

Тестирование ПО

Особенности тестирования мобильных приложений

Moblie vs web & desktop - 1

- Постоянные прерывания в работе приложения: сворачивание, блокировка, входящий звонок или сообщение, уведомления, обновления приложений, выключение или перезагрузка, выгрузка системой из ОЗУ, разряд АКБ, подключение зарядки или других устройств, переход в энергосберегающий режим и режим ожидания, включение и отключение функций необходимых устройству (gps, bluetooth, связь), использование/отзыв разрешений, подключение/отключение карты памяти/симки/АКБ, платежи NFC, принудительная остановка);
- Работа в беспроводной сети с изменяющейся стабильностью и скоростью приема сигнала, переключение между сотовой связью и wi-fi;
- Уведомления;
- Взаимодействие с web view;
- Использование вертикальной и горизонтальной ориентации, повороты;
- Распространение приложений через маркеты;
- Необходимо соответствие гайдлайнам систем;
- Большое внимание уделяется UI и UX;
- В случае Android большая фрагментация устройств и прошивок со своими особенностями;
- Тач-интерфейс, мультитач, жесты;

Moblle vs web & desktop - 2

- Множество каналов ввода: стоковая клавиатура, сторонние клавиатуры, хардовые клавиатуры, голос, жесты и т. д.;
- Биометрия;
- Повышенные требования к энергопотреблению и использованию аппаратных ресурсов;
- Ситуации установки и обновления при нехватке памяти, переноса приложения на карту памяти и обратно;
- Важность инсталляционного тестирования, особенно обновлений, т.к. в условиях высококонкурентного мобильного рынка приложения обновляются часто, чтобы предоставить пользователям новые функции как можно скорее и при этом должны проходить бесшовно;
- GPS и локализация;
- Размер дисплея, челка и вырез под фронтальную камеру;

