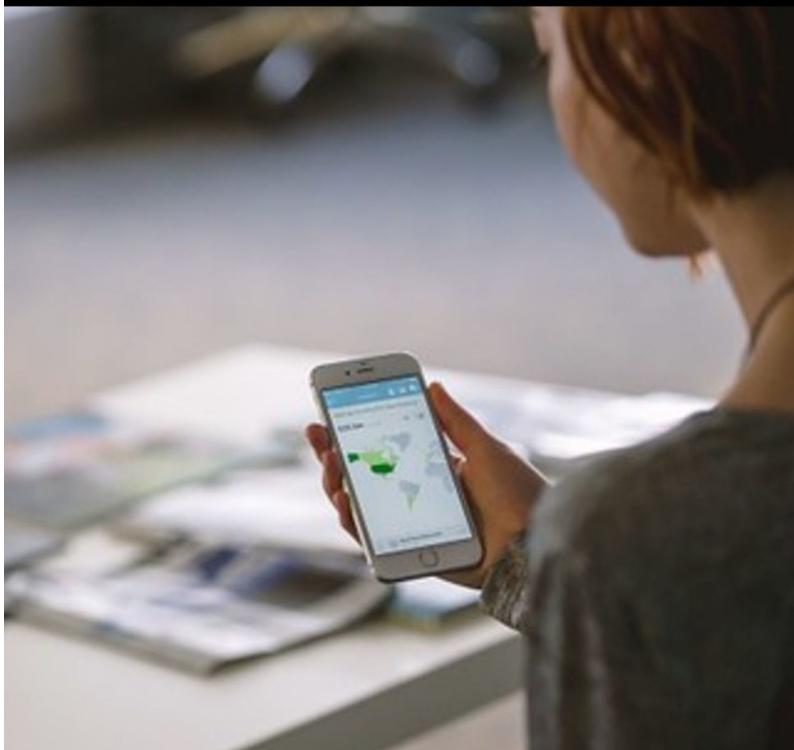


Маргарита Акулич

*Успешная разработка и запуск
мобильных приложений*



Маргарита Акулич

Успешная разработка и запуск мобильных приложений

*http://www.litres.ru/pages/biblio_book/?art=42756324
ISBN 9785449694003*

Аннотация

В данной книге пересказано о типах приложений, которые вы можете исследовать в рамках своего проекта по разработке мобильных приложений. Дано описание того, что нужно для разработки успешного мобильного приложения на современном рынке. Раскрыты общие причины неудачи мобильных приложений. Описаны шаги, которые надо пройти для обеспечения успешного запуска мобильного приложения.

Содержание

Предисловие	5
I Типы приложений. Веб против натива и гибрида	6
1.1 Приложения разных видов	8
1.2 Разработка веб-приложений. Как создаются веб-приложения? Что такое прогрессивные веб-приложения?	11
1.3 Преимущества и недостатки веб-приложений. Разработка собственных мобильных приложений	16
1.4 Преимущества и недостатки нативных приложений. Собственные (родные) приложения	19
1.5 Разработка приложений Xamarin. Нюансы, касающиеся разработки собственных приложений	24
Конец ознакомительного фрагмента.	28

Успешная разработка и запуск мобильных приложений

Маргарита Акулич

© Маргарита Акулич, 2020

ISBN 978-5-4496-9400-3

Создано в интеллектуальной издательской системе Ridero

Предисловие

В данной книге пересказано о типах приложений, которые вы можете исследовать в рамках своего проекта по разработке мобильных приложений.

Дано описание того, что нужно для разработки успешного мобильного приложения на современном рынке.

Раскрыты общие причины неудачи мобильных приложений.

Описаны шаги, которые надо пройти для обеспечения успешного запуска мобильного приложения.

Характер книги – образовательный.

Пожалуйста, обратите внимание на ранее опубликованные в системе Ridero книги М. Акулич «Мобильный маркетинг» и «Маркетинг мобильного приложения».

I Типы приложений. Веб против натива и гибрида



Существует три основных типа приложений, которые вы

можете исследовать в рамках своего проекта по разработке мобильных приложений: веб против нативных и гибридных. Но какой из них целесообразнее выбрать?

В зависимости от целей вашего бизнеса и общих целей, это решение может повлиять на успех вашей мобильной стратегии. Принимая решение о создании вашего приложения как веб-приложения, нативного или гибридного, необходимо учитывать множество факторов.

Далее рассмотрим приложения разных видов.

