

Открытые и закрытые тесты в сторсах. Гайды мобильных сторсов

Процесс тестирования мобильных приложений



Оглавление

На этом уроке	4
Термины, используемые в лекции	5
Интеграция SDK	5
Зачем мы интегрируем SDK в наши программы?	6
Из чего состоит SDK?	6
Библиотеки	7
Пользовательский интерфейс	7
Документация	7
Примеры	8
Прочие инструменты	8
Основные платформы для мобильных приложений и их SDK	8
Google	8
iOS SDK	12
Google Play Console	12
Тестирование через Google Play Console	14
Внутреннее тестирование	14
Закрытое тестирование	15
Открытое тестирование	16
Тестирование лицензий	16
Завершение тестирования	17
Копирование выпуска	17
App Store Connect и Testflight	17
Раздел TestFlight	20
Внутреннее и внешнее тестирование	20
Тестирование приложения	22

Получение фидбэка	23
Требования и гайдлайны сторов	23
Google Play	23
App Store	25
Общие	25
Выводы	27
Домашнее задание	27
Используемая литература	27

На этом уроке

На прошлых уроках мы подробно рассмотрели сервисы, которые обычно используются для дистрибуции тестовых сборок приложений и поработали с инструментами тестирования: эмуляторами, симуляторами и мобильными фермами.

Познакомившись с тем, из чего состоит этап подготовки к тестированию, мы переходим на этап проведения тестирования.



Как правило, ближе к финальному этапу тестирования и релизу мы переходим тестировать в мобильные сторы. Это необходимо, чтобы проверить, что наше приложение нормально загружается в сторы и корректно работает, а сервисы, связанные непосредственно с площадками (например, покупки внутри приложения), функционируют.

На этом уроке мы познакомимся с двумя главными мобильными сторями, App Store и Google Play:

- Интеграция SDK. Что такое SDK и почему нам необходимо это знать?
- Работа в Google Play
- Тестирование с помощью Google Play
- Где посмотреть отзывы пользователей на Google Play
- Работа в App Store
- Тестирование с помощью App Store

- Где посмотреть отзывы пользователей на App Store
- Основные требования и гайдлайны площадок
- Что запрещено в приложениях для App Store и Google Play

Термины, используемые в лекции

Software Development Kit (SDK) — набор инструментов и программ, используемых разработчиками для создания приложений для определённых платформ, операционных систем, компьютерных систем или устройств.


Xcode — это IDE (Integrated Development Environment, «интегрированная среда разработки»), созданная для разработки iOS-приложений.

Android Debug Bridge (ADB) — приложение, в котором мы с помощью командной строки взаимодействуем с подключённым устройством.

Google Play Console — это портал, на котором мы можем выпускать и настраивать параметры запуска нашего приложения, видеть основные метрики, управлять внешним видом страницы, просматривать оценки и отзывы и отвечать на них.

Интеграция SDK

В самом начале разработки приложения разработчики интегрируют несколько *Software Development Kit (SDK)* в наши приложения. Несмотря на то, что это их зона ответственности, нам, тестировщикам, нужно понять, что вообще такое SDK и какие они бывают.

 **Software Development Kit (SDK)** — набор инструментов и программ, используемых разработчиками для создания приложений для определённых платформ, операционных систем, компьютерных систем или устройств.

SDK помогает разработчикам программного обеспечения создавать приложения, которые интегрируются с существующими платформами. Этими платформами могут быть:

- Google Play и App Store (для выпуска приложения, настройки покупок внутри приложения и получения статистики)

- Facebook¹ (для использования инструментов рекламы, продвижения и статистики Facebook)
- Crashlytics (для отслеживания статистики и крашей пользователей)
- HelpShift (для интеграции сервиса работы с техподдержкой через приложение)
- Многие другие

Разработчики со стороны этих платформ создают SDK с возможностью расширить функционал наших приложений за счёт интеграции готовых инструментов.

Зачем мы интегрируем SDK в наши программы?

SDK важны из-за их способности делать разработку программного обеспечения проще и эффективнее.

Во-первых, это может быть обязательным требованием для работы с платформой. Например, наша команда решила, что мы будем собирать статистику по нашему проекту, используя Adjust. Для того чтобы подключить этот сервис, нам обязательно нужно интегрировать их SDK. Только после этого платформа может понять, какие данные и по каким приложениям собирать.

Во-вторых, это облегчает и ускоряет процесс разработки и тестирования. Представим, что у нас в приложении есть встроенные покупки. И вместо того, чтобы самостоятельно разрабатывать систему, которая будет обрабатывать и принимать платежи (а это довольно объёмная и сложная задача), мы интегрируем SDK платформы, на которой выпускаем приложение (пусть будет Google Play и App Store). Таким образом, мы будем использовать инструмент, который создавался сторонней компанией, уже протестирован и, что немаловажно, — выпускает обновления.

В-третьих, у крупных платформ есть саппорт. К нему разработчики могут обратиться, когда возникают какие-то сложности с интеграцией, или если необходимые функции не работают как надо.

Из чего состоит SDK?

Этот блок может меняться от платформы к платформе, но всё же мы выделим основные блоки, из которых состоят SDK:

¹ Meta Platforms Inc. признана экстремистской организацией на территории РФ.

Библиотеки

Библиотека (англ. library) в программировании — сборник подпрограмм или объектов, используемых для разработки программного обеспечения.

То есть это уже написанные функции, которые разработчик может использовать при создании собственных приложений. По сути, библиотеки избавляют разработчиков от самостоятельного написания всех этих функций. Комбинируя предварительно написанные функции с собственным кодом, разработчик может сделать своё приложение более эффективным.

Если рассматривать библиотеки, которые входят в SDK, примером может быть библиотека, отвечающая за покупки внутри приложения. Разработчик подключает библиотеку, настраивает корректный вызов готовых функций, и всё.

Если брать не только SDK, то существует огромное количество библиотек, которые выполняют разные функции. Самый простой пример: у нас есть приложение с функцией заказа билета, в котором пользователь должен выбрать дату, используя календарь. Мы можем просто взять стороннюю библиотеку, в которой уже реализован календарь, и подключить к нашему приложению.

Пользовательский интерфейс

Графический пользовательский интерфейс — это компонент программного обеспечения, предоставляющий визуальные элементы для взаимодействия пользователя с приложением.

Графический пользовательский интерфейс может упростить разработчику работу по интеграции SDK, а также визуальную разработку приложений. Например, SDK может содержать функцию, в которой пользователь создаёт дизайн для веб-приложения, а SDK генерирует код для автоматического создания необходимого форматирования.

Документация

Большинство инструментов, которые используются при разработке, имеют документацию. SDK — не исключение. Документация по SDK содержит основные инструкции по интеграции, описание всех функций, которые может выполнять SDK, разные правила и ограничения при работе с SDK. Эти инструкции и описания помогут эффективно использовать все возможности инструментов.

Примеры

SDK может включать в себя конкретные образцы приложений или веб-страниц. Для разработчиков это может быть отличным примером того, как эффективно интегрировать и использовать SDK. Изучение примера полезно для понимания принципа работы определённых функций, и разработчик получает лучшие практики для реализации.

Прочие инструменты

Сюда входят инструменты для отладки, тестирования и анализа приложения. Это необходимо для обеспечения бесперебойной работы наших приложений.

Платформы, которые предоставляют нам SDK, как и любые другие разработчики, обновляют свои инструменты: расширяют функционал, чинят баги и прочее. После выпуска обновления SDK нашим разработчикам необходимо обновить версию SDK у себя в приложении. Обычно для этого есть инструкции, но, так как инструменты всё-таки не наши, при возникновении проблем может потребоваться намного больше времени, чем планировалось (здесь часто возникает ситуация, когда задача оценивается на два часа, а по факту может занять целый день). После обновления SDK нам, как тестировщикам, необходимо проверять, что функционал, который выполняет SDK, не сломан. Например, после обновления SDK HelpShift нам нужно проверить, что пользователь может отправлять сообщения в саппорт, а наша команда видит эти сообщения и может на них отвечать.

Основные платформы для мобильных приложений и их SDK

Поговорим подробнее про две основные платформы для мобильных приложений и их SDK.

