработки, которые позволяют использовать единый код сразу на нескольких системах. Они сокращают время разработки и ускоряют процесс внедрения новых функций за счет единой логики, кроме того современные кроссплатформенные фреймворки обладают

встроенными инструментами для тестирования и отладки, что значительно упрощает работу разработчиков.

Однако помимо очевидных преимуществ имеется ряд серьёзных недостатков: низкая производительность, ограниченность нативных функций, различия пользовательского интерфейса на разных платформах. Таким образом, кроссплатформенные технологии предоставляют эффективное решение для мобильной разработки, однако их использование требует взвешенного подхода. Далее в статье будут рассмотрены современные кроссплатформенные фреймворки, их ключевые особенности, сильные и слабые стороны, а также проанализированы изменения в их популярности и перспективы дальнейшего развития.

Современную кроссплатформенную разработку можно разделить на два основных подхода:

1. Гибридные решения – основаны на веб-технологиях (HTML, CSS и JavaScript), где приложения запускаются внутри встроенного браузера (WebView) [4]. Такой подход удобен для веб-разработчиков, так как позволяет быстро адаптировать веб-приложения для мобильных платформ. Однако использование WebView негативно сказывается на производительности и доступе к нативным функциям устройств.

2. Нативные кроссплатформенные фреймворки – компилируют код в нативный, обеспечивая лучшую производительность и доступ к системным API. Они также позволяют писать код один раз, но при этом компилируют его в нативный код для каждой платформы. Это обеспечивает высокую производительность и более естественный пользовательский интерфейс, чем у гибридных решений.

Каждый из подходов имеет сильные и слабые стороны. Рассмотрим несколько наиболее популярных решений: Flutter, React Native и Ionic – их ключевые особенности, преимущества и недостатки.

Ionic – один из старейших кроссплатформенных решений, который появился в 2013 году. В его основе лежат хорошо знакомые веб-технологии: HTML, CSS и JavaScript. Главная идея данного фреймворка – это дать возможность веб-разработчикам быстро разработать мобильное приложение, доступное сразу на нескольких платформах [3]. В основе Ionic лежит WebView, встроенный браузер в мобильной системе, это значит, что приложение по своей сути является веб-приложением.

Главным преимуществом является простота и скорость разработки мобильного приложения за счет использования веб-технологий. Ещё одним плюсом является поддержка работы популярных библиотек (таких как Angular, React и Vue), а также большое количество готовых решений.

Однако у фреймворка имеются и значительные недостатки, которые являются следствием использования WebView: производительность, значительно уступающая нативным решениям и ограниченный доступ к нативным API и интерфейс готового приложения, который может не соответствовать стилистике и поведению нативных iOS и Android приложений.

Flutter – на текущий момент является самым популярным фреймворком, в его основе лежит собственный язык разработки Dart, который схож с JavaScript и C#, для рендеринга используется собственный движок Skia [1].

В преимущества данного фреймворка можно записать высокую производительность благодаря Skia, также с его помощью интерфейс выглядит одинаковым на всех платформах, так как он использует собственные UI-компоненты. В плюсы также можно записать Hot Reload, позволяющий показывать результат изменения в коде без пересборки всего приложения.

Основными минусами являются новизна и относительная сложность языка программирования Dart и большой размер приложения, в сравнении с конкурентами

React Native – позволяет писать мобильные приложения на JavaScript с использованием знакомого по веб-разработке React. Фреймворк в отличии от Дневник науки | www.dnevniknauki.ru | СМИ Эл № ФС 77-68405 ISSN 2541-8327

Flutter компилирует написанный код в реальные нативные компоненты операционных систем.

Основными преимуществами можно назвать крайне популярный язык программирования JavaScript, на котором основан фреймворк, использование реальных нативных компонентов, множество готовых пользовательских библиотек и решений и наличие функции Hot Reload.

Недостатки – более низкая производительность, в сравнение с Flutter, некоторые нативные компоненты доступны только в качестве сторонних пользовательских библиотек, интерфейс на разных платформах может отличаться.

