**Разработка мобильных приложений**

**Типы мобильных приложений**

Познакомимся с видами приложений, которые подразделяются по определенным критериям. Очень часто при разработке приложения, программисты и заказчики формируют целый перечень пунктов в Договоре и техническом задании. Вид устройства и виды ОС, а также девайсов, с которыми будет взаимодействовать мобильное приложение является не менее важным условием, нежели прототипирование или разработка функционала.На практике можно разделить приложения для мобильных устройств на три типа.  
  
**Мобильные сайты, веб-приложения**  
  
Это самый распространенный тип приложений для мобильных устройств. Современные смартфоны в состоянии отобразить обычный сайт. Им доступно все то, что мы привыкли видеть в десктопных приложениях — поддержка HTML5 делает свое дело. Веб-приложения отлично подходят для стартапа: именно они позволяют получить большой результат за маленькие деньги и за небольшой срок. Еще один плюс мобильного сайта по сравнению с другими мобильными приложениями – это кроссплатформенность. Однако есть и минус, притом весомый: с ними достаточно сложно заработать.  
  
**Гибридные приложения**  
  
При таком подходе вы получаете доступ ко всем плюсам API операционной системы: приложение обрастает push-уведомлениями и другими приятными бонусами, кроме того, теперь ваш продукт можно размещать в магазинах. При этом основной контент все еще представляет собой платформонезависимую страничку с версткой, размещенную на сервере. Это позволяет вносить косметические изменения в продукт без выпуска новой версии: достаточно добавить изменения на сервер. Гибридные приложения – отличное решение для тех, кто начинает бизнес или хочет проверить свою идею, показать ее инвестору, друзьям.  
  
**Нативные приложения**  
  
Этот вид приложений самый ресурсоемкий, но вместе с этим он позволяет по максимуму использовать возможности, предлагаемые каждой конкретной операционной системой. Как следствие, нативные приложения выигрывают как по функционалу, так и по скорости работы у других типов мобильных приложений. Именно к такому подходу сейчас приходят те компании, которые делали комбинированные приложения. Например, Facebook начинала с комбинированного приложения: нативные контролы (переключатели, вкладки и так далее) и веб-страница в качестве контента. Несмотря на то, что это неплохое решение, проблемы с производительностью приводят к тому, что разработчики отходят от комбинации с вебом. Именно такие мобильные приложения используются банками для входа в личный кабинет пользователя и проведения операций. Здесь работает множество структур и система безопасности по праву занимает первое место.

##### **Процесс создания мобильного приложения**

**Анализ бизнес-модели и изучение конкурентов**

От успешности и целесообразности идеи зависит его дальнейший рост и перспектива. Если Вы решили разрабатывать мобильную игрушку, это не лучший выбор, так как конкуренция в игровой категории баснословная. Что касается анализа конкурентов, существуют отличные сервисы, позволяющие определить наиболее активных игроков рынка по соответствующим критериям. Есть, как платные зарубежные сервисы, так и бесплатные российские. Например, Appintop, модуль CPIera Spy, позволяющий при регистрации и предоставления аккаунта получить весьма интересную статистику.

Необходимо исследовать наиболее конкурентную нишу, в которой ежедневно идет борьба за ТОП в маркете, целевую аудиторию и ее платежеспособность,

понять, какими особенностями обладает Ваш продукт и что нового Вы можете дать пользователям.

**User Story**  
  
На этом этапе важно проработать все возможные сценарии, чтобы не было неприятных сюрпризов на более поздних этапах разработки.  
Важно понимать, что за каждым пунктом в вашем to-do листе скрывается огромный айсберг функционала. Старайтесь фрагментировать и конкретизировать задачи.  
  
**Проектирование и дизайн**  
  
После составления User Story начинается проектирование и разработка дизайна.  
  
На этом этапе используются прототипы, которые можно вешать на доску и стрелочками показывать, как будет происходит навигация.  
  
При разработке дизайна обязательно используются гайдлайны.  
Гайдлайн в общем понимании – это документ, который выпускает компания, и по которому дизайнеры и разработчики понимают принцип построения взаимодействия приложения с пользователем. Условно говоря, для iOS кнопки надо делать круглыми, а для Windows Phone – квадратными. Однако используются и внутренние гайдлайны для разработчиков. Таким образом результат работы дизайнера чаще всего состоит из макетов, гайдлайнов и нарезки графики.  
  