1.1 Приложения разных видов

Веб-приложения

Веб-приложения размещаются в веб-браузерах и представляют собой веб-сайты, которые выглядят как нативные приложения. Однако вместо установки приложения на домашний экран устройства, как в собственном приложении, пользователи взаимодействуют с приложением через веб-просмотр.

Этот тип приложения прост в сборке и в обслуживании. Он представляет собой недорогой вариант. Однако подобные приложения требуют наличия веб-браузера, работают намного медленнее, чем собственные приложения, и не могут использовать утилиты устройства. Веб-приложения также намного менее интерактивны и интуитивно понятны, чем нативные.

Нативные приложения



Нативные приложения созданы для конкретных платформ и написаны на языках, поддерживаемых платформой (например, Swift и Objective-C являются общими языками для приложений iOS, а Java или Kotlin являются общими для приложений Android).

Нативные приложения бывают быстрыми и отзывчивыми, распространяются в магазинах приложений, предлагают интуитивно понятный пользовательский ввод и вывод и не требуют подключения к Интернету.

В целом, нативные приложения предлагают лучший пользовательский интерфейс, но их разработка стоит дороже разработки других вариантов.

Гибридные приложения



Гибридные приложения – это комбинация нативных и веб-приложений.

Гибридное приложение состоит из двух частей: первая представляет собой внутренний код, а вторая представляет собой встроенную оболочку, которая загружается и загружает код с помощью веб-просмотра.

Гибридные приложения дешевле, чем нативные, не требуют браузера и могут использовать API-интерфейсы устройства. Однако они работают медленнее, чем собственные приложения, и их нельзя настраивать для отдельных платформ, таких как собственные приложения.

1.2 Разработка веб-приложений. Как создаются веб-приложения? Что такое прогрессивные веб-приложения?

Разработка веб-приложений



Чем веб-приложения отличаются от веб-сайта? Веб-сайт обычно предоставляет пользователям больше информации, чем обычно отображается на мобильном сайте, тогда как веб-приложение сокращает эту информацию для улучшения функциональности.

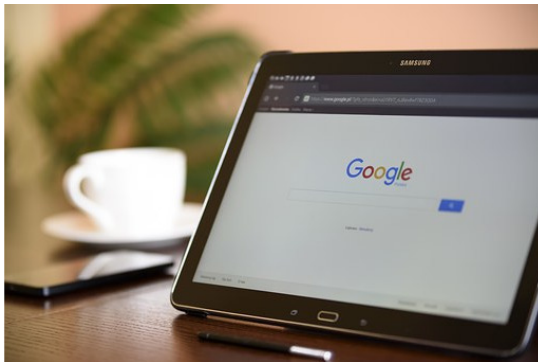
Веб-приложения загружаются в браузерах, таких как Chrome, Safari или Firefox, и их не нужно загружать из магазинов приложений, таких как магазины для мобильных приложений. Веб-приложения также не занимают места на устройстве пользователя.

Люди рассматривают веб-приложения как новую технологию, которая стирает границу между веб-приложениями и приложениями нативными и гибридными. Настолько, что иногда трудно определить, какие приложения являются веб-приложениями, какие гибридными, а какие – веб-сайтами с адаптивным дизайном.

Иногда то, что кажется нативным приложением, загруженным из магазина приложений, на самом деле является приложением в веб-представлении (которое написано на родном языке), содержащим URL-адрес веб-приложения. Большая часть кода обрабатывается за пределами родного языка. Для некоторых это удовлетворяет требованиям нативного приложения, а для других это просто еще один формат для веб-приложения.

Веб-приложения – это, по сути, веб-сайты, которые выглядят как собственные приложения, но вместо установки на домашнем экране пользователь может создать закладку для этой страницы.

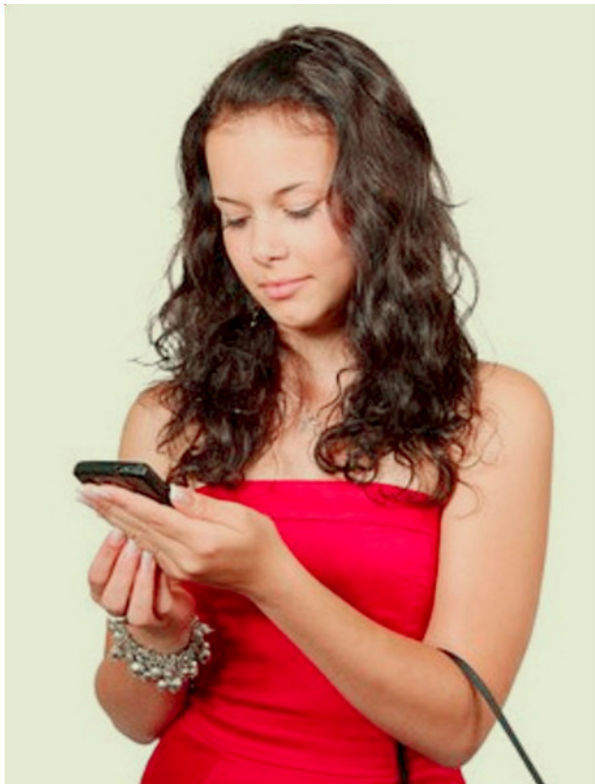
Как создаются веб-приложения?