Согласно данным [2], в 2021 году наиболее популярными кроссплатформенными фреймворками были Flutter (42%) и React Native (38%), тогда как Ionic, Cordova и Xamarin занимали значительно меньшую долю рынка. Однако к 2023 году произошли изменения в предпочтениях разработчиков, что отразилось на графике долей рынка и популярности отдельных технологий. На основе анализа данных можно выявить ключевые тренды развития кроссплатформенной мобильной разработки, что может позволить спрогнозировать дальнейшее движение рынка.

На рисунке 1 представлена диаграмма изменения популярности технологий за три года. На основе этих данных можно выделить тенденции кроссплатформенной разработки.

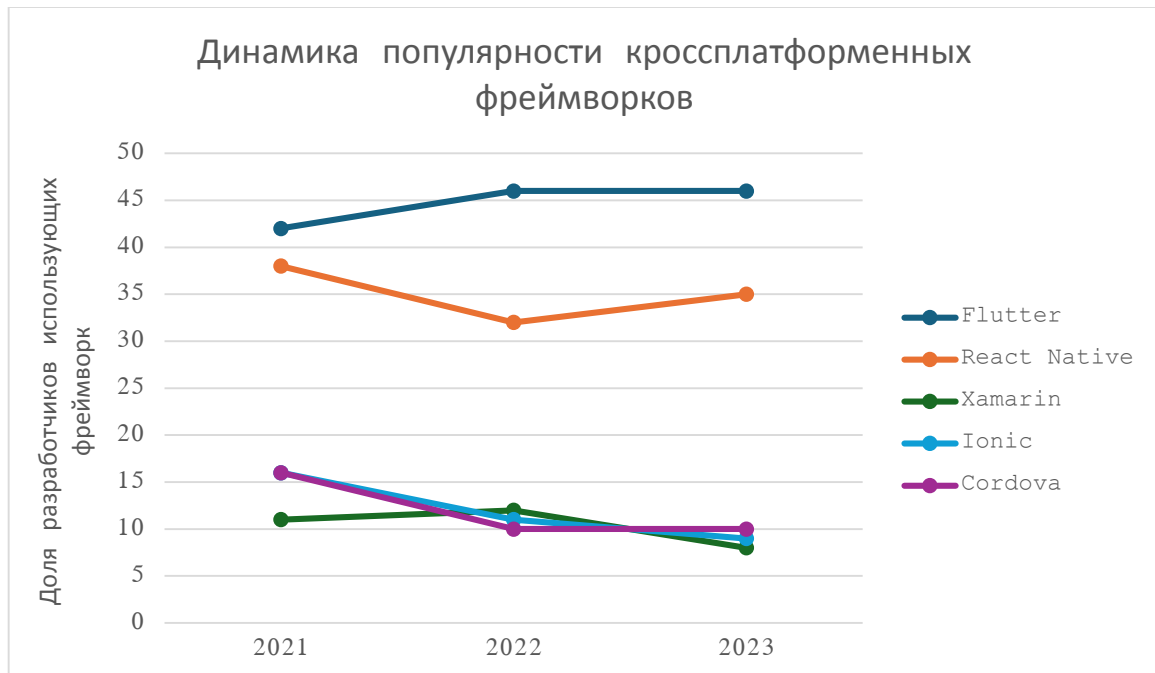


Рис. 1. Динамика популярности кроссплатформенных фреймворков¹

Прежде всего явно заметен рост популярности Flutter. Фреймворк увеличил свою долю с 42% до 46%, продолжая лидировать среди кроссплатформенных технологий. Данный рост обусловлен несколькими факторами:

- Активная поддержка Google способствует развитию фреймворка и регулярным обновлениям (включая Flutter 3).
- Высокая производительность, которая достигается благодаря рендерингу через движок Skia, что позволяет минимизировать зависимость от нативных UI-компонентов.
- Рост экосистемы, приводящий к увеличению числа библиотек, плагинов и документации, облегчающих вход в технологию.

Таким образом, Flutter продолжает доминировать на рынке, привлекая новых разработчиков за счет скорости, гибкости и производительности.

¹ Составлено авторами

У второго по популярности кроссплатформенного фреймворка React Native наблюдается снижение доли рынка на 3% (с 38% до 35%). Несмотря на это, он уверенно удерживает второе место и почти не уступает лидеру.