Google

Google предоставляет большое количество SDK, которые помогают нам закрыть разные потребности. Самый первый SDK, с которым мы начинаем работу, — это Android SDK или Android Software Development Kit, разработанный Google для платформы Android. С помощью Android SDK мы можем быстрее и проще создавать приложения для Android. Пройдёмся по основным его компонентам:

- **Android SDK Build-Tools** (инструменты сборки)

Используется непосредственно для создания билдов приложений. Основные функции: сборка, отладка, запуск и тестирование приложений Android.

- **Android Emulator** (эмулятор)

С этим инструментом мы уже знакомы. Напомню, что эмулятор позволяет имитировать устройство на Android у нас на компьютере. С его помощью мы можем посмотреть, как наше приложение будет выглядеть и работать на том или ином устройстве.

Процесс простой: в Android Studio создаём виртуальное устройство, запускаем на нём наше приложение. И дальше в процессе тестирования приложения мы можем снимать логи или взаимодействовать с эмулятором (например, тестировать функции звонков, использования карт и так далее).

- **Android SDK Platform-tools** (платформенные инструменты)

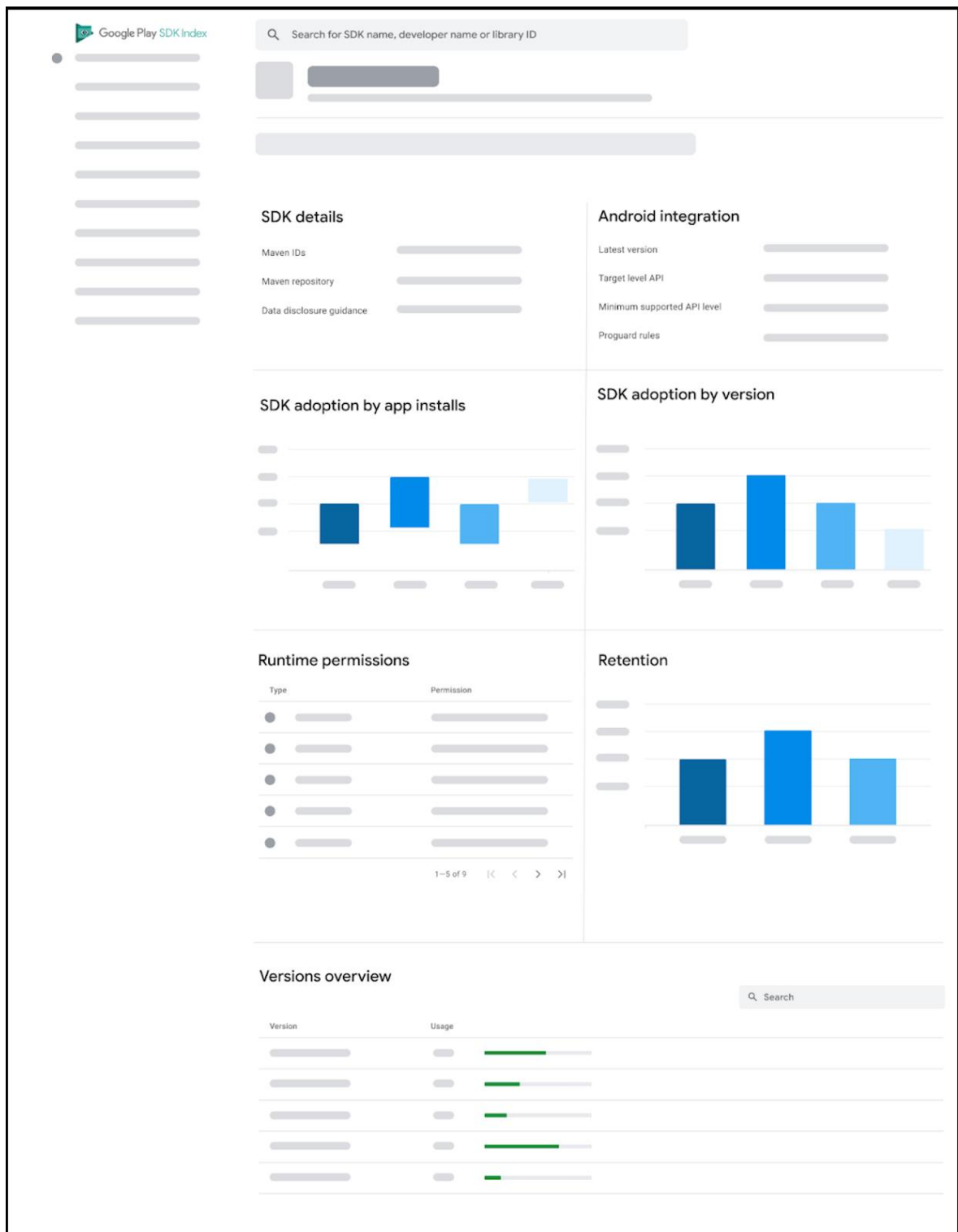
Набор инструментов, полезных для тестирования и отладки приложения. Он включает в себя:

1. Android Debug Bridge (ADB) — приложение, в котором мы с помощью командной строки взаимодействуем с подключённым устройством. Можем снимать логи, делать скриншоты, устанавливать и удалять приложения.
2. Fastboot помогает настроить мобильное устройство (перепрошить или поменять настройки).
3. Systrace предоставляет информацию об устройстве: системная информация, процессы и так далее.

Помимо этого, у Google Play есть отличная площадка под названием [Google Play SDK Index](#). Это общедоступный портал, на котором перечислены более 100 наиболее широко используемых коммерческих SDK и предоставлена информация о каждом из них.

Различные SDK представлены по категориям:

- | | |
|--------------------------|--------------------------------|
| ● Реклама и монетизация | ● Платежи |
| ● Аналитика | ● Социальные |
| ● Управление данными | ● Аутентификация пользователей |
| ● Местоположение | ● Поддержка пользователей |
| ● Маркетинг и вовлечение | |



Для каждого из SDK портал предоставляет данные об использовании из приложений Google Play и анализ кода, что позволяет легко понять, подходит ли нам тот или иной SDK. Мы сразу видим следующую информацию:

- Какие разрешения для приложений Android может запрашивать SDK
- Обеспечивает ли поставщик SDK соответствие кода политикам Google Play
- Различные метрики

На скрине ниже основная информация по SDK, который предоставляет **Adjust** (это сервис, который собирает аналитику и помогает нам анализировать поведение пользователя в приложении).



Adjust SDK

Provided by Adjust ▸ Analytics



This SDK is registered in Google Play SDK Console



By registering, the provider of this SDK has accepted the Google Play SDK Console Terms of Service, which includes a commitment that their SDKs will not cause apps to violate Google Play policies. The SDK provider can also use SDK Console to monitor SDK usage and report issues to Google.

[Learn more](#)

SDK details

Maven ID com.adjust.sdk:adjust-android

Maven repository Maven Central

Data safety section guidance [View guidance](#)

Android integration

Latest version 4.31.0

Target API level [?](#) 31

Minimum API level [?](#) 9

ProGuard rules [?](#) Provided

Мы можем понять, что SDK зарегистрирован в Google Play Console, увидеть основную информацию о SDK и требования для интеграции SDK в приложение.

Android permissions [?](#)

Type	Permission
Phone	READ_PHONE_STATE

1 - 1 of 1 |< < > >|

Так выглядит список разрешений, которые будет требовать наше приложение после интеграции SDK.

Нам, как тестировщикам, нужно будет проверить, что все разрешения запрашиваются корректно и никаких больше не запрашивается.

iOS SDK

iOS SDK — это набор инструментов, созданный компанией Apple для упрощения разработки приложений. iOS SDK поставляется вместе с Xcode.

Xcode — это IDE (Integrated Development Environment, «интегрированная среда разработки»), созданная для разработки iOS-приложений. Это один из самых популярных инструментов для создания приложений. Это полный набор программного обеспечения, и с его помощью разработчики могут выполнять множество задач, таких как проектирование пользовательского интерфейса, разработка кода для приложений, компиляция и тестирование кода, а также проверка наличия ошибок в коде. Xcode также позволяет отправлять приложение в магазины приложений. SDK содержит инструменты, дающие разработчикам доступ к различным функциям и службам iOS-устройств. Также есть возможность запускать симулятор и смотреть, как будет выглядеть наше приложение на том или ином виртуальном устройстве.

