

Mobile Testing

Выделяют основных три вида:

- Нативное приложение
- Веб-приложение или мобильный сайт
- Гибридное приложение

Нативные приложения

Нативные мобильные приложения — приложение под определенную платформу доступное через маркетплейс (Google Play, AppStore и т.д.). Еще одно важно отличие - автономная работа в режиме оффлайн. Яркий пример мобильные игры. Это и самый распространенный и дорогой в разработке вид, так как создается отдельно для каждой ОС (iOS, Android или другие). Когда мы говорим о мобильных приложениях, чаще всего имеем в виду именно нативные. Pokemon Go, Spotify и многие другие являются именно нативными приложениями.

Преимущества

- практически вся функциональность доступна в оффлайне
- скорость работы выше других типов моб. приложений
- полный доступ к функциям девайса (FaceID, отпечаток пальца, камера и т.п.)
- распространение через магазины

Недостатки

- правки багов доезжают только при релизе следующей версии
- тестирование на каждой платформе, больше денег и времени для разработки
- занимают больше памяти
- загрузка и обновление с привлечением юзера

Веб-приложения

Веб-приложения представляют собой адаптированные веб-сайты, которые открываются через браузеры. Пользователь не скачивает приложение и не хранит его на своем устройстве. Если его «скачивают», скорее всего, речь идет о том, что оно добавляется в закладки браузера. Одним из самых распространенных подвидов считают PWA — прогрессивные веб-приложения, которые, по сути, являются нативными приложениями внутри браузера. Сложно выделить примеры, чтобы не ошибиться. Разные источники приводят в пример Google Maps, программы Microsoft Office.

Преимущества

- правки багов приезжают быстрее
- тестирование проводится в браузере и не сильно завязано на ОС/модель телефона/платформу; одна версия для всех платформ
- не требуется тестировать установку, удаление и обновление; обновление на сервере

Недостатки

- ограниченный доступ к функциональности девайса (FaceID, отпечаток пальца, камера и т.п.); нет возможности использовать функции мобильного устройства
- не работают в оффлайне
- нельзя загрузить из магазина

Гибридные приложения

Устанавливаются через маркетплейс, а отображаются внутри приложения как веб-сайт. Часто это приложения супермаркетов, недорогих доставок еды. Название говорит само за себя: это веб-приложения, которые выглядят как нативные и имеют их признаки. В частности, иконки на рабочем столе мобильного устройства пользователя, хорошую производительность и возможность работать в автономном режиме. Среди гибридных приложений выделяют Uber, Evernote. Некоторые источники относят к гибридам и Instagram.

Плюсы

- мультиплатформенные, т.е. реализация на всех ОС одна

- правки багов приезжают быстрее, т.к. по сути функционал это встроенный веб-сайт в приложение
- могут использовать большинство функций девайса
- дешевле в разработке, чем нативные

Минусы

- не работают в офлайне
- ограниченный доступ к функциональности девайса
- низкая скорость работы
- непрактичны ввиду долгого обновления фреймворка
- незащищенность

Требования к разработке приложений и их использование в тестировании:

Android разработка и требования к приложениям:

Android – операционная система, которая открыта для всех: разработчиков, дизайнеров и производителей устройств. Компания Google предъявляет ряд требований к безопасности, функциональности и внешнему виду приложений. Полная спецификация, ресурсы, лучшие практики и требования, расположены в открытом доступе на сайте <https://developer.android.com/>:

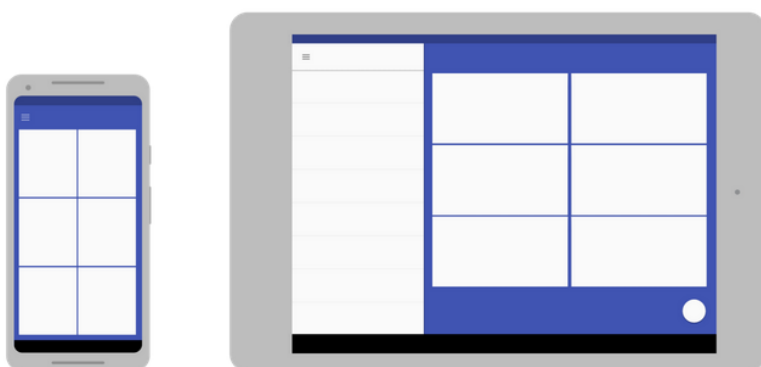
- Анимация;
- Безопасность;
- Цвета и их состояния;
- Изображения (поддерживаются форматы gif, png, jpg и webp);
- Слои (есть возможность создать кастомные);
- Требования к меню (меню опций, контекстное меню, подменю);
- Требования к строкам (одиночные, массив строк, множественные);
- Стили;

- Шрифты.

