

Бизнес-руководство по технологическому стеку разработки мобильных приложений на 2023 год

04.04.2023

В App Store и Google Play Store насчитывается более 5,7 млн мобильных приложений. Глобальный доход от мобильных приложений достиг \$133 к 2021 году и, по оценкам, вырастет до \$170 к концу 2022 года.

Год	Общий доход от приложений (в миллиардах долларов США)	IOS Доходы от приложений	Google Play Доходы от приложений
2016	\$43.50	\$28.60	\$15
2017	\$58.10	\$38.50	\$21.20
2018	\$71.30	\$46.60	\$24.80
2019	\$89	\$58.40	\$30.60
2020	\$111	\$72.30	\$38.60
2021	\$133	\$85.10	\$47.90

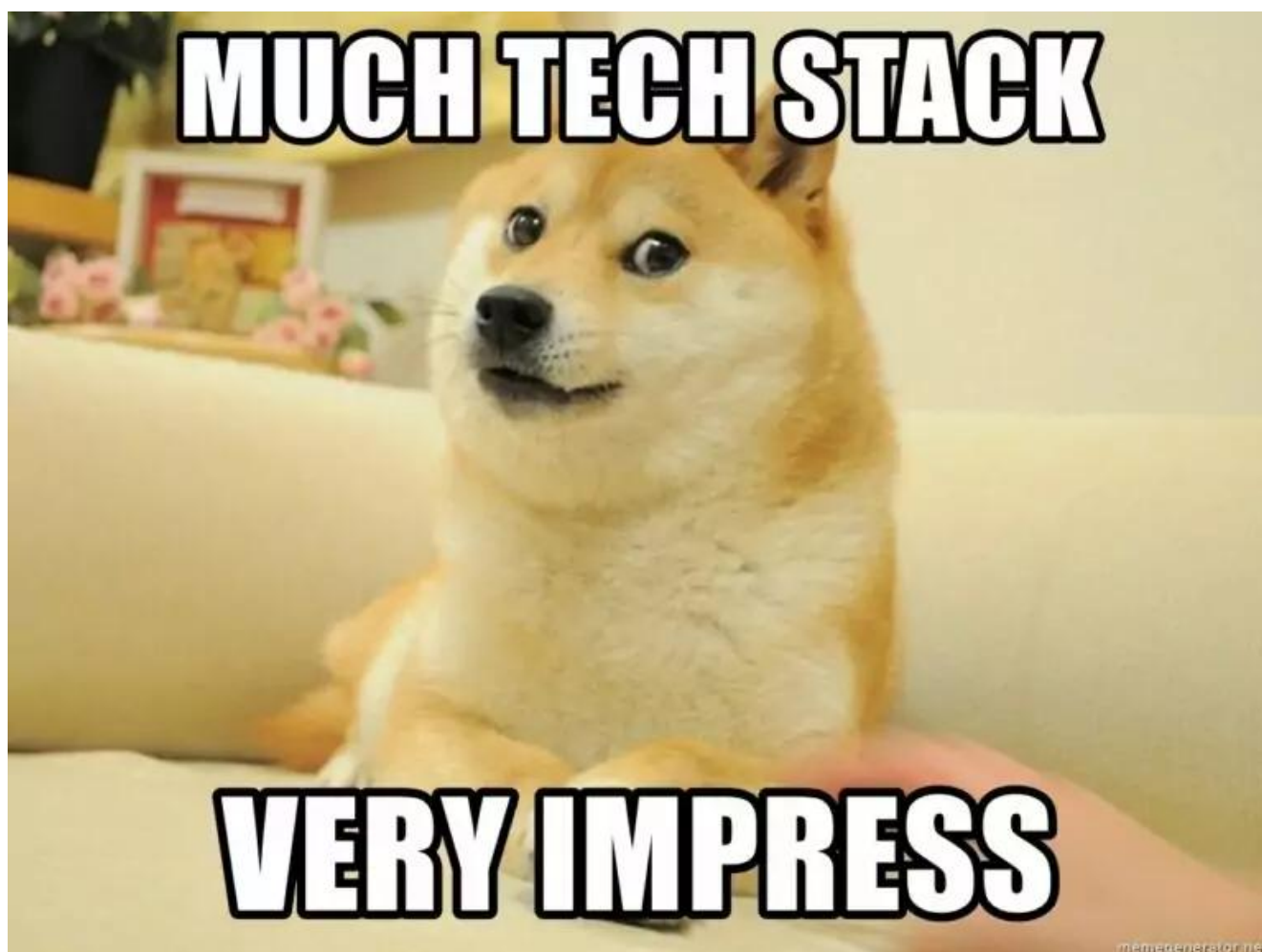
Очевидно, что рынок растет. Кроме того, уровень интернет-соединения достигает 5G и выше, а качество и мощность мобильных устройств улучшается год от года. Но меняются и требования пользователей. Хорошей идеи для приложения сегодня недостаточно. Оно должно быть быстрым, интуитивно понятным и отзывчивым, регулярно и быстро обновляться и не иметь ошибок.