Итак, в наши приложения интегрированы все необходимые SDK. Как мы уже проговорили, интеграцией SDK, как правило, занимается отдел разработки. Нам, как тестировщикам, необходимо проверить, что после интеграции в приложении ничего не сломалось (регрессионное тестирование).

А самое главное, что после интеграции наше приложение корректно работает с тем сервисом, для которого делалась интеграция (то есть, например, после интеграции Adjust SDK в сервис Adjust должна уходить корректная статистика по приложению). Обычно информацию о том, какой функционал добавляется при интеграции SDK, можно получить от отдела разработки.

Мобильные сторы

Существует несколько сторов для мобильных приложений в зависимости от вашей операционной системы. Мы все с вами, пользуясь мобильными устройствами, пользуемся и сторями: скачиваем различные приложения, устанавливаем на свои устройства и обновляем их.

То есть это платформы, на которых разработчики могут публиковать свои приложения, а пользователи — находить, загружать и обновлять их.

Два самых крупных мобильных стора это App Store и Google Play Store.

Google Play Store — официальный магазин приложений для устройств Android. Он доступен на большом количестве смартфонов, планшетов и других устройств под управлением Android. Google Play Store принадлежит компании Google.

App Store — это официальный магазин приложений для устройств Apple, таких как iPhone, iPad и Apple Watch. App Store принадлежит компании Apple.

В этих сторсах очень большое количество разнообразных приложений и игр. В обоих сторсах есть функция автоматического обновления приложений для пользователей. То есть при выходе новой версии приложения пользователям не надо заходить самостоятельно в стор на страницу приложения и обновлять его. Так же при загрузке приложений из этих сторсов пользователи могут совершать покупки внутри приложения.

Помимо этих двух сторсов для устройств на Android существует еще несколько различных сторсов. Это, например, Samsung Galaxy Store (стор для устройств компании Samsung), Amazon Appstore (стор для Kindle устройств). Эти площадки не такие большие, но в них все равно есть большое количество приложений и различных функций. Apple более закрытая система, поэтому альтернатив App Store на iOS нет.

Разработчикам мобильных приложений для того, чтобы разместить свое приложение в сторсе, необходимо создать аккаунт разработчика, заполнить всю необходимую информацию о компании. После этого можно работать с приложением: загрузить версию приложения, заполнять информацию по приложению и пройти проверку площадки.

Google Play Console

Рассмотрим пошагово, как мы будем взаимодействовать с консолями разработчиков в Google Play и App Store.

Google Play Console — это портал, на котором мы можем выпускать наше приложение и настраивать параметры его запуска, видеть основные метрики (например, количество активных пользователей), управлять внешним видом страницы, просматривать оценки и отзывы и отвечать на них. И, что важно конкретно для нас, — настраивать внутреннее и внешнее тестирование.

Рассмотрим основные разделы консоли.

Дашборд. Здесь мы можем видеть основную информацию по нашему приложению. Можно найти все входящие сообщения по приложению (в основном это будут технические сообщения об изменениях в правилах пользования или

предупреждения об изменениях, которые необходимо реализовать в нашем приложении, чтобы соответствовать политикам платформы).

Дальше будет информация по основным KPI, которые мы настроим (например, количество ежедневных пользователей, крэши, рейтинг). Анализ логов, отчётов о сбоях и отзывов поможет нам выявлять проблемные места в приложении.

В разделе **Статистика** мы можем увидеть более подробные данные по различным показателям и фильтровать их. Это может быть информация по нашим пользователям, мобильным устройствам, крэшам и ANR (от англ. Application Not Responding) — ошибкам, которые говорят нам о том, что приложение не отвечает. «Зависло», простыми словами. Это довольно важный показатель, который отслеживает сама платформа и сигнализирует, если ошибки достигают высокой частоты появления).

В разделе **Обзор публикаций** можно найти изменения, которые ждут проверки и готовы к выпуску на всех пользователей. Как правило, в последнее время проверка приложения проходит в течение одного-двух дней. Мы можем настроить, выпускать ли приложение автоматически после проверки или самостоятельно, в нужное нам время.

Далее идут разделы, связанные с разными **типами выпусков**, которые могут быть настроены в приложении. Это основная информация по рабочей версии: дата выпуска, номер билда, страны, на которые открыто приложение. Разделы тестирования мы рассмотрим позже.

Раздел **Охват и устройства** показывает нам основную информацию о тех устройствах, на которых устанавливается и используется наше приложение.

Раздел **App Bundle Explorer** хранит данные о версиях билдов, которые мы загрузили в консоль ранее. Здесь просто отображается основная информация по каждому из билдов.

Раздел **Настройки**. Здесь находятся расширенные настройки. Мы можем настраивать списки тестировщиков, доступность приложения и прочее.

Далее идут разделы, которые отвечают за то, как будет выглядеть **страница приложения** в сторе. Загружаем информацию о приложении, описание, скриншоты и видео.

Раздел **Эффективность страницы приложения**. Тут мы смотрим, откуда пришли наши пользователи. Это может быть раздел «Интересное» в Google Play (подборки, которые делает сам стор), поиск в Google Play (когда пользователь вводит в поиске название нашего приложения или похожее и переходит на страницу приложения)

или сторонние рефералы (когда пользователь смотрит рекламу и переходит в стор, кликая на рекламу).

В разделе **Игровые сервисы** Google Play настраиваются игровые параметры. Например, мы можем ввести достижения, которые есть в игре, и они будут интегрированы с сервисами Google Play.

В разделе **Оценки и отзывы** мы видим все оценки и отзывы, а также информацию о том, с каких устройств и из каких стран мы получаем те или иные оценки. Также есть возможность отвечать на отзывы.

В разделах **Android Vitals** и **Эффективность** мы видим важную техническую статистику. Есть данные по сбоям и ANRam. Мы можем нажать на конкретный сбой и посмотреть даже раздел приложения, в котором он происходит.

В разделе **Монетизация** мы можем настроить цену приложения, покупки внутри приложения, подписки.

В разделе **Правила и программы** мы заполняем разные анкеты касательно нашего приложения. Например, целевая аудитория, возрастные ограничения и прочее.

Тестирование через Google Play Console

Проведение тестирования через консоль Google Play дает определенные преимущества, включая доступ к инструментам бета-тестирования, проверки совместимости устройств и соблюдение политик Google Play. Это позволяет разработчикам выявлять и устранять проблемы до того, как приложение достигнет более широкой аудитории, обеспечивая более плавный выпуск и более высокую вероятность соответствия стандартам качества.

Как правило пререлизные версии тестируются через Google Play Console.

А теперь вернёмся к разделам тестирования и пройдём все этапы, которые доступны в консоли. Это не означает, что все из этих этапов обязательно должны быть включены в ваш потенциальный процесс тестирования. Это будет зависеть от конкретного приложения и команды, в которой вы окажетесь.

Внутреннее тестирование

Основные особенности внутреннего тестирования: оно ограничено 100 пользователями, мы можем добавлять тестировщиков по их гугл-аккаунтам. Если приложение платное, пользователям не нужно платить, чтобы скачать его.

Учитывая количество доступных тестировщиков, этот вид тестирования удобно проводить среди тестировщиков внутри команды. Приложению не нужно проходить ревью, чтобы тестировщики получили доступ к нему. При этом нам не обязательно полностью заполнять все настройки и описания, то есть тестирование можно проводить на довольно раннем этапе.

Чтобы загрузить приложение в стор для внутреннего тестирования, мы заводим версию, вносим список тестировщиков, добавляем при необходимости заметки, сохраняем выпуск. На этом этапе происходит автоматическая проверка. Из возможных ошибок тут — неправильная версия приложения, неподходящая подпись или ненастроенный список тестировщиков.

