React Native vs Swift: Что выбрать для разработки iOS?

Описание

Когда вы планируете разработать приложение для iOS, первый вопрос, который приходит вам в голову: “Использовать ли мне Swift для разработки нативного приложения или React Native для разработки кроссплатформенного приложения”? Вот правильный ответ – всегда лучше выбрать Swift для разработки iOS-приложений. Но все же в некоторых случаях React Native выигрывает конкуренцию в соответствии с вашими требованиями к платформе. TL;DR – Выбирая между React Native и Swift, выбирайте React Native, если приложение для iOS требует простого пользовательского интерфейса и простой логики кода, в то время как используйте Swift, если приложение для iOS требует различных функций и должно быть совместимо с каждым устройством Apple.

Более того, React Native предпочтительнее, если вы хотите быстрее развернуть систему и стремитесь снизить стоимость разработки, в то время как Swift – правильный выбор, если ваш бюджет не ограничен. И Swift, и React Native способны эффективно работать на устройствах iOS. Однако есть некоторые факторы и параметры, которые следует учитывать при создании приложения для iOS и выборе конкретного языка. Поэтому мы подробно описали технологии React Native и Swift, их плюсы и минусы, чтобы помочь вам выбрать наиболее подходящую для вашего проекта. Итак, давайте углубимся и разберемся в сравнении между Swift и React Native 2023.

Что такое React Native?

Поначалу каждый новичок путается между React Native и React. Однако, прежде чем выбрать язык, вы должны знать основную разницу между React и React Native. Если вы будете понимать это, вам будет легко выбрать React Native для своего приложения для iOS. Если говорить о React Native, то это JavaScript-фреймворк, созданный и поддерживаемый технологическим гигантом – Facebook. Язык был изобретен для создания кроссплатформенных мобильных приложений и ускорения процессов мобильной разработки. React Native использует JavaScript, а это самый полезный язык в мире – впереди Swift. JavaScript предлагает функцию повторного использования кода, позволяя разработчикам разделять до 70% кода между различными платформами.

В результате это поможет вам сэкономить время разработки и снизить стоимость кроссплатформенных приложений. С помощью разработки приложений React Native разработчики смогут писать код как для платформ Android, так и для iOS. Затем элементы пользовательского интерфейса отображаются в их родных версиях. Это означает, что приложение React Native для iOS использует тот же нативный пользовательский интерфейс, что и любое приложение для iOS. Аналогичный сценарий для версии Android. Давайте перейдем к следующему сегменту – преимуществам и недостаткам React Native, а затем рассмотрим основные различия между React Native и Swift.

Ведущие приложения, разработанные с помощью React Native

У вас наверняка возник вопрос, сколько стоит создание приложения React Native? Чтобы получить точное представление, вы можете обратиться к нашему блогу. Я уверен, что стоимость разработки будет стоить инвестиций вашего бизнеса.

Тем не менее, вот ведущие приложения, которые были созданы с использованием React Native:

- Facebook
- Walmart
- Bloomberg
- SoundCloud
- Instagram

Преимущества React Native

Мы должны рассмотреть светлые и темные стороны React Native, прежде чем принять решение о выборе React Native или Swift. Давайте посмотрим, какие плюсы может предложить язык, если вы нанимаете разработчиков React Native.

Горячая перезагрузка

Функция горячей перезагрузки в React Native позволяет разработчикам визуализировать свой код. Таким образом, это делает весь процесс разработки продуктивным и более простым.

Безупречная производительность

Поскольку нативные приложения разрабатываются для конкретной ОС, они используют все преимущества скорости обработки данных устройства. Приложения имеют хорошую, безошибочную производительность и просты в использовании. Кроме того, для повышения производительности необходимо обеспечить интерактивный пользовательский интерфейс. ReactJS может быть наиболее подходящим, когда речь идет о пользовательском интерфейсе. Есть вероятность, что будет сложно уследить за всем и управлять аспектами пользовательского интерфейса, поэтому, чтобы не отставать, вы можете положиться на услуги по разработке React UI/UX.

Богатая поддержка библиотек

React Native имеет около 33 000 библиотек и пакетов, в то время как Swift имеет только 3800 пакетов. Фактические цифры показывают, что React Native значительно превосходит по количеству библиотек.

Легко интегрировать

Нативные приложения легко интегрировать с различными устройствами, такими

как GPS, камера и сенсорный экран. Более того, сложные функции, такие как VR, AI, IoT и AR, могут быть легко интегрированы в нативные приложения.