Как этого добиться? Первым шагом должен стать выбор правильного технологического стека.

Тема выбора технологического стека для разработки приложений не так популярна в бизнес-руководствах. Обычно речь идет о

маркетинге и пользовательском опыте, а не о технологиях, которые стоят за приложением.

Поэтому в этой статье мы хотим подчеркнуть важность этого недооцененного решения. Потому что, если вы выберете неправильный стек, вы можете столкнуться с проблемами в процессе разработки и будущего обновления, не говоря уже о финансовых потерях и разрушении репутации.



Итак, каковы потенциальные потери, финансовые и другие, при выборе неправильного стека? Если вы не хотите переделывать свое приложение в будущем, вам необходимо наше бизнес-руководство по выбору технологического стека для разработки приложений.

Ключевые особенности:

- Выбор правильного технологического стека для разработки мобильных приложений гарантирует, что у вас будет

быстрое приложение с первоклассным UX, что приведет к увеличению числа клиентов и повышению ROI, что равнозначно ценности продукта.

- Наивный технологический стек для приложений Android обычно включает языки программирования Java и Kotlin, а также такие наборы инструментов, как Android Studio и Android Developer Services.
- Нативный технический стек приложений для iOS обычно включает языки программирования Objective-C или Swift, а также наборы инструментов AppCode и Xcode.
- Кроссплатформенные решения обеспечивают большую гибкость и позволяют работать с меньшим бюджетом. Технологический стек для такого приложения включает в себя React Native, Flutter и Xamarin.
- Еще один бюджетный вариант – гибридные приложения, которые имеют больше ограничений, но все еще широко используются. Технологический стек для них включает Cordova и Ionic.
- Чтобы выбрать подходящий набор технологий для мобильного приложения, необходимо учитывать масштаб бизнеса и потребности решения. Для MVP стартапа лучше всего подходит кроссплатформенный метод. Для уже успешного бизнеса лучшим выбором может стать разработка нативных приложений, а для первого немасштабируемого MVP начинающего стартапа подойдет гибридная модель.
- Выбор технологического стека – непростая задача, поскольку необходимо учитывать десятки факторов. Вам может понадобиться профессиональная помощь квалифицированных технических специалистов с опытом работы в вашей отрасли.

**Как выбор правильного
технологического стека для**

разработки мобильных приложений влияет на ценность продукта?

Выбор правильного технологического стека для мобильных приложений подобен подбору кусочка в пазл. Потенциально вы можете вставить неправильный кусочек, но пазл будет выглядеть неполным, некачественным и неудовлетворительным.

Правильный набор поможет:

- **Убедитесь, что дизайн и производительность соответствуют вашей аудитории.** Людям нужны быстрые, интуитивно понятные, красивые и функциональные приложения. Отвечая их требованиям с помощью современного стека мобильных технологий, вы привлекаете больше потенциальных клиентов и создаете себе репутацию. Рынок довольно конкурентный, поэтому сделать “хорошо” недостаточно.
- **Сделайте свое приложение почти вечным.** Если вы выберете стек, который будет существовать, вам будет легче обновлять приложение, подстраивая его под новые требования рынка. Кроме того, не все системы можно перенести, если что-то случится, поэтому лучше сделать безопасный выбор заранее.
- **Увеличение дохода.** Люди готовы платить за качество и скорость. Высококласный UX обеспечит вам стабильный возврат инвестиций в течение длительного времени.