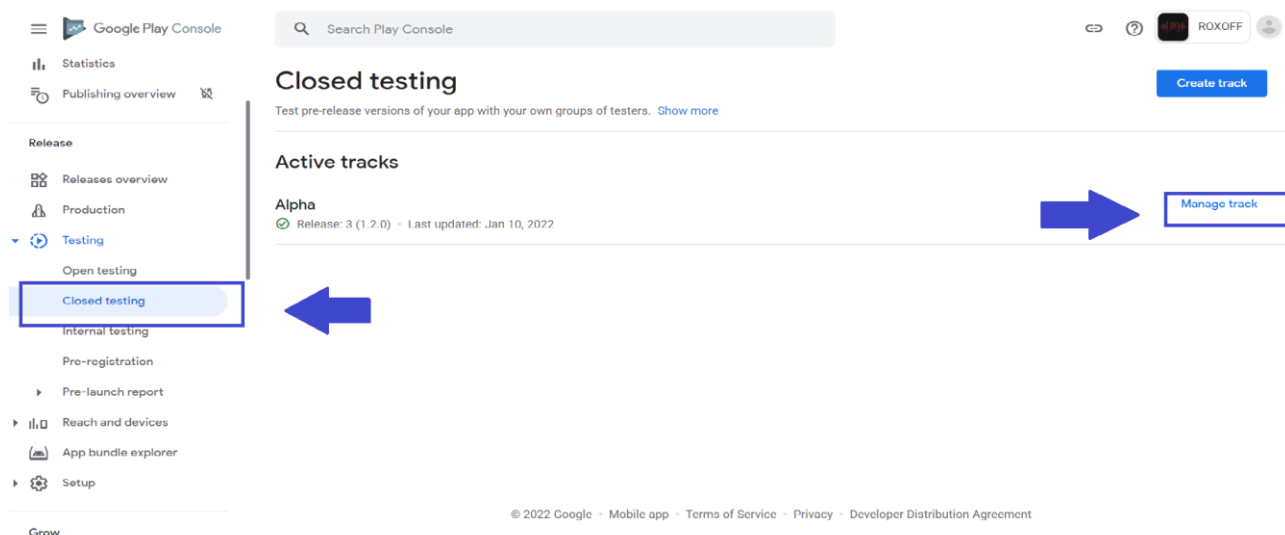
Если у нас ошибка со списком тестировщиков, можно просто поправить список активных тестировщиков и пересохранить выпуск. Если ошибка типа неправильного кода версии (такое бывает, если, например, у нас была ранее загружена версия с кодом «3», и разработчик, когда собирал новую версию, не поднял код версии до «4»), то придётся пересобрать приложение и загружать его в консоль заново.

Итак, если ошибок нет, мы можем открыть выпуск для внутренних тестировщиков.

Далее идём в информацию о выпуске, копируем ссылку на тестовую версию приложения и отдаём его всем тестировщикам, которых мы внесли в список.

Тестировщик переходит по этой ссылке (сейчас из России нужен будет VPN), подтверждает участие в программе внутреннего тестирования и уже из стора качает приложение. При этом обратите внимание: у приложения в этот момент может не быть иконки, описания или основной информации.

Чтобы обновить версию для тестирования, в консоли можно снова создать новый выпуск, повторить действия с загрузкой приложения и открыть версию.



Create production release

Drop Android App Bundles (.aab) or APKs here to upload

[Upload](#) [Add from library](#)

app-release.apk

File type	Version	API levels	Target SDK	Screen layouts	ABIs	Required features
APK	2 (2.0)	21+	29	4	All	4 →

Release details

Release name *

7 / 50

This is so that you can identify this release, and isn't shown to users on Google Play. We've suggested a name based on the first app bundle or APK in this release, but you can edit it.

Release notes [Copy from a previous release](#)

<en-GB>
Enter or paste your release notes for en-GB here
</en-GB>

Release notes provided for 0 language

Let users know what's in your release. Enter release notes for each language within the language tags.

Google Play Console

Search Play Console

[Link](#) [Help](#) [Profile](#)

← All apps

- Dashboard
- Inbox 3
- Statistics
- Publishing overview

Release

- Releases overview
- Production
- Testing
 - Open testing
 - Closed testing
 - Internal testing
 - Pre-registration

Alpha

[Edit release](#)

Track summary

Active - Draft release - 177 countries / regions

Releases Countries / regions

Testers

Testers

Testers

Choose who can access your test. You can select testers using email lists, or Google Groups.

☒ Email lists

<input checked="" type="checkbox"/> List name	Users	
<input checked="" type="checkbox"/> TEST	3	→
<input checked="" type="checkbox"/> Testers	2	→

[Create email list](#)

Edit email list

×

* - Required fields

List name *

11 / 200

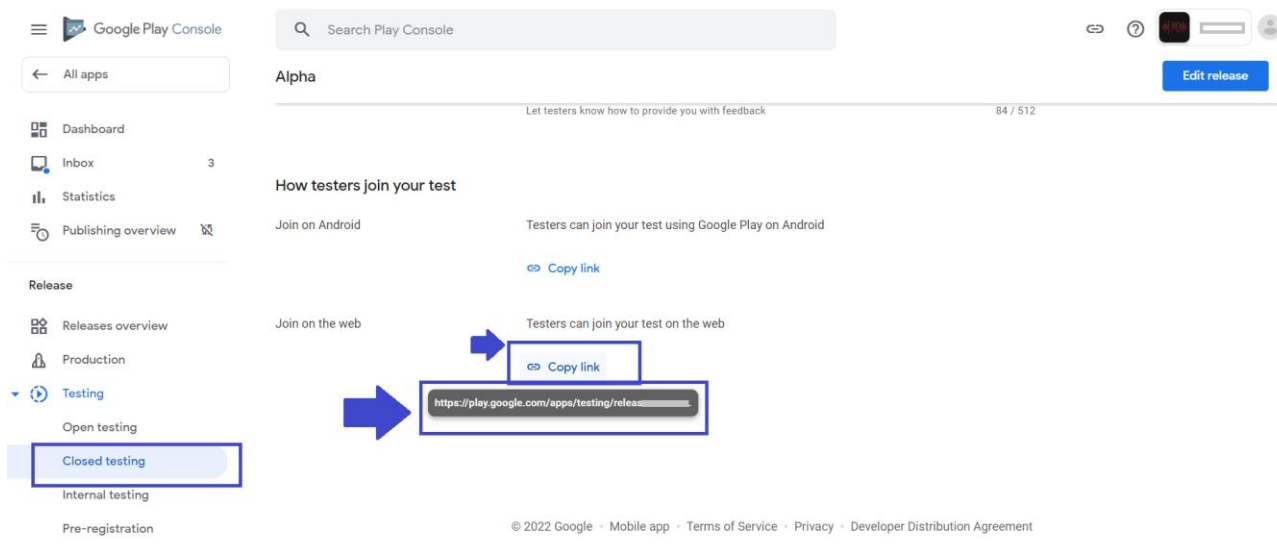
Add email addresses

Add 1 or more email addresses, separated by a comma. Press enter to add.

[Upload CSV file](#)

Email addresses added *

user@example.com	✕
user@example.com	✕
user@example.com	✕



Закрытое тестирование

В закрытом тестировании мы не ограничены 100 тестировщиками, добавляем тестировщиков либо по адресам электронной почты, либо по группам. То есть его можно проводить после того, как мы провели внутреннее тестирование на ограниченном количестве игроков.

Для выпуска этой версии технически всё выглядит так же:

- Мы создаём выпуск, добавляем тестировщиков и открываем версию. Важным отличием, помимо количества тестировщиков, здесь является то, что мы можем настраивать разные дополнительные закрытые версии для тестирования.
- Для этого опять переходим в раздел **Закрытое тестирование**.
- Нажимаем «Создать версию», добавляем тестировщиков.

Конечно, при создании версии важно правильно её называть, чтобы не было путаницы, когда у нас открыто несколько разных версий одновременно. На этапе создания закрытого тестирования мы можем ввести адрес электронной почты или URL для отзывов и предложений тестировщиков. Эта информация будет видна на странице подтверждения участия в тестировании.