Как правило, веб-приложения создаются на JavaScript, CSS и HTML5 и запускаются в браузере (Safari, Chrome и т. д.). Для разработчиков нет комплекта для разработки программного обеспечения; однако есть шаблоны, с которыми разработчики могут работать.

Если вы решите разработать веб-приложение, оно может быть простым и быстрым в создании. Хотя веб-приложения часто упрощены и не предоставляют те же функции, которые предоставляют нативные приложения.

Что такое прогрессивные веб-приложения?



Прогрессивные веб-приложения – это гибриды обычных веб-страниц и мобильных приложений. Веб-приложениям не хватает функциональности собственных приложений, таких как отправка push-уведомлений или работа в автономном режиме.

Браузеры и веб-приложения, однако, становятся все более продвинутыми. Теперь прогрессивные веб-приложения могут использовать функции, аналогичные собственным приложениям. Вот некоторые из функций:

Отправка push-уведомлений

Использование сенсорных жестов и акселерометра
вашего телефона

Использование некоторых аппаратных средств
устройства, таких как вибрация.

Несмотря на улучшения, предлагаемые прогрессивными веб-приложениями, огромным их недостатком является то, что они совместимы только с Google Chrome. Это означает, что пользователи iOS не могут использовать приложения такого типа. Учитывая, что пользователи iOS тратят больше всего денег на приложения, это может быть существенным недостатком (в зависимости от ваших бизнес-целей).

1.3 Преимущества и недостатки веб-приложений. Разработка собственных мобильных приложений



Преимущества веб-приложений

Преимущества веб-приложений:

Их легко построить. Они просты в обслуживании.
Они недорогие.

Вы можете создать одно приложение для всех платформ – iOS, Android и т. д., если оно работает в браузере.

Недостатки веб-приложений

Недостатки веб-приложений:

Для их запуска требуется браузер.

Пользователь должен сделать еще один шаг для ввода URL-адреса, что усложняет взаимодействие с пользователем.

Они гораздо медленнее, чем родные приложения. Они менее интерактивны и менее интуитивно понятны, чем родные приложения. Нет значка на рабочем столе мобильного устройства, как если бы он был загружен из магазинов приложений. Не удастся использовать утилиты устройства.

Разработка собственных мобильных приложений



Собственные мобильные приложения являются наиболее распространенным типом приложений. Они созданы для конкретных платформ и написаны на языках, которые поддерживает платформа, например, Swift и Objective-C для нативных приложений iOS и Java или Kotlin для нативных приложений Android. Собственные приложения также создаются с использованием специальной интегрированной среды разработки (IDE) для указанных операционных систем.

И Apple, и Google предоставляют разработчикам приложений собственные инструменты разработки, элементы интерфейса и SDK. Большинство компаний будут инвестировать в разработку собственных мобильных приложений из-за множества предлагаемых преимуществ, имеющихся в наличии, их намного больше по сравнению с числом преимуществ у других типов приложений.

1.4 Преимущества и недостатки нативных приложений.

Собственные (родные) приложения

Преимущества нативных приложений



Преимущества нативных приложений (приложений native) рассмотрены ниже.

Нативные приложения очень быстрые и отзывчивые, потому что они созданы для конкретной платформы. У них лучшая производительность. Они распространяются в магазинах приложений. Они более интерактивны, интуитивно

понятны и работают намного более плавно с точки зрения пользовательского ввода и вывода. Native позволяет разработчикам получить доступ к полному набору функций данной платформы. Подключение к интернету не требуется, хотя это зависит от функциональности. В целом лучший пользовательский опыт. Для пользователя процесс является более естественным, поскольку каждая платформа имеет определенные стандарты пользовательского интерфейса.