В этой статье мы рассмотрим, какой технологический стек лучше всего подходит для разработки мобильных приложений в соответствии с вашими требованиями. Мы сосредоточимся на бизнес-функциях, а не только на технических, поскольку эти два аспекта тесно связаны между собой.



Технологические стеки разработки нативных приложений

Разработка нативных приложений означает, что вы создаете приложение для конкретной ОС, например, Android или iOS. Этот подход считается самым надежным и безопасным с технической точки зрения, но с точки зрения бизнеса он имеет свои недостатки.

Во-первых, вам придется создавать два разных приложения, если вы хотите обслуживать пользователей телефонов iPhone и Android, что автоматически удваивает инвестиции. Кроме того,

когда дело доходит до добавления новых функций и обновления приложений, вам придется платить дважды. Таким образом, вам определенно придется нанимать разработчиков приложений для обеих программ.

Тем не менее, многие компании все еще выбирают этот метод, поскольку их целевая аудитория чаще использует одну платформу, чем другую.

Поэтому мы рассмотрим все виды технологического стека мобильных приложений, включая нативную разработку для Android и iOS.

Технологический стек приложений для Android

Когда речь заходит о технологическом стеке для разработки приложений для Android, лучше всего подходят: **Языки программирования – Java и Kotlin**. Java существует уже много веков и остается одним из самых популярных языков программирования, который делает ваши приложения пригодными для любого устройства с поддержкой Java. Он лучше всего подходит для разработки на уровне предприятия.

Kotlin – кроссплатформенный язык, который был запущен как язык для использования в Android. Google рекомендует его как один из лучших вариантов технологического стека для приложений Android, поскольку он безопасен и даже сильнее, чем Java.

Набор инструментов – Android Studio и ADT (Android Developer Tools)

Android Studio – это среда, которая ускоряет разработку приложений для всех устройств, поддерживающих эту ОС. Она повышает скорость и гибкость проектирования благодаря встроенным шаблонам и интеллектуальному редактору кода.

Android Developer Tools – это компиляция API, библиотек, эмулятора, примеров и отладчика. Он похож на Studio, но также

обеспечивает поддержку автоматизации тестирования, а AS предоставляет место для всего процесса разработки.

Пользовательский интерфейс – Android UI или Jetpack Compose

Android UI полон стильных и функциональных шаблонов, которые сэкономят время и деньги. Jetpack Compose – более инновационное решение, в котором разработчики могут описывать элементы алгоритму, чтобы сделать их динамичными.

Технологический стек приложения для iOS

Технологический стек приложения для iOS часто выглядит следующим образом:

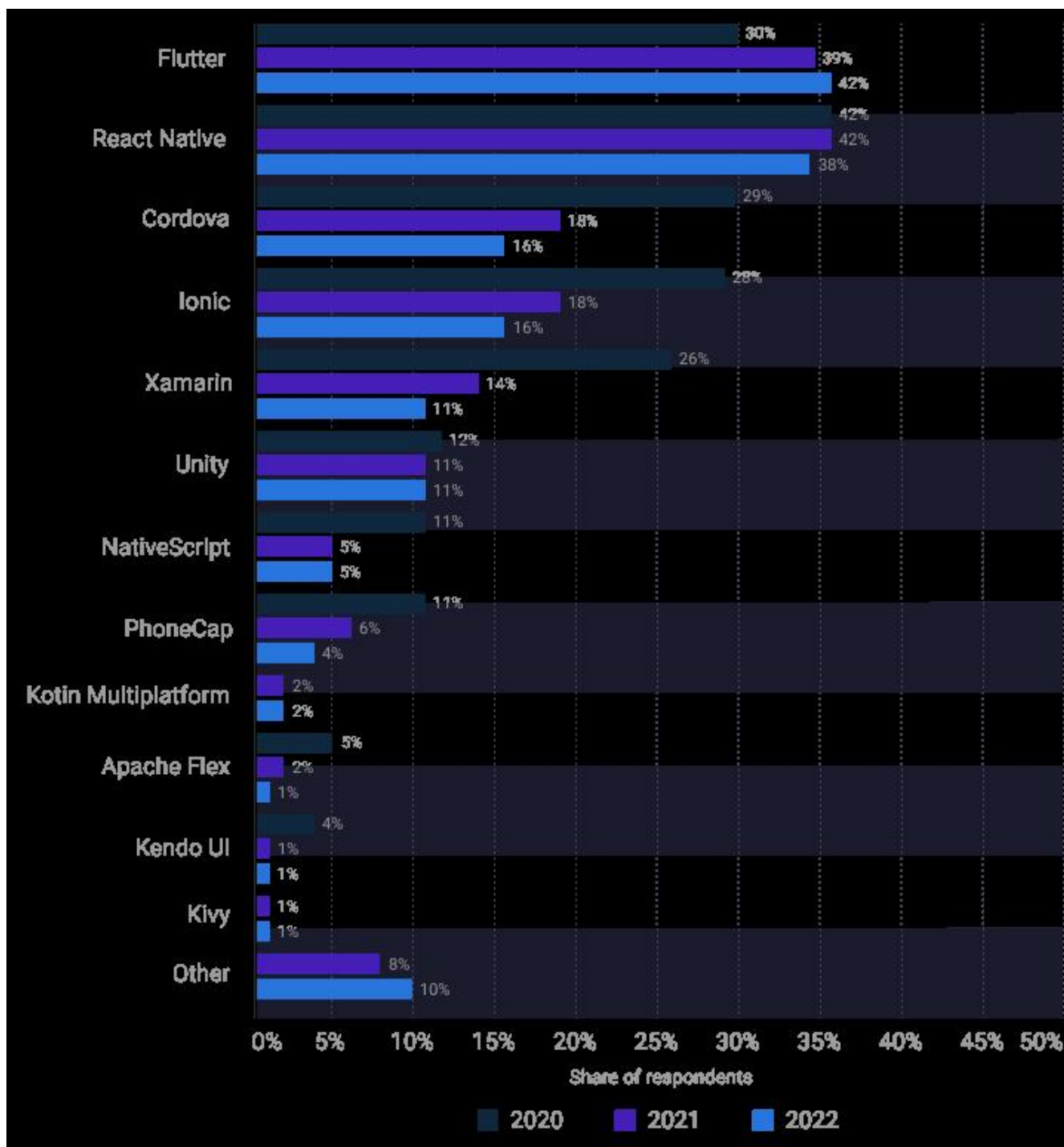
- **Языки программирования** – Objective-C и Swift. Objective-C – это очень гибкий язык для мобильных приложений OS X и iOS. Его можно использовать для создания новых приложений или модернизации существующих. Это проверенный язык с многофункциональными библиотеками сторонних разработчиков и отличной совместимостью с языком типа C. Swift был разработан как замена языкам программирования на основе C. Он быстрый, эффективный и хорошо читаемый, иногда его даже сравнивают с английским благодаря интуитивно понятной конструкции. Он более безопасен с точки зрения ошибок и имеет растущее сообщество.
- **Набор инструментов** – AppCode и Xcode. iOS SDK включает в себя симулятор iPhone, различные фреймворки, API, IDE Xcode и другие инструменты. В сочетании с Xcode этот набор позволяет разработчикам создать технологический стек для разработки приложений для iOS, включающий официальные языки Swift и Objective-C. AppCode – это сторонний набор инструментов с открытым исходным кодом, аналогичный Xcode; он имеет поддержку сообщества, но все же во многом полагается на Xcode.

- **UI – SwiftUI и UIKit.** SwiftUI – это очень свежий фреймворк, запущенный в 2019 году. Он предлагает более быстрый и продуктивный способ создания полноценных пользовательских интерфейсов. UIKit – основной фреймворк на данный момент, с шаблонами, которые помогают сэкономить время и сделать приложения для iOS более привлекательными.