Открытое тестирование

У открытого тестирования есть несколько отличительных характеристик. Наше приложение можно будет найти в Google Play просто через поиск (но можно также дать ссылку на странице приложения в сторе). То есть нам не нужно вводить пользователей. Но это также значит, что приложение должно пройти полноценную

проверку в Google Play. Поэтому при планировании этого вида тестирования необходимо заложить 1-2 дня на прохождение проверки. Пользователи через Google Play могут присоединяться к программе открытого тестирования и отправлять нам отзывы. Количество тестировщиков можно настраивать. Есть два варианта:

- Без ограничений: этот вариант выбран по умолчанию
- Ограниченное число: нужно указать максимальное число участников (не менее 1000)

Тестирование лицензий

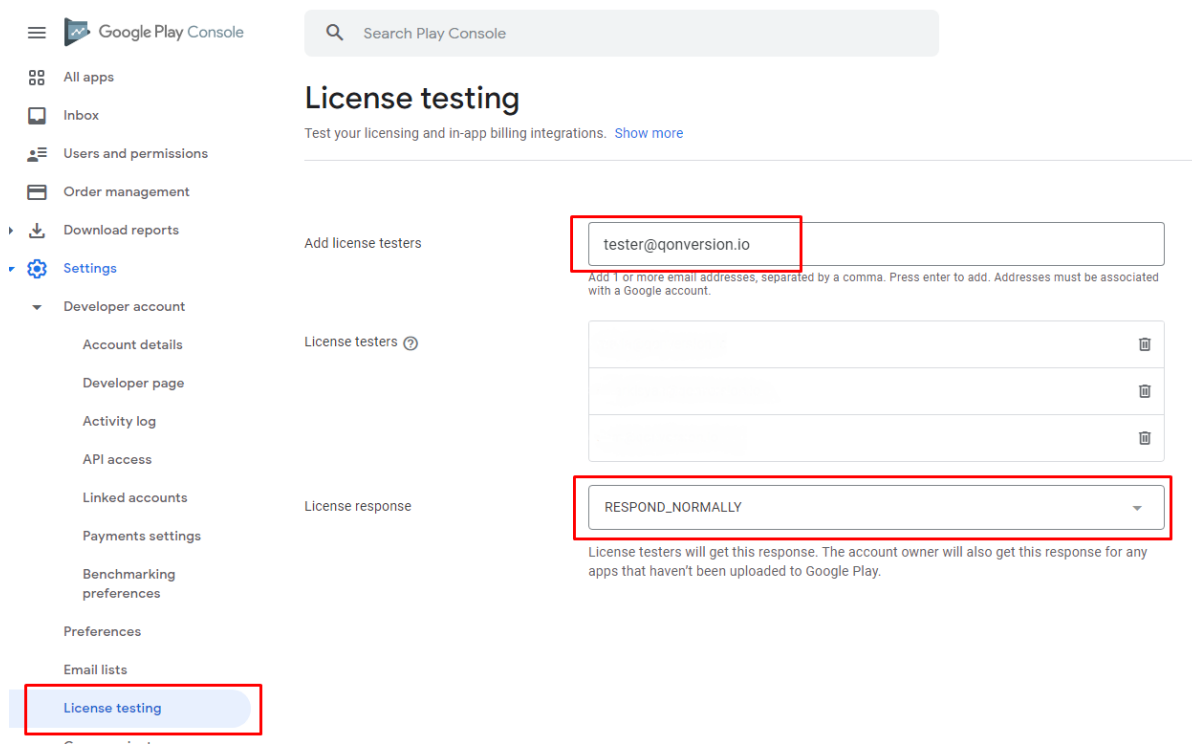
Если у нашего приложения есть покупки внутри приложения (IAP — in-app purchase), мы можем настроить тестирование лицензий.

Для этого в разделе **Настройка > Тестирование лицензий** нужно добавить тестировщиков. После того как тестировщики добавлены, они смогут совершать покупки внутри приложения без использования данных реальных карт. Эти аккаунты смогут совершать тестовые покупки приложения и контента или оформлять тестовые подписки.

При совершении покупки лицензированному тестировщику будет доступно два способа оплаты:

- Тестовая карта, всегда подтверждать
- Тестовая карта, всегда отклонять

Опция «всегда отклонять» полезна, когда нам нужно проверить, как поведёт себя приложение при вводе данных невалидной карты.



Завершение тестирования

Для завершения тестирования нам нужно перейти в версию, тестирование которой мы хотим остановить. Нажимаем **Управлять типом версии** → **Приостановить версию**. После окончания тестирования приложение останется на устройствах тестировщиков, но обновляться не будет.

После проведения всех этапов тестирования мы можем открывать версию для всех пользователей. Процесс создания версии такой же, только необходимо указать страны, на которые будет открыто наше приложение.

Копирование выпуска

На каждом из этапов мы можем скопировать наш выпуск в другой тип (то есть перевести выпуск из внутреннего тестирования в открытое и так далее).

Для этого переходим к любому типу тестирования. Нажимаем «Скопировать выпуск» и выбираем, куда копировать. Например, из внутреннего тестирования мы можем скопировать выпуск в закрытое или открытое тестирование. А можем сразу скопировать в рабочую версию.

Размещение и тестирование приложения в Google Play

На разных проектах и в разных компаниях в зависимости от процессов и размеров команд разные люди отвечают за релиз приложений в сторх. Но нам как

тестировщикам будет полезно познакомиться с тем, как происходит процесс релиза приложений на площадках. Это важно для понимания процесса в целом, а так же необходимо знать какие параметры требуют указывать сторы.

Инструкция: [Пошаговый процесс загрузки приложения в Google Play Console](#)

App Store Connect и Testflight

С помощью App Store Connect мы настраиваем наше приложение для App Store, который распространяет приложения для iOS и MacOS.

В App Store Connect основные разделы — это:

- Мои приложения
- Аналитика приложений
- Продажи и тренды
- Платежи и финансовые отчёты
- Пользователи и доступ
- Соглашения, налоги, банковские операции

Как мы видим, тут, в отличие от Google Play Console, аналитика вынесена в отдельный раздел. Нас с вами сейчас интересует раздел **Мои приложения**, где мы можем создавать новые приложения или смотреть и изменять данные по уже существующим.

Откроем какое-либо приложение. Слева будет меню, где мы видим всю основную информацию и разделы по приложению.

Первым у нас открывается раздел с **данными о версии приложения** и информацией о ней. Здесь мы загружаем скриншоты, рекламный текст, описание, ключевые слова, заметки к версии, данные о нашем сайте и дополнительную информацию. В этом же разделе мы можем выбрать билд, который мы будем отправлять на проверку в App Store.

Далее идёт раздел **Общая информация о приложении**.

В **Информации о приложении** находится техническая информация (ID, категории, возрастные ограничения).

В **Ценообразовании и доступности** мы указываем цену приложения и страны, в которых оно будет доступно.

В **Конфиденциальности приложения** указывается адрес политики конфиденциальности и анкета по типам данных, которые приложение собирает, а также информация о том, отслеживает ли приложение пользователей.

В **Оценках и отзывах** мы видим сводку по оценкам и отзывам. Данные фильтруются по странам. В зависимости от роли нашей учётной записи мы можем отвечать на отзывы пользователей. Обратите внимание: в отличие от Google Play Console мы можем сбрасывать рейтинги приложения в App Store при каждом выпуске новой версии.

В **Истории обновлений** хранятся все наши действия в консоли по всем ранее выпущенным версиям. Причём здесь всё видно по датам, и мы можем посмотреть, например, когда мы отправили на проверку приложение и через сколько эта версия прошла проверку.

В **Проверке приложения** содержится история завершённых проверок и те проверки, которые сейчас в процессе.

Последний раздел здесь — **Функции**.

Встроенные покупки: мы можем создавать новые встроенные покупки (IAP) и управлять ими. Покупки бывают двух типов: нерасходуемые и расходуемые.

В **Подписках** мы можем добавлять и редактировать подписки. Подписки могут быть с автоматическим разделением (когда у пользователя раз в определённый промежуток времени списывается конкретная сумма, после чего открывается доступ к какому-либо функционалу приложения). Подписка такого типа возобновляется автоматически, пока не будет отменена пользователем. Также подписки могут быть без автоматического продления. Пользователи могут приобрести доступ к сервисам или контенту на ограниченный период, например, в виде сезонного абонемента на внутриигровой контент.

Продвижение в App Store. Акции App Store используются для продвижения существующих покупок в приложении. Как только мы добавим рекламное изображение к своим существующим покупкам в приложении, они станут доступны для продвижения.