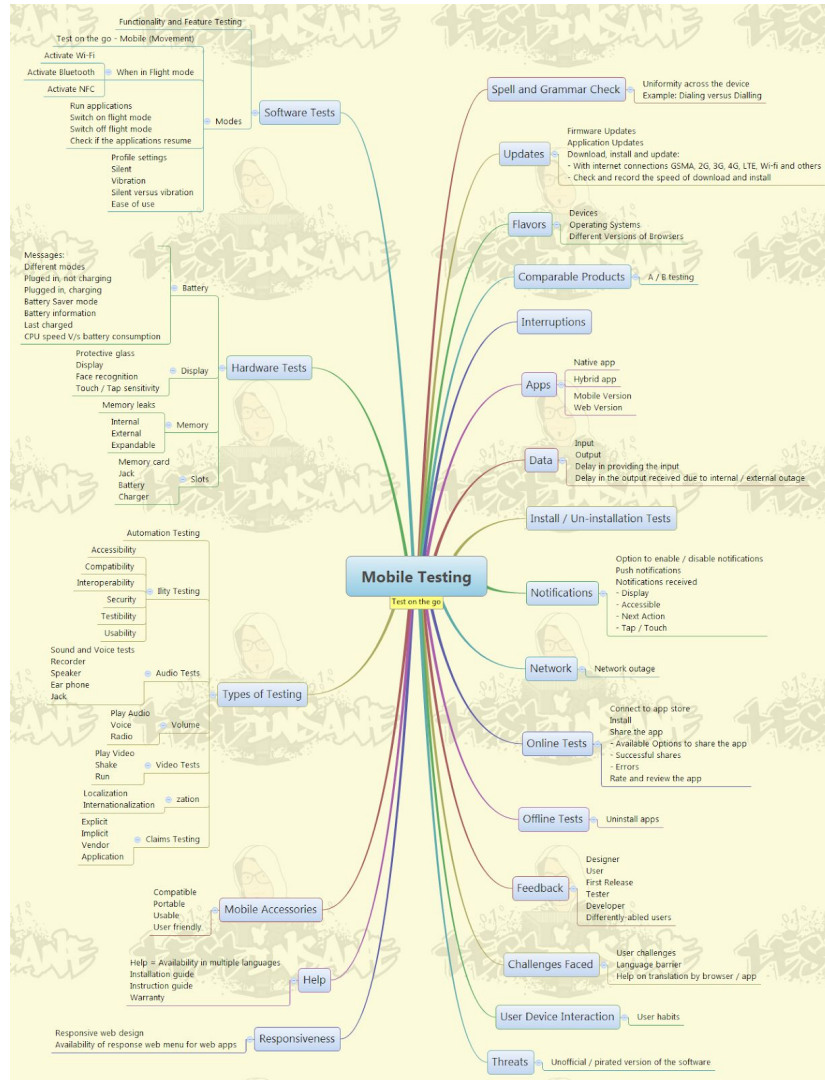
Native vs Hybrid vs PWA

IMAGINARYCLOUD 

What's the best app for you?

	Native apps	Hybrid apps	PWAs
Language	<ul style="list-style-type: none">• iOS - Swift• Android - Java and Kotlin	HTML, CSS and JavaScript	HTML, CSS and JavaScript
Process	Hardest and slowest	Mostly easy and quick	Easiest and quickest
Cost	Highest priced	Affordable	Most accessible
UX and Performance	Quickest and most responsive	Dependent of internet speed	Dependent of internet speed
Security and Maintenance	Highest security and maintenance	Safe maintenance	Lowest security and maintenance
Platform	iOS or Android	WebView	Web browser
Access to mobile features	Yes	Limited	No
SEO	Doesn't apply	Fast and responsive. No internet required	Dependent of internet speed
Distribution	App stores	App stores	URL





Основной чек-лист мобильного тестирования

- Функциональное тестирование;
- Тестирование совместимости;
- Тестирование безопасности;
- Тестирование локализации и глобализации;
- Тестирование удобства использования;
- Стрессовое тестирование;
- Кросс-платформенное тестирование;
- Тестирование производительности.

Функциональное тестирование

В данном пункте нам важно убедиться, что наш продукт соответствует нужной функциональной спецификации, упомянутой в документации по разработке.

- Установка/удаление/накатка версий;
- Запуск приложения (отображение Splash Screen);
- Работоспособность основного функционала приложения;
- Корректное отображение ошибок;
- Работа с файлами (отправка/получение/просмотр);
- Тестирование тайм-аутов;
- Тестирование заглушек (не соединения с интернетом/нет, например, товаров и т.д.);
- Тестирование pop-up, алертов;
- Тестирование WebView;
- Скролл/свайп элементов;
- Тестирование PUSH уведомлений;
- Сворачивание/разворачивание приложения;
- Разные типы подключений (сотовая связь/Wi-Fi);
- Ориентация экрана (альбомная/портретная);
- Темная/светлая темы;
- Реклама в приложении;
- Шаринг контента в соц. сети;
- Работа приложения в фоне;
- Пагинация страниц;
- Политики конфиденциальности и прочие ссылки на документы.

Тестирование совместимости

Тестирование совместимости используется, чтобы убедиться, что ваше приложение совместимо с другими версиями ОС, различными оболочками и сторонними сервисами, а также аппаратным обеспечением устройства.

- Корректное отображение гео данных;
- Информации об операциях (чеки и т.д.);
- Различные способы оплаты (Google Pay, Apple Pay);
- Тестирование датчиков (освещенности, температуры устройства, гироскоп и т.д.);
- Тестирование прерываний (входящий звонок/смс/push/будильник/режим «Не беспокоить» и т.д.);
- Подключение внешних устройств (карта памяти/наушники и т.д.).

Тестирование безопасности

Данная проверка нацелена на поиск недостатков и пробелов с точки зрения безопасности приложения.

- Тестирование разрешений (доступ к камере/микрофону/галерее/и т.д.);
- Данные пользователя (пароли) не передаются в открытом виде;
- В полях, с вводом пароля и подтверждением пароля, данные скрываются астерисками.