Основными причинами снижения могут быть:

- Укрепление конкуренции со стороны Flutter, который предлагает более предсказуемый UI и производительность, а также продолжает улучшать экосистему.

- Ограничения JavaScript-моста, который необходимо использовать для работы с нативными модулями, что негативно влияет на скорость работы.

- Проблемы с анимацией и сложными вычислениями – разработчики предпочитают Flutter для графически сложных приложений.

Тем не менее, React Native остается востребованным благодаря популярности и относительной простоты JavaScript и React, что делает его удобным выбором для веб-разработчиков, переходящих в мобильную разработку.

Xamarin потерял 3% разработчиков (с 11% до 8%), что указывает на его постепенное вытеснение с рынка. Тем не менее это один из немногих вариантов на рынке для разработки на языке C# с использованием платформы .NET. Основные причины падения:

- Microsoft делает ставку на .NET MAUI, заменяя Xamarin, что означает низкий темп обновлений и сокращение сообщества.

- Малое сообщество и количество библиотек в сравнении с Flutter и React Native.

Ionic и Cordova показали наибольшее снижение – с 16% до 9% (Ionic) и с 16% до 10% (Cordova). Оба фреймворка являются представителями гибридных WebView технологий, что, вероятно, и является основным драйвером значительной потери популярности:

- Гибридные приложения не могут полноценно использовать некоторые нативные функции устройств.

- Переход разработчиков на Flutter и React Native, которые обеспечивают лучшую производительность.

Таким образом, гибридные технологии постепенно уходят с рынка, уступая место более производительным инструментам.

Подводя итоги можно отметить, что будущее на данный момент за Flutter и React Native – именно они будут определять развитие кроссплатформенной мобильной разработки. Ionic и Cordova постепенно уходят с рынка из-за наличия более совершенных инструментов кроссплатформенной мобильной разработки, однако они всё ещё востребованы в случае, когда необходимо быстро создать прототип или приложение с простым графическим интерфейсом. Xamarin теряет позиции из-за того, что Microsoft перестает активно его поддерживать, однако стоит следить за .NET MAUI, который может активно продвигаться корпорацией в ближайшем будущем. Также стоит отметить, что общий рост кроссплатформенных технологий продолжится, что приведет к дальнейшему улучшению их интеграции с нативными API.

Кроссплатформенные технологии упрощают и ускоряют разработку мобильных приложений, сокращая затраты и издержки, однако они уступают нативным решениям в производительности и доступе к системным возможностям. Современные фреймворки, такие как Flutter и React Native, продолжают лидировать, тогда как гибридные решения теряют актуальность. В дальнейшем ожидается развитие кроссплатформенных технологий: улучшение их интеграции с нативными API и повышение производительности.

Библиографический список

1. Biessek, A. Flutter for Beginners: An introductory guide to building cross-platform mobile applications with Flutter 2.5 and Dart. 1st ed. Birmingham, United Kingdom of Great Britain: Packt Publishing, 2021. 370 p.

2. Cross-platform mobile frameworks used by software developers worldwide from 2019 to 2021. Statista. [Электронный ресурс]. URL: <https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/> (дата обращения: 02.03.2025).

3. Столяров, И.С. Кроссплатформенная разработка мобильных приложений: сравнение фреймворков и инструментов / И.С. Столяров, Р.Р. Садриев, Н.А. Романовский // Студент и наука: актуальные вопросы современных исследований : сборник статей V Международной научно-практической конференции, Пенза, 12 марта 2024 года. – Пенза: Наука и Просвещение (ИП Гуляев Г.Ю.), 2024. – С. 42-44.

4. Чуриков, Е.А. Сравнение современных средств разработки мобильных приложений / Е.А. Чуриков, Т.В. Зудилова, И.В. Ананченко, Н.А. Осипов, С.Е. Иванов, И.С. Осетрова // Современные наукоемкие технологии. – 2022. – № 12-1. – С. 82-87; URL: <https://top-technologies.ru/ru/article/view?id=39441> (дата обращения: 02.03.2025).

Оригинальность 76%