Специальные страницы продукта. Специальные страницы продукта позволяют разработчикам приложений создавать варианты основной страницы продукта, изменяя скриншоты приложения, видео или рекламный текст. Можно запустить до 35 специальных страниц продукта. У этих страниц будет свой URL-адрес, который можно использовать для продвижения через различные каналы.

Далее у нас идёт вкладка **Услуги**, где мы можем управлять **Game Center**, **Шифрованием** и **Промокодами**.

Game Center — если у нас игровой продукт, мы можем использовать Game Center для вызовов, лидербордов и достижений.

Шифрование. Этот раздел мы заполняем в том случае, если применяем стандартные или частные алгоритмы шифрования.

Раздел **Промокоды** позволяет генерировать промокоды для приложения, по 100 промокодов на каждую покупку. Пользователю необходимо будет просто ввести промокод в App Store, и он получит доступ к платному приложению или функционалу.

Раздел **Аналитика** содержит следующие метрики:

- **Показы** — сколько раз пользователи видели иконку приложения в App Store (то есть в списках в результатах поиска или подборках).
- **Просмотры страницы продукта** — сколько раз пользователи магазина просматривали страницу приложения в App Store.
- **Коэффициент конверсии** — общее количество загрузок, разделённое на количество уникальных показов на устройствах.
- **Всего загрузок** — количество первых и повторных загрузок на устройствах.
- **Выручка** — предполагаемая сумма выручки, которую разработчик получит от своих продаж, за вычетом комиссии Apple.
- **Доходы / платящие пользователи** — соотношение доходов и количества уникальных пользователей, оплативших приложение или покупку в приложении.
- **Сеансы / активные устройства** — количество раз, когда люди использовали приложение в течение как минимум двух секунд, разделённое на общее количество устройств с хотя бы одним сеансом за выбранный период.
- **Сбои** — общее количество сбоев.

Раздел TestFlight

TestFlight — это место, где мы можем управлять всем, что связано с процессом тестирования приложения, а также протестировать своё приложение перед выходом на рынок. Основные разделы:

- Сборки для каждой версии приложения с общим обзором

- Обратная связь со сбоями и скриншотами
- Внутреннее тестирование с пользователями App Store Connect
- Внешнее тестирование и тестировщики, которые имеют доступ к тестируемому приложению
- Общая информация — список всех тестировщиков и информация о тестировании

Что касается непосредственно настройки тестирования, сборкой и загрузкой билда с TestFlight обычно занимается разработка. Мы с вами сейчас настроим тестирование. Тестирование бывает внутреннее и внешнее.

Внутреннее и внешнее тестирование

- **Внутренний тестировщик** — это пользователь App Store Connect, у которого есть роль администратора, менеджера приложений, юриста, разработчика или маркетолога с доступом к нашему приложению. Обычно это член команды или клиент, для которого мы разрабатываем приложение. Мы можем добавить до 100 внутренних тестировщиков, и каждый участник может тестировать до 30 устройств.
- **Внешний тестировщик** — это любой пользователь за пределами нашей команды, который хочет протестировать приложение. Внешние тестировщики не имеют доступа к нашей учётной записи App Store Connect. Они могут только загрузить и установить приложение. Мы можем добавить до 10 000 внешних тестировщиков.

Для того чтобы настроить внутреннее тестирование, нам не надо проходить проверку приложения. Просто смотрим, что билд есть в разделе TestFlight и у него корректный статус: «Готов к отправке».

BUILDS

Builds

iOS

macOS

tvOS

Feedback

Crashes

Screenshots

Internal Testing

AS App Store Connect Users

ET Executive Team

External Testing

TF Tropical Forest Explorers

+8 See More

General Information

All Testers (9,931 of 10,100)





Test Information

About TestFlight Data ?

iOS Builds

The following builds are available to test. [Learn more about build status and metrics.](#)

2.3.0

BUILD	STATUS	GROUPS	INVITES	INSTALLS	SESSIONS	CRASHES	FEEDBACK
 4	Testing Expires in 47 days	AS	16	4	11	3	6
 3	Testing Expires in 24 days	ET	32	3	9	2	4
 2	Ready to Submit Expires in 20 days	TF	26	3	–	–	–
 1	Expired	XT	26	3	–	–	–

Version 2.2.0

Version 2.1.0

Когда билд появляется, мы можем настроить тестировщиков.

1. Нажимаем на сборку.
2. В разделе **Группы** создаём новую группу и выбираем необходимых пользователей. Нажимаем «Добавить». Все выбранные тестировщики получат электронное письмо со ссылкой для загрузки и установки этого билда через приложение TestFlight.
3. Для создания внешнего тестирования необходимо будет отправить билд на проверку. Для настройки в разделе TestFlight переходим: **Общая информация > Информация о тестировании**. Необходимо заполнить следующие поля:
 - Описание приложения для бета-тестирования
 - Email для обратной связи
 - Политика конфиденциальности (URL)
 - Контактная информация

1. После заполнения всех полей нажимаем «Сохранить», идём в раздел **Добавить внешних тестировщиков**.

Тут нам нужно создать группу (в этом разделе в дальнейшем можно добавлять новых тестировщиков или новые группы).

2. Вводим название группы и с помощью кнопки (+) добавляем электронные адреса тестировщиков (их можно добавить вручную по одному, а можно списком с помощью CSV-файла).

25

Forest Explorer ▾ App Store Features TestFlight Activity

Builds
iOS

Feedback
Crashes
Screenshots

Testers & Groups +
All Testers (7)
App Store Connect Users
External Testers
Tropical Forest Explorers

App Information
Test Information
About TestFlight Data ?

Tropical Forest Explorers [Edit Name](#)

You can add anyone to this group, and they can test builds using the [TestFlight app](#). Builds may need approval from Beta App Review.

Tester Management

Public Link ?
<https://testflight.apple.com/join/QXTE8A73> [Disable Link](#) | [Copy Link](#)

Tester Count ?
0 [Set limit](#)

Tester Feedback ?
[Disable](#)

Testers (3) +

[Add New Testers](#)
[Add Existing Testers](#)
[Import from CSV](#)

INVITATION	NAME	STATUS ^	SESSIONS	CRASHES	FEEDBACK
mruiz2@	Maria Ruiz	Invited Resend Invite Apr 20, 2020			
annejohnson1@mac.com	Anne Johnson	Invited Resend Invite Apr 20, 2020			
billjames2@mac.com	Bill James	Invited Resend Invite Apr 20, 2020			

- После добавления всех тестировщиков нажимаем на группу и в разделе **Сборки** добавляем нужный билд. Обратите внимание: мы добавляем билды для внутреннего и внешнего тестирования отдельно. Это значит, что для каждой группы пользователей мы можем назначать свой билд для тестирования.
- Добавили билд, появляется окно, которое предлагает нам отправить билд на проверку. Соглашаемся. Далее нужно ждать, когда пройдёт проверка. Как правило, она занимает пару дней.

Когда приложение пройдёт проверку бета-версии, мы получим электронное письмо с подтверждением того, что приложение может начать внешнее тестирование. Если мы установили флажок «Автоматически уведомлять тестировщиков», на этом этапе наши внешние тестировщики получают уведомления по электронной почте. Или же нам придётся вернуться в App Store Connect, чтобы начать тестирование, и после этого внешние тестировщики получают приглашение по электронной почте.

Builds

iOS
macOS
tvOS

Feedback

Crashes
Screenshots

Internal Testing

AS App Store Connect Users
ET Executive Team

External Testing

TF Tropical Forest Explorers
+8 See More

General Information

All Testers (9,931 of 10,100)

Test Information

About TestFlight Data

Test Information

To start testing your builds, provide the following information.

Save

Beta App Information

English (U.S.)

Beta App Description

Go wild and discover trails, parks, and off-the-beaten-path terrain with Forest Explorer. Whether you're bushwhacking in backcountry or looking for a quick local hike, Forest Explorer has thousands of trails and destinations from around the globe to explore.

- Over 90,000 trails with photos to browse
- Explore trails rated by distance, elevation, duration, and difficulty

2,811

Feedback Email

chavez4@gmail.com

Apple TV Privacy Policy

Marketing URL

http://example.com

Privacy Policy URL

http://example.com

6,000

Add New Testers to the Group "Top-Testers"

We'll invite these testers to test the builds you add to this group.