Но что если вы не хотите создавать два отдельных приложения, но при этом хотите охватить оба рынка (пользователей iOS и Android)? Тогда есть два пути: кроссплатформенная и гибридная разработка. Мы рассмотрим оба варианта.

Технологический стек для кросс-платформенных приложений

Кросс-платформенная разработка предлагает вам создать приложение, которое будет работать на нескольких ОС. Мы часто рекомендуем такой технологический стек для разработки приложений, потому что это гибко, экономит время и экономически эффективно.



Когда вам нужно быстро создать универсальное приложение, кроссплатформенность – это выход. Кроме того, пользовательский опыт и уровень безопасности при таком подходе почти такие же высокие, как при нативной разработке, а время и затраты на создание сокращаются вдвое.

Технологии, которые мы рассмотрим ниже, также подходят для интеграции новых технологий, включая ИИ и блокчейн, которые уже помогают многим проектам привлекать больше клиентов. Блокчейн можно использовать для безопасных сделок, обмена

документацией и т.д., а ИИ уже несколько лет используется в качестве элемента умных чатов поддержки клиентов.

Flutter

Flutter – это SDK от Google, созданный для:

- Android
- Google Fuchsia
- iOS и Mac
- Linux
- Windows
- Веб-разработка

Таким образом, включив Flutter в свой стек технологий мобильной разработки, вы сможете использовать один код для любой из этих платформ. Это также оптимальный выбор, если вы хотите сократить время выхода на рынок, поскольку процесс сборки происходит быстро и надежно, а также исправление ошибок и обнаружение новых функций.

Xamarin

Платформа, разработанная Microsoft, предлагает нативные приложения для:

- Android
- iOS
- Windows

Фреймворк использует .NET и C#, при этом 96% его кода является многоразовым. Он допускает как MVVM, так и MVC дизайн и удобен для отладки. Кроме того, он активно поддерживается сообществом, что означает, что это стабильная современная платформа.

React Native

React Native – это JavaScript-фреймворк, используемый для создания решений для:

- Android
- iOS
- Web
- Universal Windows Platform (UWP)

Приложения, построенные с использованием фреймворка, работают с нативными API, что позволяет создавать выдающуюся функциональность для мобильных и веб-сервисов. React Native используется, когда проекту необходим интуитивно понятный пользовательский интерфейс и высокая отзывчивость.

Фреймворк имеет большое сообщество, а это значит, что у ваших разработчиков всегда будет поддержка, и вам не составит труда масштабировать и модернизировать ваше будущее приложение, повышая производительность и привлекая больше клиентов.

Существует множество предприятий с удивительными, амбициозными идеями и ограниченным бюджетом.

И кроссплатформенные технологические стеки для разработки приложений – отличный выбор для них благодаря относительно низкой стоимости и высокой адаптивности. Но есть еще один способ донести до аудитории нативное приложение без необходимости его создавать.

Технологический стек для гибридных приложений

Разработка гибридных приложений подразумевает сочетание веб- и нативных приложений. Такой подход позволяет создать веб-решение и скрыть его в случае нативного приложения. Оно открывается в браузере внутри последнего, который может быть

WebView в случае Android и WKWebView в случае iOS.

Это более экономичный вариант, чем создание двух нативных приложений, а также экономия времени. Однако в этом случае может быть сложно настроить производительность и создать качественный UX, который бы соответствовал нативному решению.

Технологический стек для мобильного приложения, использующего гибридную модель, выглядит следующим образом.

Apache Cordova

Apache Cordova – это фреймворк, который использует наиболее известные языки, такие как:

- HTML
- CSS
- JavaScript

Особенностью Cordova является его способность заключать написанный код в нативную оболочку для различных платформ, что означает, что один код может работать на нескольких ОС. Фреймворк содержит плагины, которые позволяют инженерам встраивать функции, обеспечивающие доступ к датчикам, камерам, файловым системам, GPS и другим функциям и хранилищам устройств, на которых работает приложение.

Ionic

Ionic – это SDK с открытым исходным кодом для создания гибридных и кроссплатформенных приложений для мобильных, веб- и настольных компьютеров. Он использует AngularJS в качестве основы для основных функций, что позволяет быстро и легко добавлять такие важные функции, как push-уведомления и аналитика.