Меньше отладки

Разработчики могут легко отслеживать ошибки с меньшими усилиями.

Высокая надежность и безопасность

Нативные приложения безопасны и надежны, поскольку они соответствуют всем правилам операционной системы. Эти приложения проходят оценку и авторизацию в App Store. Следовательно, они очень безопасны и надежны.

Недостатки React Native

Несмотря на все замечательные возможности React Native, у него есть определенные недостатки. Однако сообщество React Native постоянно стремится устранить эти трудности.

Сложный дизайн

Как известно, обе платформы – iOS и Android – имеют свои собственные правила публикации приложений. Это становится большой проблемой для разработчиков при создании высококачественного пользовательского пользовательского интерфейса для приложений React Native.

Более длительный процесс тестирования

Если сравнивать React Native и Swift, то скорость разработки приложений на React Native выше благодаря единому источнику кода, однако процесс тестирования занимает больше времени. Когда дело доходит до тестирования, приложения React Native требуют больше усилий и точности, поскольку вероятность появления ошибок выше на обеих платформах.

Меньше обёрток

Разработчикам может понадобиться писать собственные обертки для интеграции нативных модулей, компонентов и API в проект, что отнимает много времени.

Что такое Swift?

Swift – это современный и компилируемый язык программирования, разработанный и поддерживаемый командой Apple и сообществом разработчиков с открытым исходным кодом для создания нативных iOS-приложений для каждого устройства Apple. Выпущенный в 2014 году, Swift считается мультипарадигмальным языком общего назначения. Swift предназначен для разработки iOS-приложений для watchOS, macOS, tvOS и устройств на базе Apple. Основная цель Swift – привлечь больше внимания разработчиков к услугам по разработке iOS. Поскольку Swift предлагает лаконичный и выразительный синтаксис, интерфейс программирования является интерактивным. В первую очередь Swift был разработан для замены Objective-C, который был языком программирования Apple. Таким образом, следующая кодовая практика очень удобна и практически не имеет лазеек. Это лучший способ разработки приложений для iOS. Поэтому он завоевал большое внимание в сфере услуг по разработке мобильных приложений.

Ведущие приложения, разработанные с использованием Swift

Некоторые популярные и известные брендовые приложения разработаны с использованием Swift. Мы составили список самых лучших приложений для лучшего понимания:

- Airbnb
- Twitter
- Lyft
- Центр тестирования
- SlideShare

Преимущества Swift

Знаете ли вы, почему компании переключили свое внимание на Swift при создании приложений для iOS? Давайте узнаем о его преимуществах в следующем разделе.

Лучшая обработка ошибок

Swift имеет надежную типизацию и сильную систему обработки ошибок. Она эффективно предотвращает ошибки и сбои кода в производственной среде.

Легко масштабируется

Swift – это язык программирования с заделом на будущее. Поэтому разработчики могут легко добавить любую новую функциональность. В результате эти приложения обычно легче масштабировать по сравнению с React Native.

Повышенная безопасность и производительность

Повышение производительности и безопасности являются основными факторами Swift, поскольку он был создан для замены языка Objective-C. Во время его первоначального выпуска было заявлено, что его производительность на 40% выше.

Пользовательский опыт

Приложения на Swift занимают меньше времени на установку и потребляют меньше памяти на устройстве, поскольку они легковесны. Язык также был создан с учетом особенностей разработки под iOS, что дает разработчикам больше возможностей для работы с родным устройством. В результате приложения на Swift выглядят и работают лучше.

Сообщество разработчиков с открытым исходным кодом

У Swift есть активное сообщество разработчиков с открытым исходным кодом, которое поддерживает платформу. Кроме того, ее очень легко изучить и понять новичкам.

Недостатки Swift

Существуют ли какие-либо основные недостатки Swift, о которых вы должны знать? Давайте разберемся в его минусах в данном разделе.

Ограниченные таланты

Несмотря на быстрый рост Swift, он все еще имеет небольшой пул разработчиков по сравнению с другими языками программирования. Нанять разработчика Swift может быть непросто.

Только для устройств Apple

Swift можно использовать только на “родных” устройствах, поскольку это “родная” платформа. В результате, если вы хотите создавать приложения как для Android, так и для iOS, вам придется создавать два разных приложения.

