

*Колчанова Светлана Анатольевна, студент 2 курса магистратуры  
факультета математики и информационных технологий  
ФГБОУ ВО «НИ МГУ им. Н. П. Огарёва», г. Саранск*

## **СРАВНЕНИЕ ФРЕЙМВОРКОВ FLUTTER И REACT NATIVE, ИСПОЛЬЗУЕМЫХ В РАЗРАБОТКЕ ГИБРИДНЫХ ПРИЛОЖЕНИЙ**

**Аннотация:** Актуальность разработки гибридных приложений не вызывает сомнений. Такие приложения в отличие от программ, создаваемых для каждой платформы в отдельности, экономят при разработке множество ресурсов, как временных, так и материальных. В этой статье для сравнения представлены два наиболее популярных среди разработчиков инструмента для создания гибридных приложений – Flutter и React Native. Эти фреймворки обладают как достоинствами, так и недостатками, что сказывается на их дальнейшей перспективе сохранения востребованности на рынке программного обеспечения.

**Ключевые слова:** Flutter, React Native, Android, iOS, SDK, UI, фреймворк, Dart, JavaScript, кроссплатформенная разработка, гибридные приложения.

**Abstract:** The relevance of the development of hybrid applications is beyond doubt. Such applications, unlike programs created for each platform separately, save a lot of resources, both time and material, during development. For comparison, this article presents two of the most popular tools for creating hybrid applications among developers – Flutter and React Native. These frameworks have both advantages and disadvantages, which affects their further prospects of maintaining demand in the software market.

**Keywords:** Flutter, React Native, Android, iOS, SDK, UI, framework, Dart, JavaScript, cross-platform development, hybrid applications.

Рынок программного обеспечения заполнен кроссплатформенными (гибридными) приложениями. Возникает вопрос – почему так много компаний выбирают именно его? Давайте проясним концепции, прежде чем обсуждать это.

Каждая платформа (Android и iOS) имеет определенные языки, которые программисты используют для создания приложений для них. Для Android это Java, а для iOS – Swift и Objective-C. Каждая платформа также имеет набор определенных API и SDK, которые также используются в собственной разработке. В целом приложения, разработанные с использованием этих языков, программных интерфейсов и наборов для разработки, являются родными.

В результате собственной разработки мы получаем приложение, которое может быть запущено только в определенной операционной системе. Это затрудняет обслуживание и обновление. Более того, если компания хочет создать собственное приложение для Android и iOS, то необходимо выполнить эту работу дважды – разработать приложение с нуля для обеих систем по отдельности, что приведёт к увеличению затрат на разработку.

Гибридные приложения – это приложения следующего поколения, поскольку они обеспечивают независимость от платформы и одновременно доступ к встроенным функциям устройства. Кроме того, они эффективно работают в автономном режиме и могут быть загружены в магазины приложений. Все это достигается с помощью лучших гибридных фреймворков, включая Flutter и React Native.

Flutter – портативный инструментальный пользовательского интерфейса для создания изначально скомпилированных приложений на мобильных, веб- и настольных компьютерах из одной кодовой базы. Официально представлен компанией Google в декабре 2018 года.

Он основан на Dart – быстром объектно-ориентированном языке программирования. Он относительно новый и прост в освоении – особенно для опытных разработчиков, более знакомых с Java и C#. Более того, Dart поддерживает sound null safety. Это означает, что переменные не могут содержать null, если вы этого не скажете. Следовательно, по умолчанию типы в языке Dart

не подлежат обнулению, что предотвращает ошибки типизации [2].

React Native – фреймворк для построения нативных приложений с использованием React на языке JavaScript. Представлен раньше – 15 марта 2015 года на конференции Facebook.

Оба этих инструмента являются open source проектами с открытым исходным кодом.

Приложения Flutter выглядят так же хорошо в современных операционных системах, как и в старых версиях. Поскольку у них только одна кодовая база, приложения выглядят и ведут себя одинаково на iOS и Android, но благодаря Material Design и виджетам Cupertino они также могут имитировать сам дизайн платформы.