Тестирование локализации и глобализации

Тестирование интернационализации/глобализации приложения включает тестирование приложения для различных местоположений, форматов дат, чисел и валют, а также замену фактических строк псевдостроками. Тестирование локализации включает тестирование приложения с локализованными строками, изображениями и рабочими процессами для определенного региона.

- Все элементы в приложении переведены на соответствующий язык;
- Тексты защиты внутри приложения и пользователь в настройках приложения может выставить необходимый язык;
- Тексты зависят от языка в системных настройках;
- Тексты приходят с сервера;
- Корректное отображение форматов дат (ГОД - МЕСЯЦ - ДЕНЬ или ДЕНЬ - МЕСЯЦ - ГОД.);
- Корректное отображение времени в зависимости от часового пояса.

Тестирование удобства использования

Тестирование удобства использования помогает удостовериться в простоте и эффективности использования продукта пользователем, с целью достижения поставленных целей. Иными словами, это не что иное, как тестирование дружелюбности приложения для пользователя.

- Корректное отображение элементов на устройствах с различными разрешениями экранов;
- Все шрифты соответствуют требованиям;
- Все тексты правильно выровнены;
- Все сообщения об ошибках верные, без орфографических и грамматических ошибок;
- Корректные заголовки экранов;
- В поисковых строках присутствуют плейсхолдеры;
- Неактивные элементы отображаются серым;
- Ссылки на документы ведут на соответствующий раздел на сайте;
- Анимация между переходами;
- Корректный возврат на предыдущий экран;
- Поддерживаются основные жесты при работе с сенсорными экранами (swipe back и т.д.);
- Пиксель-перфект.

Стрессовое тестирование

Стрессовое тестирование направлено на определение эффективности производительности приложения в условиях повышенной нагрузки. Стресс-тест в этом контексте ориентирован только на мобильные устройства.

- Высокая загрузка центрального процессора;
- Нехватка памяти;
- Загрузка батареи;
- Отказы;
- Низкая пропускная способность сети;
- Большое количество взаимодействий пользователя с приложением (для этого может понадобиться имитация реальных условий состояния сети).

Тестирование производительности

Если пользователь устанавливает приложение, и оно не отображается достаточно быстро (например, в течение трех секунд), оно может быть удалено в пользу другого приложения. Аспекты потребления времени и ресурсов являются важными факторами успеха для приложения, и для измерения этих аспектов проводится тестирование производительности.

- Время загрузки приложения;
- Обработка запросов;
- Кэширование данных;
- Потребление ресурсов приложением (например расход заряда батареи).

Как получить приложение на тест?

Android:

- Установить .apk или .aab, который нужно разархивировать;
- Из CI-агента. Jenkins/TeamCity может присылать ссылку на билд в tg-канал или можно забрать его вручную;
- Сбилдить в Android Studio самому из нужной ветки;
- Открытые и закрытые бета-тестирования приложений в Google Play;

iOS:

- сбилдить в Xcode;
- установить ipa, будучи в группе разработки;
- внутреннее и внешнее тестирование в TestFlight;
- appbox

Почему приложение может не попасть в маркет?

- **App Completeness** - не работающие приложения, приложения с крашами и багами;
- **Inaccurate Metadata** - подробно и точно описывайте функционал приложений, прикрепляйте правильные скриншоты;
- **Incomplete Info** - важно рассказать о вашем продукте все, что стоит знать ревьюерам. Проверяйте актуальность тестовых учеток перед сабмитом. Если вы что-то экспериментируете в билде, расскажите об этом команде ревью;
- **Unusual Interface** - приложения должны работать так, как пользователи от них этого ждут;
- **Web Content Aggregators** - сайты обернутые в iOS приложение обычно не принимаются;
- **Similar App Submission** - приветствуются уникальные приложения;
- **Misleading Offers** - не обещать того, чего нет;
- **Not Enough Value** - не нужно делать десяти тысячный калькулятор и миллионный фонарик, их достаточно, также не нужно делать приложение, которым будут пользоваться три человека, ваше приложение должно нести в себе нужный широкому пользователю, уникальный функционал;
- Не прячьте части функционала приложения от ревьюеров;
- Не пытайтесь обойти комиссию платформ и прятать 3rd-party эквайринг;
- Не вводите пользователя в заблуждение;
- Не запрашивайте лишние доступы, объясняйте зачем нужны доступы, которые запрашиваете.

