

Emulator-Simulator: особенности и отличия. Особенности и функционал Android Studio SDK.

Эмуляторы и симуляторы — это программные инструменты, которые имитируют различные операционные системы. Поскольку поисковые системы быстрее индексируют сайты, имеющие мобильные версии по сравнению с сайтами, ориентированными только на десктоп, перед выпуском приложения на рынок критически важно решить все серьезные проблемы. Фундаментальные принципы тестирования для мобильных и десктопных приложений одинаковы, но для мобильных все же есть нюансы.

Мобильный эмулятор — это программа, которая аппаратно и программно имитирует работу реального необходимого устройства. Эмуляторы настраиваются в виртуальной среде. Настроив виртуальную среду, команда разработки получает доступ к полноценному непрерывному функциональному тестированию.

Примеры аппаратных компонентов, которые можно протестировать с помощью эмулятора, включают производительность батареи, оперативную память (RAM), процессор, камеру и клавиатуру. С программной точки зрения можно говорить об обмене сообщениями, безопасности и просмотре веб-страниц. Если веб-сайт тяжелый, эмулятор может не справиться с поддержкой желаемого темпа из-за технических характеристик телефона.

Мобильный симулятор — это программа, созданная в виртуальной среде, которая также копирует конфигурацию и поведение реального целевого устройства. По описанию звучит очень похоже на эмулятор, потому что обе программы служат аналогичной цели. Довольно часто эти названия используются взаимозаменяемо, но разница все же есть, и лежит она в плоскости аппаратной части.

Симуляторы не имитируют аппаратную часть устройств. Только с помощью эмуляторов можно протестировать, например, работу микрофона или динамика. Симуляторы созданы для того, чтобы проверять только программную часть, которая существует в производственной среде мобильного устройства.

Типы мобильных тестов

Есть множество мобильных приложений и несколько видов тестирования этих приложений. Самый распространенный вид — это функциональное тестирование, целью которого является подтверждение того, что приложение работает в соответствии с требованиями.

Другие виды:

- Тестирование совместимости (Compatibility Testing) — с его помощью проверяется, совместимо ли мобильное приложение с разными операционными системами и версиями.
- Тестирование установки (Installation Testing) — направлено на проверку удобства процесса установки и удаления приложения для пользователя. С его помощью можно удостовериться, что обновления не содержат ошибок и исследовать, что происходит с приложением, если пользователь его не обновляет.
- Проверка помех (Interruption Testing) — целью этого вида тестирования является проверка поведения приложения в состоянии приостановки, а затем возврата в предыдущее состояние. Например, пользователь смотрит видео и в этот момент ему позвонили. Приложение, проигрывающее видео, должно перейти в фоновый режим, а после вернуться в предыдущее состояние до звонка.
- Нагрузочное тестирование (Load Testing) предназначено для проверки того, как отвечает приложение, если к нему обращаются одновременно очень много

пользователей. Пользователи пытаются одновременно установить, удалить и использовать приложение, чтобы посмотреть, не даст ли оно сбой или медленную загрузку.

- Тестирование локализации (Localization Testing) проверяет качество адаптации под определенный географический регион. Например, тест может включить верификацию определенного языка, соответствие требованиям местного законодательства или временной зоны.
- Тестирование производительности (Performance Testing) — проверяет производительность приложения под определенными условиями загруженности, таких как проверка скорости приложения, плохое покрытие сети и недостаток доступной памяти.
- Тестирование безопасности (Security Testing) — проверяет уязвимость аутентификации, авторизации, взломов и безопасности данных.
- Юзабилити-тестирование (Usability Testing) проверяет, удобно ли пользователям использовать приложение. Цель в том, чтобы удостовериться, легко ли с помощью приложения выполнить определенную задачу.

Инструменты и фреймворки для автоматизированного мобильного тестирования

Тенденция использования инструментов автоматизации тестирования мобильных устройств набирает обороты. Причины тому — быстрый фидбек, CI/CD, более точные тесты и увеличенная производительность. Вот список некоторых новых инструментов:

- [Appium](#) — опенсорсный фреймворк для автоматизации для тестирования нативных, гибридных и мобильных приложений.
- [TestProject](#) — бесплатная платформа для автоматизации тестирования мобильных приложений (Android и iOS), веба и API.
- Detox — опенсорсный комплексный фреймворк для тестирования, работающий на реальном устройстве или симуляторе.
- [Espresso](#) — нативный фреймворк для тестирования Android-приложений.
- [XCUITest](#) — фреймворк для тестирования iOS-приложений.

В каких случаях для тестирования можно использовать эмулятор или симулятор, а в каких — нужно реальное устройство

В использовании эмуляторов, симуляторов и реальных устройств для тестирования есть свои сильные стороны. Выбранный путь тестирования зависит от степени риска и фазы. К примеру, если единственный риск связан с рендерингом веб-сайта, тогда лучше выбрать эмулятор или симулятор.

Оба варианта приемлемы, если необходимо проверить важную для конечного пользователя отзывчивость. Кроме того, стоимость такого тестирования будет более оптимальной по сравнению с использованием реального устройства. Однако некоторые дефекты можно найти только на реальном устройстве. Если основной риск связан с такими действиями, как свайп или перетаскивание приложения, то, пожалуй, будет лучше не выбирать эмулятор или симулятор.