Testers

1 of 10,000 Available

	Email	First Name	Last Name
1	<input type="text" value="a.test@raywenderlich.com"/>	<input type="text" value="External"/>	<input type="text" value="Tester"/>
2	<input type="text"/>	<input type="text"/>	<input type="text"/>
3	<input type="text"/>	<input type="text"/>	<input type="text"/>

Cancel

Add

Тестирование приложения

Рассмотрим процесс уже после того, как мы сделали все настройки в App Store Connect. Тестировщики, независимо от того, внутренние они или внешние, получат письмо с приглашением принять участие в тестировании.

App Store Connect

Dear [REDACTED]

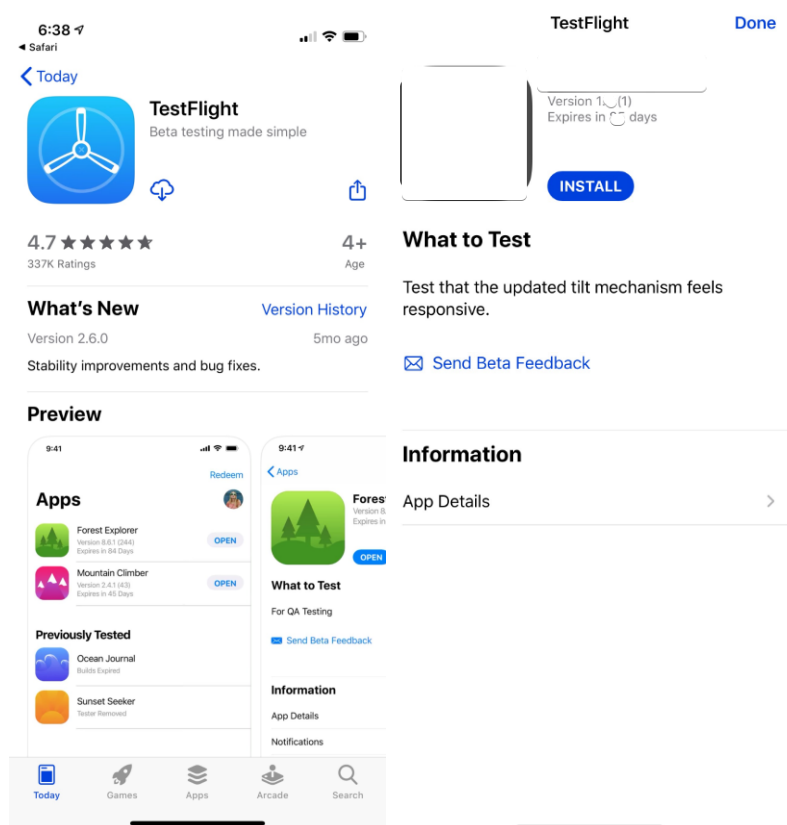
[REDACTED] has invited you to access App Store Connect so you can help distribute and manage their apps on the App Store. This suite of tools is used to submit apps, view sales reports, access app analytics, beta test apps, and more. Please accept this invitation within three days.

[Accept invitation](#)

If you have any questions, [contact us](#).

Best regards,
The App Store Team

1. Скачиваем в App Store приложение TestFlight.
2. Логинимся.
3. Далее нам нужно открыть письмо-приглашение и принять приглашение.
4. После принятия приглашения в TestFlight будет отображаться билд, который нам отправили для тестирования. Тут важно отметить, что сборка активна в течение 90 дней, после чего она деактивируется.



5. Когда мы откроем тестируемое приложение, первым делом можно сделать скриншот. После того как скриншот сделан, выбираем опцию «Поделиться». И

у нас появится пункт «Поделиться бета-отзывом». Отправляем скриншот разработчику.

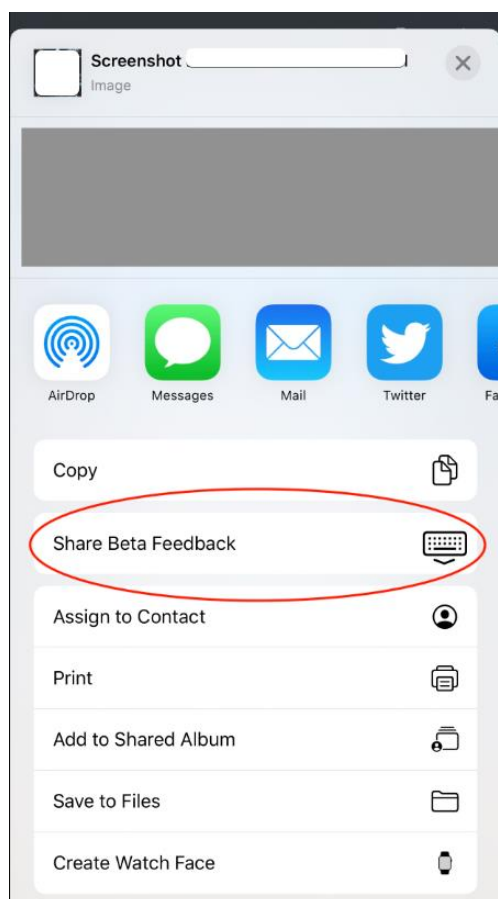
Ещё один способ оставить фидбэк — отправить его через сам TestFlight. Открываем TestFlight, нажимаем «Оставить фидбэк». Вносим необходимую информацию и прикрепляем файлы, если это необходимо.

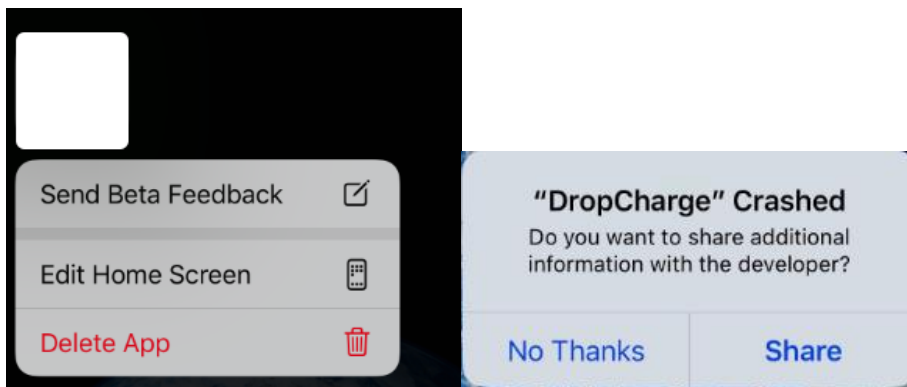
Последний способ фидбэка через TestFlight: на домашнем экране находим тестируемое приложение, нажимаем на иконку. Появляется опция «Оставить фидбэк». Далее процесс аналогичен.

Если тестируемое приложение крешится, то после краша появится контекстное меню с предложением отправить отчёт разработчикам. Нажимаем «Отправить».

Получение фидбэка

1. Заходим в App Store Connect, раздел **TestFlight**.
2. Переходим в **Отчёты**. Тут мы можем просматривать фидбэк, комментарии, скриншоты и краш-репорты.





Итак, мы с вами посмотрели все основные разделы в консолях Google Play и App Store. Узнали, как проводить тестирование с помощью этих площадок непосредственно перед релизом.

Размещение приложения в App Store

Давайте детальнее посмотрим на процесс размещения приложения в App Store. Глобально процесс и необходимая информация схожи с Google Play. [Пошаговый процесс загрузки приложения в AppStore](#)

Требования и гайдлайны сторов

Чтобы приложения прошли проверки в мобильных сторах, мы должны выполнять требования площадок. Остановимся на них подробнее.

Все актуальные требования всегда есть в документации. Полезно постоянно следить за новостями площадок, чтобы быть в курсе всех актуальных требований.

Google Play

У Google Play есть несколько документов, которые можно использовать, как набор правил, которым необходимо следовать перед выпуском приложения.

В первую очередь рассмотрим чек-лист под названием «Основное качество приложения» — [Core app quality | Android Developers](#).