Макеты лучше всего подавать «перелинкованными», например с помощью ProtoTypr, чтобы была понятна логика переходов. Гайдлайны содержат в себе информацию об отступах, размерах, визуальных эффектах, механике анимации и пр. Этот этап можно пропустить, если в вашем проекте один дизайнер и один разработчик, сидящие рядом друг с другом. Третья часть результата — нарезка графики — должна содержать минимум необходимых графических ресурсов (заботимся о весе приложения), иметь версии для разных разрешений экранов.   
  
**Передача в разработку. Обсуждение и необходимые правки описания**  
После получения макетов, гайдлайна и нарезки, начинается работа разработчика. Передается в разработку все то, что придумали, и ожидается ранний результат. Это не значит, что работа над архитектурой и пользовательским интерфейсом закончена. Иногда у разработчиков появляются интересные идеи, которые вносят коррективы в изначальный план. Когда разработка завершена, наступает стадия тестирования.  
  
**Тестирование**  
  
Существует немалое количество способов протестировать приложение.  
В мобильной разработке тестировщик – это человек, вокруг которого одни телефоны. Чаще всего тестирование производится по тест-кейсам - если внедряется новая функция, по ее описанию составляется тест-план.  
Существуют сервисы, помогающие в тестировании. Например, HockeyApp – приложение, позволяющее раздавать продукт бета-тестерам. Можно написать в социальных сетях: «Ребята, у нас новое крутое приложение. Кто хочет попробовать?» Желающие получают билд, пользуются приложением, а сервис собирает статистику, составляет креш-репорт и отправляет все это Вам.  
Также есть сервисы, позволяющие протестировать приложение на разных операционных системах — например, все Android-прошивки версии 2.1 или 2.3. Вы отдаете приложение, сервис скриншотит весь путь, который вы задали, присылает картинки вам на почту, и вы проверяете, все ли в порядке.  
  
**Мониторинг**  
  
Итак, вы разработали, протестировали приложение, добавили его в магазин. Для отслеживания статистики скачиваний можно использовать сервис Distimo. Он показывает статистику по пользователям, которые приходят в магазин, чтобы скачать приложения, и агрегирует комментарии.  
  
Важно понимать, что люди более склонны оставлять негативные комментарии. Если у человека все хорошо, он чаще всего просто пользуется приложением, не комментируя. Поэтому имейте в виду, что комментарии – это не полная оценка вашей работы, скорее еще один баг-трекер.  
  
Изменить ситуацию может довольно распространенных «хак» — окно Rate Us. С предложением оставить положительный комментарий в store, а в случае проблем написать разработчику. Эффект достаточно сильный, главное — правильно продумать алгоритм показывания диалога пользователю.  
  
Помимо комментариев Distimo показывает количество скачиваний, заработанные деньги, а также откуда скачивают ваши приложения.  
  
Еще один интересный мониторинговый сервис – Flurry. Он помогает собирать клиентскую статистику. Flurry предоставляет отчет о том, что делает пользователь в вашем приложении: сколько раз он нажал на кнопку, сколько раз возвращался в приложение и более общие параметры — аудитория, география, пол, возраст и пр.  
  
Несмотря на большое количество сторонних сервисов, полезно иметь собственную статистику. Какими бы хорошими не были внешние источники, их нужно проверять.