Подробное сравнение: Swift vs React Native

Основное различие между React Native и Swift заключается в том, что React Native подходит для кроссплатформенной разработки, в то время как Swift – правильный выбор для разработки нативных приложений для iOS. Давайте рассмотрим приведенные ниже параметры, чтобы сравнить Swift и React Native и решить, какой из них подходит для разработки приложений для iOS.

React Native vs Swift: Производительность

Производительность React Native и Swift зависит от множества факторов, таких как скорость GPU, потребление CPU, использование памяти и многое другое. Если говорить о тестах, то React Native выигрывает у Swift по использованию памяти и скорости GPU, в то время как Swift опережает React Native по потреблению CPU. Кроме того, разработчики могут легко встраивать нативные коды в приложения React Native и использовать различные инструменты для решения задач. Таким образом, с точки зрения производительности React Native выигрывает конкуренцию у Swift.

Победитель: React Native

Swift против React Native: Пользовательский интерфейс

Используя Swift, разработчики могут легко создавать удивительные пользовательские интерфейсы и визуально привлекательные приложения для платформ iOS. Более того, разработчики Swift могут использовать SwiftUI с нуля, чтобы обеспечить полностью “родной” и лучший пользовательский опыт. Если говорить о React Native, то он использует библиотеки JavaScript для создания пользовательских интерфейсов. Разработчики могут легко создавать ориентированные на платформу версии компонентов для нативного опыта. С помощью JavaScript и декларативного пользовательского интерфейса компоненты

React оборачивают нативный код и взаимодействуют с родными API. Таким образом, с помощью JavaScript очень легко создавать удивительные пользовательские интерфейсы, но очень сложно следить за изменениями и усовершенствованиями, которые происходят в нативной среде.

Победитель: React Native

React Native vs Swift: Стабильность

Swift обладает всеми возможностями платформы для создания нативных приложений. Более того, Swift выигрывает у React Native в игре Swift vs React Native, когда дело доходит до обработки тяжелых задач и визуальных эффектов. Таким образом, Swift обеспечивает большую стабильность платформы iOS. Несмотря на то, что React Native является лидером среди кроссплатформенных фреймворков на рынке, он не является предпочтительным для создания нативных приложений. Основное внимание в React Native уделяется использованию внутренних API и библиотек для бесперебойной работы приложений. Это означает, что в процессе разработки может быть много слоев, что делает приложение немного нестабильным.

Победитель: Swift

React Native против Swift: Скорость кодирования

Поскольку JavaScript поддерживает React Native, этот язык существует на рынке уже много лет. Он становится популярным благодаря простоте изучения и поддержке различных инструментов. Таким образом, разработчикам очень легко изучать и создавать кодовую базу на React Native по сравнению со Swift. С другой стороны, разработчики Swift могут условно создавать нативные приложения для iOS. Поскольку Swift появился, чтобы преодолеть наследие Objective C, он сделал процессы более плавными, устранив недостатки.

Победитель: React Native

Swift vs React Native: Кривая обучения

Если говорить о React Native для разработки приложений для iOS, то разработчики считают этот процесс довольно простым и легким. Но здесь процесс веб-разработки отличается от процесса разработки приложений. Поэтому разработчики не

чувствуют себя как дома, работая с React Native. Однако React Native имеет различные подробные документы и обширные библиотеки, которые могут помочь разработчикам легко освоить язык. С другой стороны, Swift имеет очень сложную кривую обучения. Однако компания Apple выпустила официальный документ, который поможет разработчикам легко понять язык, его основы и особенности. Кроме того, в Swift есть замечательная функция Swift Playgrounds, которая делает кривую обучения плавной даже для профессионалов с нулевым техническим опытом.

Победитель: Swift

React Native против Swift: Зрелость платформы

Если говорить о зрелости платформы для React Native и Swift, то они оба довольно молоды в индустрии. React Native был выпущен в 2015 году, а Swift – в 2014. Тем не менее, оба сообщества приветствуют повышение надежности с каждым обновлением версии. Как и язык Swift, React Native не является полностью родным языком для разработки приложений для iOS, несмотря на то, что это нативный конструктор приложений. React Native функционирует как мост между платформами iOS и кодом. Когда речь идет об интенсивной вычислительной и графической работе, Swift показывает превосходные результаты.

Победитель: React Native

Swift против React Native: Стоимость услуг разработчика

Если вы планируете разработать приложение для iOS, нанять разработчика React Native будет очень дешево. Хотя вы не найдете большой разницы в найме одного программиста для языков программирования Swift и React Native. Позже, когда вы наймете команду разработчиков, они станут экономически эффективными и более дешевыми. Кроме того, React Native не снижает качество проекта, а также является экономически эффективным. В результате в долгосрочной перспективе это становится беспроигрышной ситуацией для организаций.