Flutter содержит два набора виджетов, которые соответствуют определенным языкам дизайна: виджеты Material Design реализуют одноименный язык дизайна Google; виджеты Cupertino имитируют дизайн iOS от Apple. Это означает, что приложение Flutter будет выглядеть и вести себя естественно на каждой платформе, имитируя их родные компоненты [1].

Компоненты приложения, написанного с использованием React Native, выглядят так же, как и в любой изначально встроенной программе на смартфоне (например, кнопка на устройстве iOS выглядит так же, как родная кнопка iOS, и то же самое на Android).

Тот факт, что React Native использует собственные компоненты, позволяет с уверенностью утверждать, что после любого обновления пользовательского интерфейса операционной системы компоненты вашего приложения также будут мгновенно обновлены.

Тем не менее, это может сломать пользовательский интерфейс приложения, но это происходит очень редко.

Если требуется, чтобы приложение выглядело почти идентично на разных платформах, а также корректно отображалось на экране устройств со старой версией операционной системы (как это предусмотрено во Flutter), необходимо рассмотреть возможность использования сторонних библиотек. Они позволят

использовать виджеты Material Design вместо компонентов, установленных по умолчанию.

С Flutter 2 (анонсированным в марте 2021 года) можно использовать одну и ту же кодовую базу для доставки собственных приложений в пять операционных систем: iOS, Android, Windows, macOS и Linux; а также веб-интерфейсы, ориентированные на такие браузеры, как Firefox, Chrome, Safari или Edge.

Flutter можно даже встроить в автомобили, телевизоры и умную бытовую технику.

Пожалуй, самое большое обновление Flutter до версии 2.0 вывело использование фреймворка для создания веб-приложений на новый уровень. Его можно использовать для:

- прогрессивных веб-приложений (PWA), которые сочетают охват сети с возможностями настольного приложения,
- одностраничных приложений (SPA), которые загружаются один раз и передают данные в интернет-сервисы и из них.
- существующих мобильных приложений, которые легко можно развернуть на веб-стороне.

React Native в свою очередь поддерживает iOS и Android, но есть отдельные библиотеки, которые позволяют использовать один и тот же код для создания приложений iOS, Android, web и Windows10.

Время выхода на рынок приложения с использованием Flutter обычно намного быстрее, чем при нативной разработке. С использованием React Native обычно разработка также занимает немного времени, однако этот фреймворк использует bridge и native elements, поэтому для каждой платформы может потребоваться отдельная оптимизация – проблема, с которой Flutter на основе виджетов не сталкивается. Это может сделать разработку приложения с React Native дольше.

Компоненты в React Native довольно просты, поэтому, если нужно что-то дополнительное, то при стилизации потребуются значительные усилия. Только

несколько компонентов адаптированы к платформе, в то время как большую часть времени приходится использовать другой компонент как для iOS, так и для Android; или стилизовать его по-другому. С другой стороны, с Flutter все является виджетом, что делает их легко настраиваемыми. Большинство виджетов адаптивны, и можно использовать один и тот же виджет как на Android, так и на iOS.

Когда дело доходит до производительности, Flutter имеет преимущество, поскольку он скомпилирован в собственные библиотеки ARM или x86, что делает его действительно быстрым. React Native не компилируется в собственный код, и у него все еще есть слой JavaScript, что делает его менее производительным, чем Flutter.

Трудно сравнивать их в целом, потому что они имеют разную архитектуру и, следовательно, функционируют по-разному. Воспользуемся некоторыми конкретными критериями, чтобы оценить их.

Язык. React Native использует JavaScript в качестве основы, в то время как Flutter использует Dart. С этой точки зрения, React Native легче изучать, потому что JavaScript является языком № 1 среди программистов, согласно опросу StackOverflow. Очевидно, что с учетом того, что почти 70% специалистов используют его, переход на React Native не составит труда. С Flutter ситуация иная. Этот фреймворк основан на Dart. Это новый и все еще незрелый язык. Несмотря на то, что молодые разработчики часто предпочитают изучать его, все еще очень мало программистов, которые уже освоили его. Однако, если разработчик когда-либо работал с объектно-ориентированными языками, то он быстро освоит Dart.