Area	ID	Tests	Description
Navigation	VX-N1	CR-3	The app supports standard Back button navigation and does not make use of any custom, on-screen "Back button" prompts.
	VX-N2	CR-3	The app supports gesture navigation for going back / going to the home screen.
	VX-N3	CR-1 CR-3 CR-5	<p>The app correctly preserves and restores user or app state.</p> <p>The app preserves user or app state when leaving the foreground and prevents accidental data loss due to back-navigation and other state changes.</p> <p>When returning to the foreground, the app should restore the preserved state and any significant stateful transaction that was pending. Examples include: changes to</p>

Здесь очень хорошо структурированы основные проверки, на которые может обращать внимание Google Play при проверке приложения. Эти проверки относятся к категориям:

Визуальные проверки:

- Навигация
- Уведомления
- Пользовательский интерфейс и графика
- Визуальное качество
- Доступность

Функциональность:

- Аудио
- Медиа
- Обмен
- Фоновые службы

Производительность и стабильность:

- Стабильность
- Производительность
- SDK
- Потребление батареи

Конфиденциальность и безопасность:

- Разрешения

- Данные и файлы
- Аутентификация
- Сеть
- Шифрование

Google Play-функции:

- Соответствие политикам
- Информация о приложении
- Поддержка пользователей

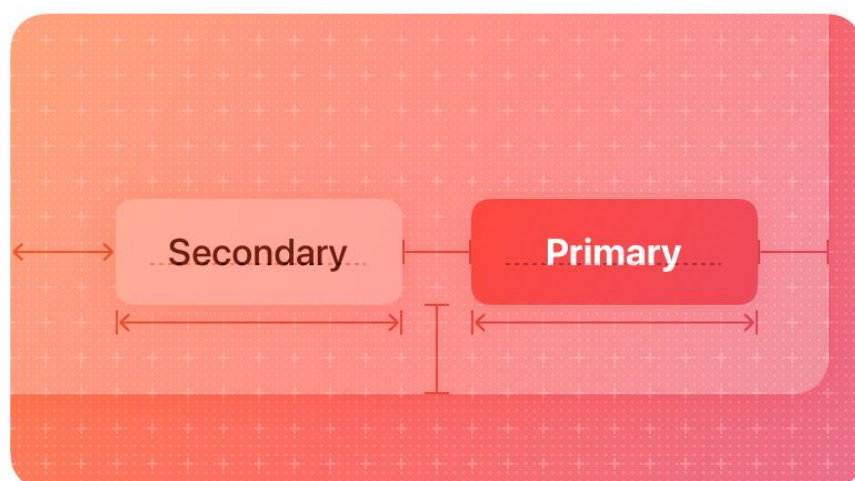
Также полезный документ, с которым можно ознакомиться (советую полистать в свободное время) — Центр правил для разработчиков ([Developer Policy Center](#)). Здесь указаны основные правила, которые следует учитывать при проектировании и разработке приложений. Кроме того, перечислены некоторые вещи, которые запрещены для добавления в приложение. Например, запрещённый контент, нарушение конфиденциальности, вредоносное ПО.

App Store

У App Store нет полноценного чек-листа с проверками, которые нужно провести перед релизом. Но есть хороший гайдлайн — Руководство по проверке приложений [App Store Review Guidelines — Apple Developer](#).

Ещё нём можно найти требования, касающиеся безопасности, функциональности, дизайна, производительности.

Также советую ознакомиться с руководством по интерфейсу приложения. App Store внимательно следит за тем, чтобы требования по дизайну соблюдались разработчиками — [Human Interface Guidelines — Design — Apple Developer](#).



Общие

Несмотря на то что документация у площадок разная, есть вещи, на которые стоит обращать внимание на любых платформах. Например, приложение не пройдет проверку, если в нём есть:

- Вредоносное ПО

Тут всё достаточно просто. Любое вредоносное ПО запрещено на любой официальной площадке. Apple и Google постоянно совершенствуют свои алгоритмы поиска приложений с вредоносным содержимым. К сожалению, это не значит, что все приложения в сторах на 100% безопасны, так как постоянно появляются новые типы вирусов. Также мы можем даже не знать, что в нашем приложении есть вредоносное ПО (когда, например, подключаем библиотеки из сомнительных источников).

- Использование чужой интеллектуальной собственности

Площадки не допустят использования чужих брендов, логотипов и прочей интеллектуальной собственности.

- В описании приложения упоминаются другие платформы

На странице магазина или внутри самого приложения не должно быть упоминаний других площадок. Скриншоты для страницы приложения в магазине должны быть сделаны на устройстве Apple для App Store и на устройстве Android для Google Play.

- Неточное описание приложения

Приложение не должно содержать информации, которая может ввести пользователей в заблуждение (например, описание несуществующей функции или имитация известного приложения). Скриншоты в магазине должны содержать только те функции, которые реализованы в приложении.

- Не соответствующий возрасту контент

При публикации приложения разработчику необходимо заполнить анкету для установки возрастных границ приложения. Если содержание не соответствует возрастному ограничению, приложение или обновления будут отклонены при публикации.

- Несоблюдение конфиденциальности

Платформы стремятся защищать конфиденциальность пользователей. Приложения, которые могут злоупотреблять использованием или неправильно использовать любых личные данные, строго запрещены.

- Неправильно настроенная монетизация

Площадки строго следят за выполнением требований по правильному оформлению покупок внутри приложения. Возможно, вы слышали об относительно недавнем случае суда между Apple и Epic Games. Проблемы начались в 2020 году, после удаления игры Fortnite из App Store. Все приложения в App Store обязаны использовать платёжную систему Apple в приложениях для iPhone, комиссия которой составляет от 15 до 30% от каждой транзакции. Epic Games нарушила это требование, после чего Fortnite была удалена из стора, а Epic Games подала жалобу на Apple как на монополистов. Дело ещё продолжается. Тем не менее нам нужно внимательно следить за меняющимися правилами площадок в отношении монетизации, чтобы наше приложение не было удалено из сторов.

- Спам и неработающий функционал

Ни одна площадка не пропустит приложение, которое сильно спамит или является неработающим. Необходимо тщательно тестировать наше приложение, перед тем как открывать его на всех пользователей.

Выводы

Сегодня мы:

- Подробно рассмотрели две основные площадки для мобильных приложений
- Рассмотрели процесс тестирования с их использованием
- Познакомились с основными гайдлайнами площадок

Домашнее задание

В качестве необязательного домашнего задания попробуйте найти открытый бета-тест мобильного приложения какой-либо компании (*через поиск в интернете*).



1. Изучите условия и примите участие в бета-тестировании.
2. При нахождении проблем в приложении, сообщите разработчикам.

Пример программы

Как стать участником бета-тестирования

Принять участие в бета-тестировании может любой пользователь, оплативший подписку на сайте сервиса.

⚠ Внимание. В программе тестирования не могут участвовать пользователи с аккаунтами, которые были зарегистрированы через номер телефона, — с помощью таких аккаунтов нельзя авторизоваться и оплатить подписку на сайте сервиса.

1. Перейдите на [страницу бета-версии приложения](#)  и подтвердите свое участие в программе.
2. Если у вас уже установлена общедоступная версия Яндекс Музыки, дождитесь ее автоматического обновления или [загрузите обновления вручную](#). Если приложение еще не установлено, [скачайте его](#)  в магазине Google Play. На устройство будет установлена бета-версия приложения.
3. Пользуйтесь приложением и тестируйте новые функции (их перечень см. в описании приложения в Google Play). О найденных ошибках сообщайте в техническую поддержку — для этого в настройках приложения нажмите на строку **Написать разработчикам**.

[Написать в службу поддержки](#)

< [Предыдущая](#)
[Особенности бета-версии ...](#)

[Следующая](#) >
[Как выйти из программы б...](#)

Источник: [Как стать участником бета-тестирования - Мобильная Яндекс Музыка. Справка](#)

Такие программы бывают платными и бесплатными. Они, как правило, ограничены в длительности. Но когда одна программа перестает работать, можно найти другую.

💡 Если вы сможете присоединиться к программе, то получите отличную практику тестирования и + 1 хороший кейс для портфолио!

Используемая литература

1. [Библиотека \(программирование\) — Википедия](#)
2. [Google Play SDK Index | Android Developers](#)
3. [Xcode 14 Overview — Apple Developer](#)
4. [Как провести открытое, закрытое или внутреннее тестирование — Справка — Play Console](#)
5. [Core app quality | Android Developers](#)
6. [App Store Review Guidelines — Apple Developer](#)