Тестирование требований -1

Общие вопросы по приложению

- Для каких платформ будет разрабатываться приложение и какие версии ОС будут поддерживаться? Необходимо всегда помнить о минимальной поддерживаемой версии ОС. Иначе можно обнаружить, что функциональность не работает у пользователей, когда задача уже закрыта.
- На каких устройствах необходимо проверить приложение? Например, приложение должно работать как на смартфонах, так и на планшетах. Или должна быть поддержка Apple Watch.
- Какую ориентацию экранов поддерживает приложение? Портретная и/или ландшафтная? Неприятный момент: если смена ориентации экрана хорошо работает на смартфонах, это не значит, что всё будет так же на планшетах.
- На каких девайсах приоритетнее всего смотреть? На ваших девайсах приложение может идеально работать, а вот у заказчика на любимом (вставьте китайский android-смартфон) все разъехалось...
- Должен быть идеальный пиксель-перфект или допускаются некоторые погрешности? Привет, тестирование на соответствие макетам! Ещё неплохо уточнить, должна ли растягиваться верстка или под каждый размер экрана должны быть свои макеты?
- Существуют ли другие клиентские приложения? Например, есть админка, которая внезапно начнет удалять или добавлять элементы. Или веб-версия, которая существует уже в продакшене. Главное - узнать об этом как можно раньше.
- Есть ли какие-то внешние устройства, которые могут повлиять на логику мобильного приложения?
- Какая целевая аудитория у приложения? Все пользователи в Play Market/AppStore или 50 человек в компании заказчика?

Тестирование требований -2

Разбор приложения по экранам

- Состав экрана и возможные действия на нем. Из каких элементов состоит экран? Какие действия можно совершить? Какие состояния экрана возможны? Какие есть переходы и на какие экраны они ведут? Что должно отображаться при возврате на этот экран? Ответы на эти вопросы необходимо найти, а лучше зафиксировать в документации.
- Взаимодействие с сервером на экране. Какие запросы идут на экране? Понимание того, какие запросы на сервер отправляются на экране, поможет выявить такие требования, которые не сможет реализовать сервер по тем или иным причинам.
- Активность по таймеру. Например, отправляется важная аналитика раз в две минуты или идет обновление данных.
- Кэширование данных. Загрузка одних и тех же данных при каждом входе на экран может раздражать пользователей. При кэшировании необходимо продумать, когда должна обновляться информация на экране? Когда должен очищаться кэш?
- Заглушки. Что отображается, если данных нет? Пустой экран - неинформативный для пользователя. А съехавшая заглушка может быть поводом для недовольства заказчика.
- Поведение в случаях ошибки. Что должно отображаться, если произошла ошибка? Например, отсутствие интернета, серверная или незадокументированная ошибка.
- Медленная загрузка данных. Что должно происходить при медленной загрузке данных? Лоадеры, блокировка действий, кастомные анимации - всё должно быть продумано.
- Действия, которые влияют на поведение других экранов. Как действие на одном экране повлияет на поведение на других? Сквозные действия - опасная штука. Особенно, если разработка и тестирование идет по экранам или по отдельным фичам. Тут без регрессии обойтись сложно. Поэтому на некоторых проектах, прежде чем писать тест-кейсы, мы строим матрицу влияния для новых фич.
- Обновление данных на экране. Когда происходит обновление? Есть следующие варианты и они могут сочетаться:
 - Каждый раз при открытии экрана (данные живут только пока у пользователя открыт экран).
 - Каждый раз при запуске приложения (данные живут только пока у пользователя запущено приложение).
 - По pull-to-refresh'у/по специальной кнопке обновления/по таймеру (данные хранятся в локальном хранилище устройства и при перезапуске приложения восстанавливаются).