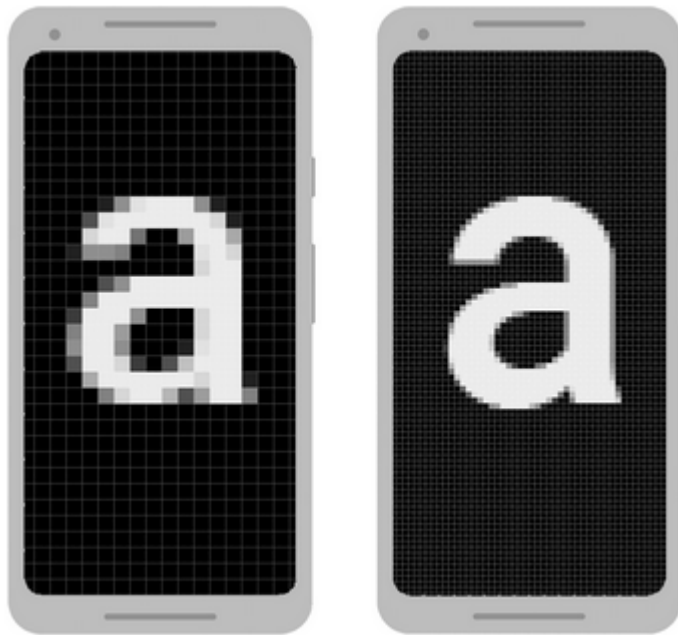
Так же не стоит забывать, что андроид устройств множество, разных форм и размеров, не все устройства смогут поддерживать современные функции. При разработке приложения, стоит уделить внимание гибкости интерфейса, способного подстраиваться под разные конфигурации экранов.

Компановка экрана:

- **Альтернативное меню и слои** - разработка альтернативного меню для других устройств;



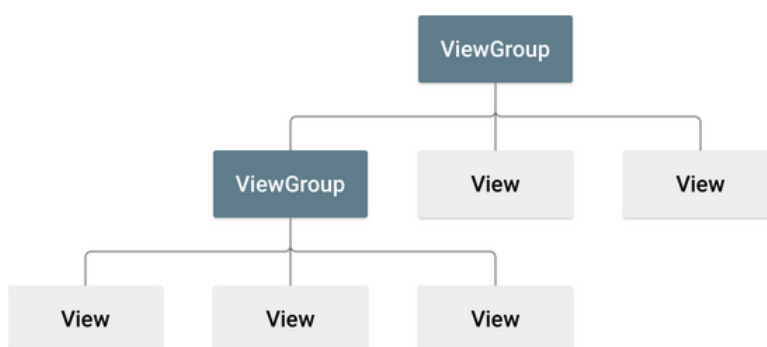
- **Растягивающиеся элементы** - может привести к артефактам и искажениям изображения. Для избежания проблемы, используется точечная растяжка (масштабируются не все объекты);
- **Плотность пикселя** - на разных устройствах один и тот же контент может выглядеть по-разному, даже если они одного размера, но с разной плотностью пикселей;



- **Использование векторной графики** (создание картинок, иконок и тд разного разрешения);
- Не стандартные **устройства**. Складные устройства обычно имеют несколько дисплеев (или даже комбинациями дисплеев), которые становятся активными для разных состояний складывания устройства;

Требования к пользовательскому интерфейсу:

- Иерархия слоёв меню;



- Размеры отступы и рамки;
- Компановка;

Linear Layout



Relative Layout



Web View

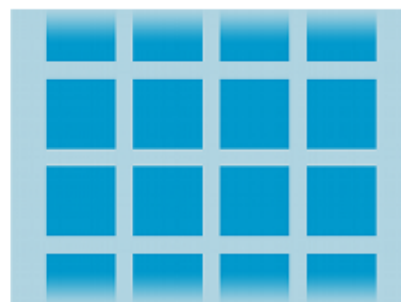


- Построение списка с помощью адаптера;