Набор можно интегрировать с фреймворками Angular, Vue и React и использовать CLI (интерфейс командной строки), созданный с

помощью Node.js.



Дополнительные технологии для вашего мобильного приложения

Вот некоторые дополнительные технологии, которые могут вам понадобиться:

- Вам понадобится база данных для хранения данных зарегистрированных пользователей сервиса и поддержания правильной сборки. Вы можете использовать Mongo или MySQL, которые являются наиболее популярными базами данных на данный момент.
- Вы также можете выбрать облачные вычисления и хранение данных, для чего можно использовать Amazon Web Server (AWS) или Google Cloud.
- Еще одним важным аспектом является программное обеспечение для анализа аварий, которое поможет вашей команде разработчиков выявить ошибки и другие проблемы приложения и устранить их до запуска. Некоторые отличные

варианты для анализа – Firebase Crashlytics и Firebase Analytics, Google Analytics, Amplitude Analytics & Crash Analysis (для App Store и других платформ, кроме Google Play).

- Платежные платформы, такие как PayPal, Stripe и GPay, будут необходимы, если вы хотите монетизировать приложение через подписки, eShop и т.д.
- Интеграции для социального входа, такие как Google, Apple и Facebook, должны быть в вашем наборе.
- Интеграции для карт – Google Maps.

Хотя основной стек состоит из фронт-энда, бэк-энда, базы данных и других компонентов, дополнительное программное обеспечение не менее важно, когда речь идет о создании привлекательного пользовательского опыта.

Итак, вы видите, что существуют различные замечательные технологии, которые могут обеспечить нативное приложение даже без создания его вашей командой. Но как правильно выбрать стек технологий для приложений Android и/или iOS?

Как выбрать стек мобильных технологий?

При выборе технологического стека для разработки приложений необходимо помнить о масштабах вашего бизнеса и реально оценивать требования и ресурсы, которыми вы располагаете. Нет необходимости использовать самые мощные, редкие технологии, если вам нужно немасштабируемое MVP с базовыми функциями.

А когда вы создаете масштабное приложение с возможностями масштабирования, самый дешевый стек мобильных технологий может привести лишь к проблемам с привлечением клиентов из-за низкой производительности.

MVP стартапа

Для MVP стартапа мы рекомендуем кросс-платформенную разработку. Таким образом, вы сэкономите время и средства, запустив высококачественное базовое приложение с возможностью расширения.

Фреймворки React Native и Flutter являются достойными вариантами, поскольку они легко масштабируются в дальнейшем и могут предоставить функции нативного приложения за половину времени и стоимости. Пул талантов огромен и надежен, если вы выберете правильные каналы.

Тем не менее, вы должны принять решение после консультации с профессионалами о потребностях вашего проекта, его планах на будущее, текущих бизнес-требованиях и многом другом.

Мобильное приложение для успешного бизнеса

Если у вас успешный бизнес и вы хотите перейти на цифровые технологии, вам подойдут нативные или кроссплатформенные технологические стеки. Окончательный выбор зависит от бюджета и функций, которые вам нужны. Если вам нужно быстрое приложение с первоклассным UX и множеством сложных функций, выбирайте native. Если же вам нужен экономичный проект, который можно запустить как можно скорее, смирившись с немного меньшей скоростью, кроссплатформенный вариант – то, что нужно.

Нативные приложения имеют множество преимуществ, включая:

- Более высокая скорость за счет меньшего количества программных слоев, необходимых для работы приложения.
- Они работают в автономном режиме, за исключением специфических онлайн функций
- Они наиболее безопасны.

Однако вам придется разрабатывать два отдельных решения, что ударит по вашему бюджету.

Следующим лучшим вариантом является кроссплатформенная разработка, например, с помощью React Native. Это решение обладает такими преимуществами, как:

- Меньшая потребность в команде благодаря наличию одного стека для всех платформ
- Простая и быстрая разработка (а также время выхода на рынок)

React Native – прекрасный выбор, если вы хотите внедрить технологии IoT в свое приложение для повышения скорости, точности и безопасности.