Тестирование требований -3

- Локализация
- Разрешения
- Списки
- Формы ввода
- Учетные записи
- Регистрация/авторизация через соцсети
- Ролевая модель
- Карта
- Чаты
- Отправка файлов на сервер и скачивание на устройство
- Внешние устройства
- Аудиоплеер/видеоплеер
- Оплата банковской картой
- Время, календарь, таймер
- Нотификации

Шпаргалка: <https://habr.com/ru/companies/mobileup/articles/336992/>

Тестирование дип линков

Дип линкинг (deep linking) позволяет конечному пользователю с помощью ссылки открыть страницу с нужным контентом внутри мобильного приложения, минуя его домашнюю страницу и минимизируя трату времени на поиск необходимого контента.

В случае, если приложение не установлено на девайсе, отложенный дип линкинг (deferred deep linking) позволяет пользователю открыть нужную страницу внутри мобильного приложения даже после его установки.

Проблемы, с которыми сталкиваются разработчики и тестировщики:

- Поддержка различных технологий работы с дип линками и разработка линков, которые будут работать во всех мобильных браузерах;
- Интеграция с соц.сетями и мессенджерами;
- Встроенные браузеры (in-app browsers);
- Отслеживание атрибуции (attribution);
- Предотвращение мошенничества;
- Некоторые end-to-end сценарии очень тяжело (или невозможно) автоматизировать;
- Постоянная поддержка.

Покрытие устройств

Статистика -> Целевая аудитория -> Особенности приложения

- Производители. Посмотреть лидирующих (по трафику) вендоров можно на Statcounter. Определиться с разработкой версии для Huawei без гугл-сервисов;
- Соотношение сторон экрана. При выборе постарайтесь захватить оба значения ближе к краю (из используемых) и среднее;
- Размер. Нужны ли планшеты, а также отдельно варианты для Android и для iOS, т.к. у них немного отличается и подход и обозначение;
- Особенности. На iOS могут быть нюансы работы нативной «Назад» у моделей с монобровью/челкой, хотя физически она и расположена в зоне статус-бара. Любое приложение с ландшафтной ориентацией и полным использованием экрана (например плеер) также желательно посмотреть на моделях с бровью. Если у вас не веб, а приложение, рассчитанное не на премиум-сегмент, да еще и с записью данных на устройство - работа с SD-картой иногда вызывает вопросы, включаем в список.

Tips & Tricks

- Используйте принципы эвристики и мнемоники — они помогают удержать в голове все аспекты, которые нужно учесть при тестировании фичи или приложения. (ex: I SLICED UP FUN)
- Скриншоты, логи и видео — лучшие аргументы тестировщика!
- Используйте «обезьянок» для поиска крашей и зависаний, пока вы тестируете функционал более осмысленно.
- Для того чтобы чувствовать себя более комфортно перед релизами, используйте страховочную сеть в виде бета-версий.
- Очень удобно иметь отладочное меню с функциями, которые облегчают работу разработчикам и тестировщикам (особенно команде автоматизации). Например, с возможностью симулировать ответы от сервера, открывать определённых пользователей, выставлять определённые флаги, очищать и терять сессии, обнулять кэш.
- Девелоперское меню iOS и Android — ваш лучший друг.
- Если приложение поддерживает портретный и ландшафтный режим, то уделите смене ориентации девайса особое внимание.
- Переходите между скринами много раз.
- Не пренебрегайте тестированием на эмуляторах и симуляторах
- Заполняйте оперативную память девайса перед запуском приложения.
- Запускайте приложение под отладчиком.