Недостатки нативных приложений

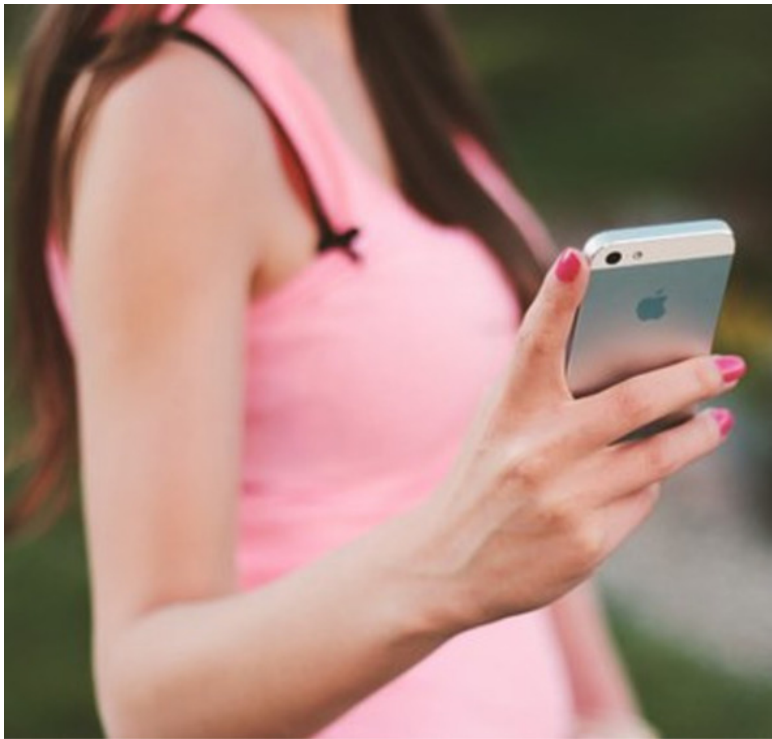
Недостатки нативных приложений :

Сложные языки программирования для изучения, а это значит, что вам нужны опытные разработчики.

Они более дорогие.

Они не являются лучшим вариантом, если необходимо очень простое приложение.

Собственные (родные) приложения



Помните, что родные приложения требуют, чтобы пользователи продолжали загружать обновления в целях улучшения доступности и удобства их работы, в то время как разработчики могут публиковать обновления для веб-приложений без какого-либо участия пользователя.

Хотя первоначальная стоимость может быть выше при

разработке собственных приложений, эта опция сэкономит ваше время и деньги в долгосрочной перспективе. Предлагая отличный пользовательский опыт, лучшую производительность и используя функции устройства, вы можете предложить пользователям более персонализированный опыт.

Многочисленные преимущества родных приложений по сравнению с нативным приводят к более высоким показателям конверсии и в конечном итоге повышают лояльность клиентов

1.5 Разработка приложений Xamarin. Нюансы, касающиеся разработки собственных приложений



Разработка приложений Xamarin

При использовании native вам необходимо создать приложение для каждой платформы, например, одно для iOS и одно для Android; однако у разработчиков есть новые технологии для работы. Две платформы, в частности, Xamarin и React Native, можно квалифицировать как другие типы разработки собственных приложений.

Xamarin создан Microsoft и позволяет разработчикам создавать одно приложение, которое работает на многих платформах с использованием C #. Используя общую кодовую базу на C #, разработчики могут использовать инструменты Xamarin для написания собственных приложений для iOS, Android и Windows с собственными пользовательскими интерфейсами и совместного использования кода на нескольких платформах.

Нюансы, касающиеся разработки собственных приложений



React Native построен на Facebook и позволяет разработчикам создавать настоящие нативные приложения для iOS и Android с одной кодовой базой. С React Native вы создаете мобильное приложение, идентичное приложению, созданному с использованием Objective-C или Java. Однако в React Native вы используете JavaScript и React.

В разработке мобильных приложений термин «*родной*» определяется как экосистема, которую Apple или Google, например, выбирают для разработки приложений для своих операционных систем. Технология Xamarin или React Native очень сложна и может быть квалифицирована как нативная или гибридная. В частности, Xamarin можно

рассматривать как платформу для разработки собственных и гибридных приложений, так как он создает собственные приложения для Android, iOS и Windows в C # с помощью Visual Studio или Xamarin Studio. Он также создает гибридные приложения для нескольких операционных систем, совместно использующих кодовую базу C #, IDE, язык и API. Однако, поскольку между вашим кодом и платформой есть еще один уровень, многие считают его гибридом.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «ЛитРес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на ЛитРес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.