Победитель: React Native

React Native vs Swift: Доступность для разработчиков

Поскольку язык JavaScript очень популярен в сообществе разработчиков, найти

JavaScript-разработчиков очень легко. Кроме того, Swift является сравнительно новым языком. Таким образом, шансов найти команду профессиональных Swift-разработчиков или компании по разработке Swift меньше. По данным опроса Stack Overflow 2022, около 65,36% разработчиков считают JavaScript предпочтительным языком программирования. Эти данные доказывают популярность React Native по сравнению со Swift.

Победитель: React Native

React Native против Swift: Поддержка сообщества и документация

Как мы знаем, React Native – это язык программирования с открытым исходным кодом и поддерживается многими профессиональными разработчиками по всему миру, в то время как разработчики Apple поддерживают язык Swift. Но разработчики Swift могут получить техническую поддержку от общественных платформ. React Native имеет глубокую, краткую и точную документацию, а Swift имеет постоянное обновление документации, что также полезно.

Победитель: React Native

React Native против Swift: Краткое резюме

Параметры	React Native	Swift
Разработано	Facebook	Apple Inc
Тип	Программная основа пользовательского интерфейса с открытым исходным кодом	Язык программирования общего назначения, мультипарадигмальный, компилируемый язык программирования
Первоначальный выпуск	2015	2014

Разработка пользовательского интерфейса	Нативныекомпоненты, обернутые с использованием API, кросс-платформенная поддержка	Нативныепользовательские интерфейсы iOS с использованием SwiftUI
Поддерживаемые платформы	Android, Android TV, iOS, macOS, Web, Windows, UWP	Каждая ОС Apple
Сообщество	Лучший выбор для кросс-платформенной разработки с большой поддержкой сообщества	Относительно небольшое сообщество, будучи проектом с открытым исходным кодом при поддержке Apple Inc
Производительность	Производительность, подобная нативной, с высокой скоростью, плавной навигацией и лучшим пользовательским опытом	Более быстрая и высокая производительность при запуске сложных приложений, поскольку она использует возможности устройства и обеспечивает лучшие результаты при работе с приложениями, чувствительными к графике и тяжелым вычислениям
Пригодность	Кроссплатформенные приложения, приложения для стартапов и малого и среднего бизнеса	По-настоящему нативное приложение для iOS, использующее возможности платформы и требующее серьезных вычислений и графики
Использование	Наиболее широко используемые	Меньше используется
Разработка приложений	Быстрее	Медленнее

Язык и синтаксис	Фреймворк на основе JavaScript, JSX Syntax – расширение синтаксиса для JavaScript	Альтернатива Objective-C, лаконичный и действительно выразительный синтаксис
Стоимость разработки	Относительно низкий	Высокий
Документация	Великолепно и элегантно	Великолепно и элегантно
Используется	Facebook, Flipkart, Shopify, Skype, Instagram, Wix, Walmart, Tesla, Discord, Tencent и многие другие.	Khan Academy, Eventbrite, LinkedIn, Airbnb, Sky Guide и т.д.

Когда следует использовать React Native?

Теперь мы рассмотрели все стороны React Native. Если говорить кратко, то React Native можно выбрать, когда:

- Вам нужно создать приложения для iOS и Android в рамках заданного бюджета.
- Приложение требует горячей перезагрузки для ускорения разработки.
- Вам нужно создавать кроссплатформенные приложения.

Когда следует использовать Swift?

Вы можете использовать Swift в качестве основного языка программирования, когда:

- Вам нужно разработать приложение только для платформы iOS.
- Приложение требует большого объема памяти и сложного пользовательского интерфейса.
- Приложению нужны нативные функции iOS и API.

Окончательный вердикт

Наконец, эту битву между Swift и React Native можно рассматривать как кроссплатформенную разработку против нативной. Но здесь мы рассматриваем платформу iOS в качестве контекста. Таким образом, оба языка способны позволить вам разрабатывать iOS-приложения с удивительными возможностями. Одним словом, выбирайте React Native, если вы хотите разработать кроссплатформенное приложение, и переходите на Swift, если приложение требует возможностей нативной платформы. Итак, какой язык вы выберете для разработки приложений для iOS? Сообщите нам о своих мыслях.

Дата Создания
06.06.2023