

4 лекция.

Тема: Принципы разработки мобильного приложения, изучение требований и формирование технического задания, выбор платформы

План:

- 1. Основные стадии разработки мобильного приложения**
- 2. Техническое задание в процессе создания мобильного приложения**
- 3. Разработка**

Каждый год популярность мобильных приложений только растет — с помощью нескольких нажатий пальцем мы заказываем продукты и одежду, покупаем авиабилеты и записываемся на прием к врачу. Если вы уже хотите [разработать приложение](#), сейчас самое время для его создания — удобный, быстрый и бесконтактный сервис в следующие несколько лет будет как никогда кстати.

В статье рассказываем про основные этапы и принципы разработки приложений — от аналитики и тестирования до выхода на рынок.



1. Основные стадии разработки мобильного приложения

Любой проект важно начинать с детального планирования, изучения собственного бизнеса, аудитории и конкурентов. Чем качественнее будут исследования, тем меньше проблем и доработок будет в дальнейшем.

Аналитика

Во время исследования определите цели бизнеса, изучите аудиторию и каналы коммуникации, проанализируйте конкурентов — это поможет определиться с правильным позиционированием. Аналитика обычно включает в себя интервью с руководителями и клиентами, фокус-группы и экспертную оценку.

Такая подготовка поможет собрать все требования и упаковать их в понятные визуальные модели: схемы бизнес-процессов, майнд мэп, пути пользователей, чтобы определить основы для разработки и перейти к прототипу.

Варианты монетизации

Приложение — еще один способ увеличить прибыль, так что еще на берегу продумайте схему монетизации, чтобы учесть ее при создании интерфейса.

Определиться с монетизацией помогут наводящие вопросы:

- Какую проблему решает сервис?
- За какие возможности люди готовы будут заплатить?
- Сколько у вас есть времени для монетизации? Можно ли подождать, чтобы набрать базу клиентов?

Какие способы монетизации бывают:

- **Реклама.** Аудитория бесплатно использует продукт, но взамен вы показываете им объявления от рекламодателей.
- **Платные функции.** Помимо основных функций приложения, реализованы дополнительные возможности, которые доступны только после оплаты. Например, доступ к новой локации в игре или скрыть анкету и рекламу в дейтинг-сервисе.
- **Платное скачивание.** В этом случае оплата поступает при первом контакте, но для начала создается высокий спрос: ваше предложение должно быть уникальным и полезным. Такой вариант часто используют

создатели корпоративных платформ и программ для обработки фото и видео.

- **Покупки.** Люди платят за реальные и виртуальные товары — одежду и технику для себя или игровых персонажей, валюту внутри игры или другие возможности.
- **Подписка.** После знакомства с бесплатным контентом возможна подписка на дополнительный, чтобы получать больше информации. Такой подход используют платформы с подкастами, медиа и узкопрофильными материалами.
- **Вознаграждение за действия.** Человек получает бонусы за определенную активность: просмотр видео, следование правилам или прохождение опроса. В итоге выигрывают все стороны: рекламодатель рассказывает о своем бренде, разработчики получают премию, пользователи — вознаграждения.

Продвижение

Без маркетинговой стратегии даже самый перспективный проект канет в пучину небытия. Позаботьтесь о продвижении в начале, чтобы к моменту реализации приложение уже было в виш-листе покупателей.

Рекламироваться можно с помощью таргетированной рекламы, нативных материалов в СМИ и блогах, партнерских программ и виральных техник. Распишите портрет своей аудитории, изучите способы продвижения конкурентов, составьте собственную стратегию и меняйте ее в зависимости от обстоятельств.

2. Техническое задание в процессе создания мобильного приложения

После аналитики и проработки стратегии развития наступает процесс разработки [приложения для Android и iOS](#), на первоначальном этапе которого изучается техническая документация и готовится техническое задание.

Обычно в нем прописываются:

- Цели проекта.
- Пользовательские истории и карта путешествия человека — описывают, какие задачи будут решать люди с помощью сервиса, и как они будут это делать.

- Обязательные функции.
- Технические требования к интерфейсу, производительности, роли пользователей, безопасности.
- Реализация функциональности: UX и UI дизайн.
- Этапы разработки.
- Время, необходимое, для всех работ.
- Бюджет.

Описание требований к интерфейсу помогает дизайнерам и разработчикам понять, что именно хочет клиент и как это можно выполнить. Чем подробнее будет ТЗ, тем выше шанс получить то, что действительно нужно и избежать бесконечных правок.