Организации предпочитают для тестирования использовать только реальные устройства, но они стоят денег. Из соображений экономии бюджета мы стараемся использовать комбинацию из эмуляторов, симуляторов и реальных устройств. Эмпирическое правило гласит — использовать эмуляторы для тестирования аппаратной части и симуляторы для тестирования программной части. Эмулятор может проверить, как работает наше приложение, используя CPU или разное выделение памяти.

Хороший тому пример — тестирование встроенного программного обеспечения с использованием эмулятора. Встроенное программное обеспечение в нашем телефоне

повсюду. Это программное обеспечение, которое обеспечивает низкоуровневый контроль за аппаратной частью мобильных телефонов.

Симулятор может гарантировать высокую производительность внутреннего приложения во время взаимодействия с внешним приложением, например, при обмене данными между приложениями. Эмуляторы и симуляторы полезны для тестирования при широком диапазоне условий.

Несмотря на это, целесообразно, когда они не служат заменой реальным устройствам. Тестирование на реальном устройстве очень полезно, когда производится на важных этапах перед продакшеном. Чем ближе мы подходим к продакшену, тем серьезнее становится наш выбор. Организация может использовать виртуальную платформу на ранней стадии, которая включает циклы разработки. Юнит-тестирование и моук-тестирование выполняются потому, что разработчики изолируют куски кода.

Основное различие между эмулятором и симулятором:

- эмулятор — это программное или аппаратное обеспечение, которое позволяет одной компьютерной системе работать аналогично другой системе,
- симулятор — это программное или аппаратное обеспечение, которое может имитировать сценарий реальной жизни, чтобы обеспечить требуемые результаты без какой-либо опасности.

Вывод

Эмуляторы и симуляторы помогают получить быструю обратную связь. Поскольку реальные устройства довольно дорогостоящи, их оставляют для тестирования на более поздних этапах, таких как интеграционное и функциональное тестирование.

Возможности Android Studio SDK

Android SDK — это дополнительный набор инструментов Android Studio, которые помогают написать код, запустить тестирование и отладку, проверить работу приложения, протестировать на различных версиях операционной системы и оценить результат в реальном времени.

Интерфейс SDK предусматривает:

- кнопки, отвечающие за открытие и сохранение проектов;
- переходы пошагового типа;
- масштабирование окон;
- выбор apk файлов;
- возможность выбора эмулируемого железа и версии операционной системы;
- запуск и паузу эмуляции;
- настройки;
- меню помощи.

ADB (или *Android debug bridge*) — самый сильный и многофункциональный инструмент, используемый для работы с android приложениями. ADB — это инструмент командной строки, который позволяет общаться с подключенным к компьютеру девайсом. С помощью него можно делать следующее:

- Получать и менять все настройки системы устройства
- Делать скриншоты экрана девайса и записывать видео

- Симулировать касания экрана и нажатие кнопок девайса
- Работать с файловой системой
- Удалять, устанавливать и запускать приложения с определенными входными параметрами
- Забыть про USB и подключаться к девайсам через Wi-fi

Android девайс под капотом — Unix система, получив доступ к девайсу через adb shell вы получаете все те же возможности, которые у вас были бы при работе с Unix, поэтому знания в этой области необходимы, чтобы чувствовать себя комфортнее.

А в рамках тестирования QA специалист может подложить изображение в файловую систему и напрямую вызвать этот триггер без использования других приложений:

```
adb shell am start -a android.intent.action.VIEW -c android.intent.category.DEFAULT -e pathToImage your/image/path -n com.example.test
```

“-e” — это команда, которая передает аргумент по ключу “-e key value”

Плюс ко всему, через adb также легко тестировать deeplink’и. С этим поможет команда:

```
adb shell am start -a android.intent.action.VIEW -d "myAppScheme://test" com.example.test
```

где “myAppScheme” и “test” — это значения, которые умеет обрабатывать активности (их обычно указывают в Манифесте в блоке <activity>), а “com.example.test” — это имя пакета вашего приложения (тоже хранится в манифесте).

Команда выше симулирует переход по ссылке *myAppScheme://test*, а система Android проходит по манифесту и ищет активности, которые знают, как такие ссылки обрабатывать.

Мы использовали [Appetize.io](https://appetize.io) в онлайн режиме для тестирования ios, так как доступ ограничен к app, мы открыли веб-приложение ozon.ru и протестировали его на девайсе Iphone 8+ ios 12.4 модель A1863.

Особенностями функционала данной программы является возможность развертывания эмулятора без загрузки программы, в браузере. Данная программа в бесплатной версии не позволяет загрузить нативное приложение, что делает невозможным проведение тестирования интересующего нас приложения. Мы тестировали приложение в браузере. Провели функциональное, нефункциональное, локализационное тестирование, а также проверку на прерывание и зависимости от сети.

Мы использовали Android Studio SDK Tools для тестирования приложения AliExpress Samsung Galaxy Nexus API30 версия Android 11.0.

Особенностями функционала программ эмуляторов является возможность тестирования: размера экрана и его расположения, симуляция звонка, закрытие и открытие приложения, тестирование функционального состояния приложения: регистрация/авторизация, заполнение профиля, процесс покупки и оплата заказа (интеграционное тестирование) и тд.