Представление. Это, вероятно, самое существенное различие между Flutter и React Native. В этой категории Flutter является бесспорным победителем. Благодаря движку C++, используемому в фреймворке Flutter, и графической библиотеке Skia, это позволяет создавать приложения с более высокой производительностью по сравнению с React Native. Кроме того, процесс кодирования происходит быстрее. React Native использует JavaScript-мост к

родному языку. Это означает, что две стороны – JavaScript и родной язык обмениваются сообщениями JSON для установления соединения. Эти сообщения являются асинхронными, и это приводит к довольно плавной анимации. Однако, по сравнению с Flutter, производительность пользовательского интерфейса все еще может иметь некоторые проблемы, такие как задержки в рендеринге. Фреймворк Flutter не имеет этой проблемы, поскольку у него нет этого моста. Он соединяется с собственными компонентами со встроенными библиотеками и такими фреймворками, как Material Design или Cupertino.

API и пользовательский интерфейс. На данный момент у Flutter есть более богатый набор возможностей. React Native должен использовать сторонние библиотеки, в то время как Flutter имеет встроенные компоненты рендеринга, инструменты для тестирования, навигации, доступа к API устройств и т. д.

IDE. Когда дело доходит до выбора среды разработки, важно выбрать ту, которая будет наиболее удобной. В случае с Flutter этот выбор не очень велик, потому что технология все еще новая. Visual Studio Code, IntelliJ и Android Studio поддерживают его. Принимая во внимание, что для React Native можно выбрать практически любую среду разработки.

Документация. Это еще одно различие между Flutter и React Native, которое так часто обсуждают начинающие программисты. Flutter имеет более богатую документацию по сравнению с React Native. Это позволяет молодым разработчикам изучать технологию, следуя ей. Но в случае с React Native такой способ не работает, так как его документация больше ориентирована на описание сложных процессов и подразумевает, что каждый, кто ее читает, уже знаком с JavaScript.

Развертывание. Flutter снова побеждает. Он имеет очень простой рабочий процесс для развертывания приложения в App Store или Google Play. Можно сделать с помощью командной строки (если не нужны дополнительные параметры настройки). Но даже для них доступно множество сторонних инструментов. С помощью React Native можно развернуть приложение только с

помощью сторонних решений. Более того, найти руководство о том, как это сделать, может оказаться непростой задачей [3].

Все больше и больше компаний привлекают Flutter. В конце концов, наблюдаются ежемесячные улучшения в SDK Flutter, поскольку Google продолжает совершенствовать свой инструмент. Более того, Flutter позволяет создавать не только мобильные приложения, но и приложения для Интернета и десктопов (поддержка десктопов Flutter доступна в виде бета-версии на стабильном канале).

Собрав все это вместе и учитывая, что ведущие компании, такие как Alibaba, уже используют Flutter – будущее выглядит многообещающим для инструментария.

Что касается React Native – Facebook в настоящее время сосредоточен на масштабной перестройке архитектуры технологии.

Команда делает все возможное, чтобы улучшить поддержку как для пользователей React Native, так и для более широкого сообщества. И, благодаря этому, сообщество теперь может легко предлагать изменения в основных функциях фреймворка с помощью процесса RFC, который использует выделенный репозиторий GitHub.

Фактическим результатом таких улучшений архитектуры является Hermes – движок JavaScript с открытым исходным кодом, оптимизированный для мобильных приложений, который улучшает время взаимодействия, а также снижает размер приложения и использование памяти.

Создание открытой среды для обсуждения React Native – это важный шаг, который является как признаком постоянного совершенствования, так и сигналом к светлому будущему технологии.

Учитывая, что React Native занимает такое стабильное положение на рынке и находится на траектории непрерывного развития, маловероятно, что в ближайшее время он потеряет свою актуальность.

Тем не менее, Flutter – внушительный конкурент React Native.

### **Библиографический список:**

1. Eric Windmill. Flutter in action. — Shelter Island: Manning Publications Co., 2020. — 368 с.
2. Frank Zammetti. Practical Flutter. — Pottstown: Apress, 2019. — 407 с.
3. Alessandro Biessek. Flutter for Beginners. — Birmingham: Packt Publishing Ltd., 2019. — 498 с.