Чаще всего студии разработки помогают с подготовкой ТЗ. Например, в AppCraft мы всегда проверяем ТЗ на соответствие требованиям платформ и разрабатываем его с нуля, если у вас не хватает на него времени или возникли какие-то сложности.

Организация команды

Обычно в состав выделенной проектной команды входят: тестировщик, UX/UI дизайнер, мобильные разработчики, — количество зависит от масштаба проекта — и проектный менеджер, который организывает работу команды.

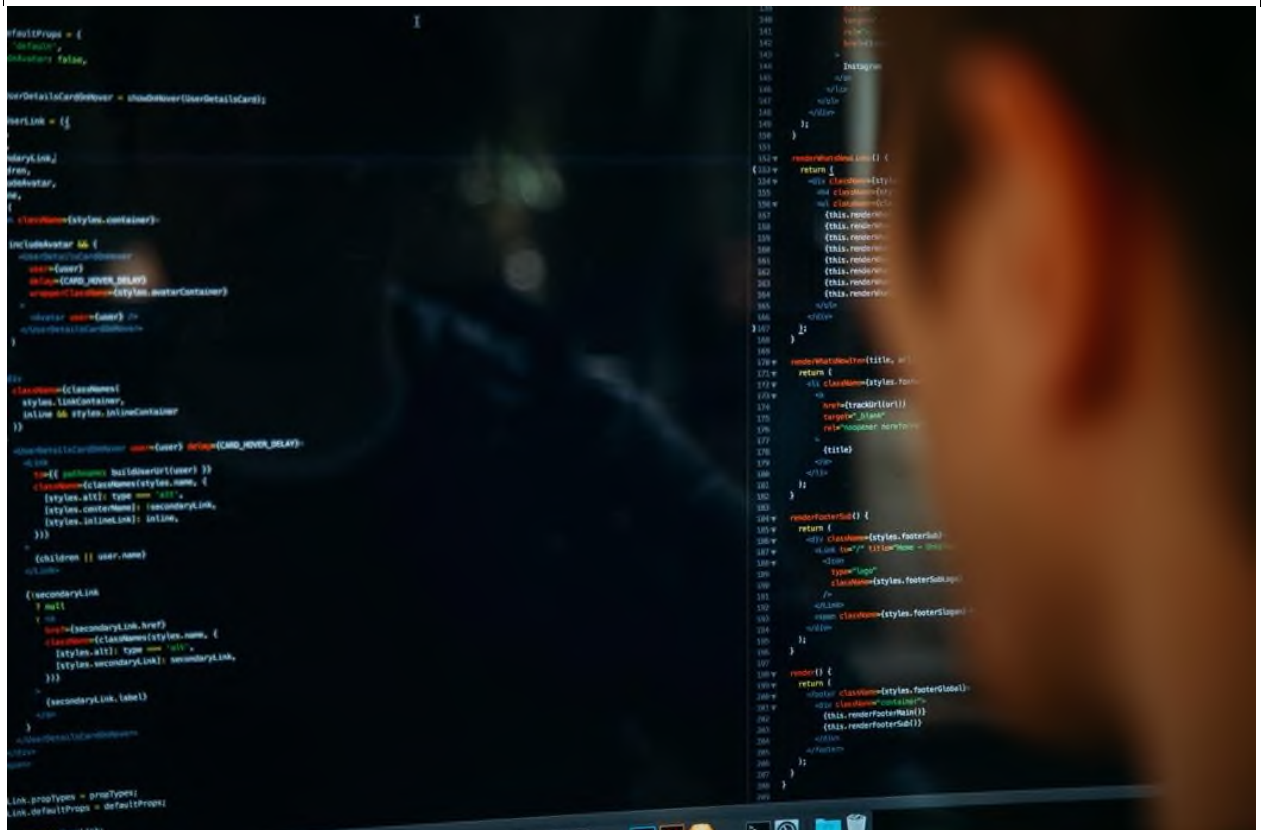
Мы в AppCraft никогда не привлекаем для работы внешних специалистов, потому что выбираем работать с проверенными временем людьми и плотно общаться внутри команды. У такого подхода есть большой плюс — каждый сотрудник сфокусирован на конечном продукте и заинтересован сделать свою работу качественно.