**Что нужно помнить?**

* На каждой новой платформе пользователь ожидает увидеть богатое приложение. Он рассуждает следующим образом: «Я сидел на плохом Java-телефоне, при этом мог пользоваться аськой. Я купил новый телефон, захожу в Marketplace, а там нет ICQ? Вы чем там занимаетесь?» Пользователь не принимает в расчет того, что вам приходится делать приложение с нуля. Новая платформа – это новые девайсы, новая документация, новые ресурсы.
* Чем популярнее платформа, тем больше у вас конкурентов. На данный момент существует два store, на примере которых это отлично видно: AppStore и Google Play. Если у вас есть идея приложения, которое легко монетизируется или просто получит много скачиваний, вбейте ключевые слова в поиске и скорее всего вы обнаружите, что такое приложение существует. Чем популярнее платформа, тем больше конкурентов. В таких случаях надо тщательно изучать аналоги, смотреть статистику, пытаться понять, по каким параметрам существующие решения можно превзойти.
* Важно понимать, как пользователи выбирают приложения. Изначально человек не собирается покупать конкретное приложение, он просто смотрит список. Например, по запросу «бесплатная музыка». Иконка и первые две строчки описания – это то, что человек видит и оценивает в первую очередь. Если иконка приличная, можно покупать; соответственно, плохая иконка уменьшает количество скачиваний.
* Очень важно попасть в топ магазина приложений. Попасть в топ store – очень хорошо, закрепиться там – залог успеха. Зачем это нужно? Когда пользователь хочет что-нибудь скачать, он заходит в чарт и видит, какое приложение сейчас на первом месте по числу скачиваний. Очень важно туда попасть, потому что это своего рода замкнутый круг. Приложение попадает в чарт, его видят пользователи, они его скачивают, оно снова попадает на первое место, и дальше итерации продолжаются. Поэтому всеми силами добывайте скачки и рейтинг: просите мам, бабушек, соседей ставить приложению пятерки. Стоит сказать, что на рынке полно решений для гарантированного вывода программы в топ. Однако органических пользователей это приносит мало, что не мешает продолжать эксперименты.
* Помните, что время публикации может доходить до нескольких недель. Допустим, вы разработали и протестировали приложение, обзвонили всех блоггеров или СМИ и сказали: «У меня новое приложение, приходите на пресс-конференцию». Вы его отправляете в AppStore и вынуждены ждать семь рабочих дней в России. В этом случае Google Play – рай для оперативных обновлений, где публикация занимает несколько часов.
* Фрагментация операционной системы. Если вы разрабатываете приложения под Android, учитывайте фрагментацию и существование целого зоопарка устройств. Это сказывается на времени разработки на всех этапах: проектирование, дизайн, разработка и особенно тестирование.
* В store невозможно общаться с пользователями. Например, пользователь пишет, что у него в ICQ не ходят сообщения. Он недоволен и выражается очень красочно, но не очень информативно. Нет возможности с ним связаться и узнать о проблеме подробнее. Все, что можно сделать – залезть в мониторинг.
* Сейчас некоторые store обзаводятся админкой, в которой можно задать вопросы пользователю, выяснить его контактные данные, посмотреть, в каких странах скачивают ваше приложение, но пока ситуация не слишком улучшилась. Стоит отметить, что Google Play продолжает добавлять эту возможность некоторым разработчикам.

**Платформы разработки**

Каждая из платформ для мобильных приложений имеет интегрированную среду разработки, предоставляющую инструменты, позволяющие разработчику программировать, тестировать и внедрять приложения на целевую платформу. Их достаточно много, языки программирования используются тоже разные. То есть если вы очень хотите написать свое приложение под IOS, необязательно в срочном порядке учить Objective C. Пример - webMethods Mobile Designer подходит для разработки под Java ME, Android, BREW, BlackBerry, Nintendo DS, iOS (iPhone/iPad), Palm/webOS, Sony PSP, Samsung bada, Symbian, Windows Mobile, Windows Phone 7, Windows Desktop, OSX. Язык программирования — Java.

Основные — Android ([Java](https://ru.wikipedia.org/wiki/Java), частично [C](https://ru.wikipedia.org/wiki/Си_(язык_программирования)), [C++](https://ru.wikipedia.org/wiki/C%2B%2B), [Delphi](https://ru.wikipedia.org/wiki/Delphi_(язык_программирования))), IOS SDK(Objective-C, Object Pascal, C++), Windows Phone (C#).

Так же существуют платформы для самостоятельного конструирования мобильных приложений, не обращаясь к услугам IT-специалистов.

Например, BuildAnApp - конструктор для самостоятельной генерации приложений BlackBerry, Windows, iOS и Android. Сборка осуществляется в шесть шагов, на выходе получаем нативное приложение или веб-приложение в зависимости от пожеланий клиента.