Например, мы сотрудничали с компанией Roth River, используя React Native для создания приложения для контроля дистилляции бурбона. Фреймворк позволил интегрировать IoT для точности отслеживания данных.

Решение о выборе лучшего технологического стека для разработки мобильных приложений зависит от многих факторов, таких как масштабируемость приложения, ожидаемое время выхода на рынок и бюджет, стадия, на которой находится проект в данный момент, и многое другое.

Стартап на начальном этапе с немасштабируемым MVP

Основной проблемой для MVP стартапов на начальном этапе является стоимость разработки приложения. Самый дешевый вариант – гибридная разработка. Она немного более ограничена, чем кроссплатформенная, но если вам нужен очень базовый, но эффективный проект без планов будущего масштабирования, такой подход поможет вам сэкономить много денег.

Заработная плата гибридных разработчиков ниже, чем у двух

других методов, время разработки быстрое, а полученное приложение будет выглядеть так же, как и нативное, только работать будет немного медленнее и возможности будут ограничены.

Но для общих функций, которые обычно нужны MVP стартапа (а они зависят от отрасли, в которой вы работаете), гибридные инструменты работают просто отлично.

Часто задаваемые вопросы

Какие технологии используются для разработки мобильных приложений?

Наиболее популярными технологиями для разработки мобильных приложений в последние годы являются Ionic, Kotlin, Flutter, Xamarin. Хотя этот список не ограничивается ими, поскольку для различных целей, таких как нативные, кроссплатформенные и гибридные приложения, нам необходимо выбирать различные технологические стеки.

К таким технологиям относятся:

- React Native, Flutter и Xamarin — для кроссплатформенного программирования
- Swift, Objective-C — для мобильной разработки под iOS
- Kotlin, Java — для разработки под Android
- Cordova, Ionic — для гибридных приложений

Различные технологии используются в зависимости от ОС, для которой вы создаете приложение, типа разработки (нативная, кроссплатформенная, гибридная), вашего бюджета, времени выхода на рынок и многого другого. Для создания полноценного приложения вам также понадобятся наборы инструментов, технологии бэкенда, API и другие элементы.

Как выбрать стек разработки для моего приложения?

Учитывайте свои бизнес-цели, платформы, которые чаще всего использует ваша целевая аудитория, желаемое время выхода на рынок, бюджетные ограничения и согласуйте их с техническими требованиями вашего будущего приложения.

Какая платформа лучше для разработки мобильных приложений?

React Native и Flutter становятся все более популярными, очень удобными и высокофункциональными. Однако, объективно, лучшей платформы не существует. Каждый проект уникален и имеет особые требования, в соответствии с которыми мы можем выбрать наиболее подходящую платформу разработки. Ответив на вопросы о вашей целевой аудитории, уровне поддержки технологии, вашем бюджете, сроках, на которые вы рассчитываете в процессе программирования, и т.д., вы сможете приблизиться к пониманию того, что является лучшим выбором.

Заключение

Выбор технологического стека отвечает как минимум за треть вашего успеха. Ваше мобильное решение должно быть:

- Быстрым
- высокофункциональным
- Предлагать превосходный UX
- работать на всех смартфонах
- Экономически эффективным
- Быстрое создание и запуск
- Легко модернизировать и обновлять и т.д.

И все это отражается в технологии, которую использует ваша команда для создания приложения. Вы можете выбрать разработку

нативных приложений, когда у вас большой бюджет и время и вам нужны превосходные, самые быстрые приложения как для iOS, так и для Android.

Также есть возможность кроссплатформенной разработки с помощью Flutter или React Native. Это позволит вам сэкономить деньги и время без сильного ущерба для скорости и пользовательского опыта. Технологические стеки для кроссплатформы почти так же хороши, как и для нативной, но при этом код можно использовать повторно как для Android, так и для iOS.

И, наконец, вы можете выбрать гибридную разработку, если вы владелец стартапа с ограниченным бюджетом и не возлагаете все надежды на UX. При правильном построении гибридные приложения достойны и эффективны, хотя некоторые функции могут быть ограничены.