Создание дизайна и прототипа

На этом этапе UX/UI дизайнер выстраивает логику взаимодействия между страницами экранов регистрации и авторизации, заполнения данных, личного кабинета, корзины, оплаты покупки и отслеживания заказа. Разрабатывает внешний вид будущего сервиса в соответствии с техзаданием и фирменным стилем: подбирает цветовое решение, шрифты, отрисовывает иконки, кнопки, пуш-уведомления, слайдеры и т.д.

После согласования дизайна, дизайнер готовит прототип (если это не было сделано на этапе подготовки ТЗ) — в нем воспроизводится базовая логика, структура и функционал.

Обычно прототип создается в виде экранов на каждом этапе пользовательского пути. Это еще не готовый продукт, но он помогает выстроить фундамент и протестировать его функциональность, чтобы уже на начальном этапе исправить ошибки и улучшить пользовательский опыт.



3. Разработка

Одна из трудозатратных стадий включает написание кода, проработку архитектуры и делится на Back-end и Front-end разработку. Мобильные разработчики должны знать концепцию проекта, его уникальность и включаться во все процессы, чтобы оценить жизнеспособность идеи и реализовать желания заказчика.

На этом этапе Front-end программисты разрабатывают продуманный и протестированный клиентский интерфейс и логику платформы.

Back-end разработчики создают сервер для хранения и обмена информации. Специалисты выбирают язык программирования для написания кода и

хостинг для сервера и API, выстраивают систему управления базой данных. Чем лучше выбраны параметры, тем быстрее будет работать приложение.

Разработка может быть реализована несколькими способами:

- **Нативная.** Разрабатывается отдельное приложение для каждой мобильной платформы. Этот способ самый дорогой, но надежный: вы получите полную поддержку от сторов, а интерфейс будет работать быстро и выглядеть максимально органично.
- **Кроссплатформенная.** Разработчики используют универсальный код под все платформы, но операционная система все равно запускает его как нативное. Самый оптимальный вариант в плане «цена-качество».

Подробнее о плюсах и минусах нативной и кроссплатформенной разработки писали в [этой статье](#).

Тестирование

Некоторые компании выделяют тестирование в отдельный этап и досконально проверяют приложение только перед релизом.

Мы думаем, что тестирование приложения нужно проводить на каждом этапе разработки — по готовности каждой части функционала. Лучше потратить больше времени на исправление багов до релиза, чем каждый час получать негативные отзывы на странице после публикации в сторе. Поэтому каждую страницу тестируем настолько часто, насколько это возможно.

Публикация

Перед запуском важно внимательно изучить правила Google Play Store и Apple App Store и подготовить скриншоты страниц, маркетинговый план и описание. После загрузки сторы проверяют всю информацию, актуальность проекта и дают заключение: будут они публиковать приложение или нет. Если все прошло удачно, его можно будет скачивать через несколько дней.

С публикацией могут возникать трудности, поэтому действительно важно ознакомиться со всеми правилами магазинов. В AppCraft проектные менеджеры не оставляют клиентов со всем этим наедине: помогают с публикацией приложения и консультируют по всем вопросам, связанным с регистрацией аккаунтов в магазинах, требованиями к материалам и их форматам.

Доработка и техподдержка

После запуска вы сможете анализировать, какие разделы самые популярные, а какие не очень, сколько человек завершили целевые действия, а какие страницы стоит доработать. Внимательно изучайте и обрабатывайте все входящие данные: они помогут дорабатывать приложение и убирать ненужные функции. Процесс аналитики практически бесконечен, так что вам понадобится техническая поддержка, которая будет фиксировать и оперативно решать текущие проблемы, оптимизировать приложение и дорабатывать его.

В Appcraft гарантийная поддержка кода — 12 месяцев. Мы полностью передаем заказчику права на приложение, но продолжаем мониторить системную аналитику и оперативно устраняем неполадки в приложении, если они вдруг возникают.

Описанные этапы — классический вариант процесса разработки, но мы всегда обсуждаем этот процесс отдельно с каждым новым клиентом. Потому что для нас важно синхронизироваться с заказчиком и сделать так, чтобы процесс разработки был удобным и понятным.

В AppCraft мы занимаемся всеми этапами разработки от аналитики (базовой первичной аналитики или глубоких исследований) до релиза и обеспечиваем оперативную техподдержку. За 10 лет мы создали несколько собственных проектов и больше 200 мобильных приложений для клиентов — мессенджеры, корпоративные решения, банковские системы, e-commerce и соцсети.