List View

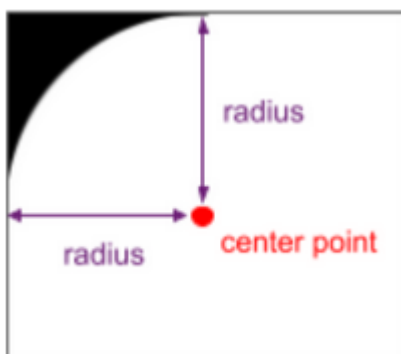


Grid View

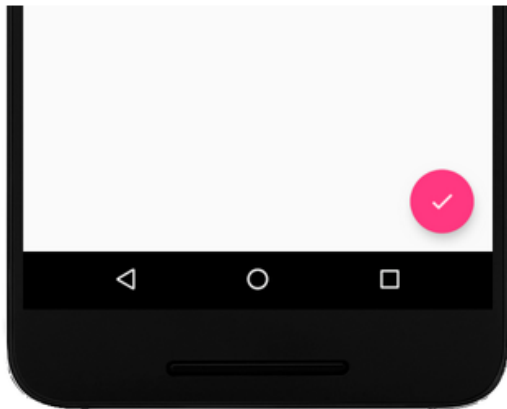


Прочие элементы на сайте:

- Добавление ВЕБ элементов в приложение (работа в темной теме, отображение на разных экранах, использование лучших практик и многое другое);
- Применение тем к вашим элементам (закругление углов, тени и прочее);



- Добавление готовых элементов (плавающая кнопка, чек-боксы и прочее);



- Очень подробные инструкции по анимации и переходам;
- Работа с тачпадом и вводом;
- Работа с текстом и эмоджи;
- Использование системных возможностей;
- Кастомизация запуска приложения;
- Создание виджетов;
- Вкладка с эталонами;
- Вкладка с образцами.

Вкладка дизайн и качество:

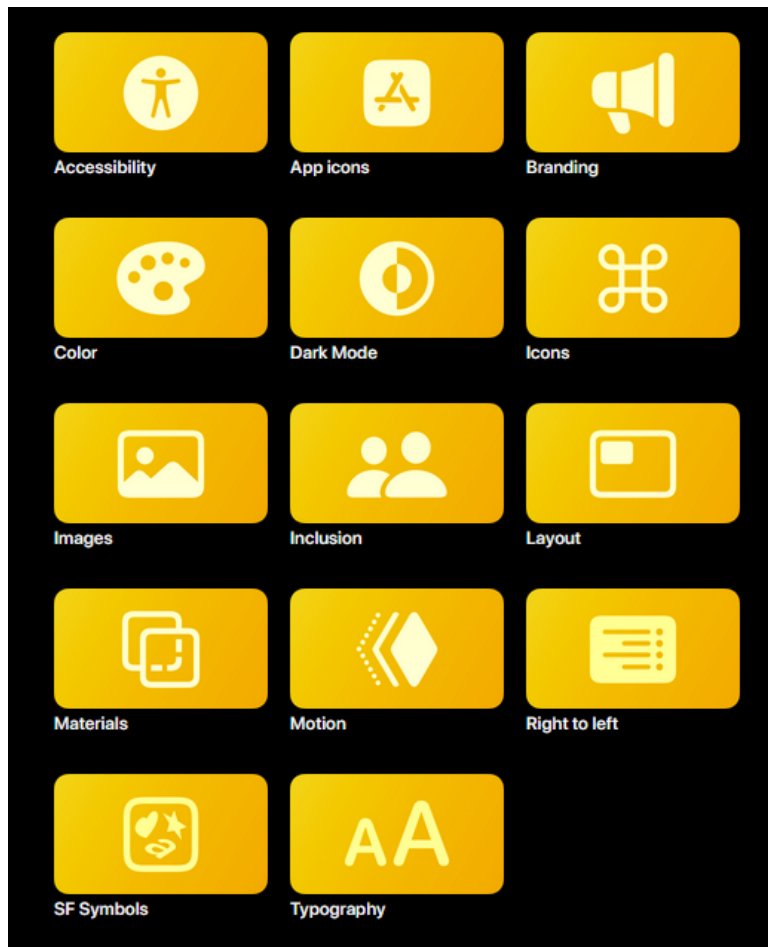
Содержит в себе чек-лист для тестирования приложения, а так же отдельные чек-листы для разного вида устройств:

- Основная проверка приложения;
 - Визуальный опыт;
 - Функциональность;
 - Продвинутый и стабильность;
 - Конфиденциальность и безопасность;
 - Гугл плэй;
 - Настройка тестовой среды;
 - Тестовые процедуры;
 - Тесты в строгом режиме.

- Качество приложения на больших экранах;
- Качество приложения на носимых устройствах (например часы);
- Качество приложения для телевизоров;
- Качество приложения для автомобиля;
- Вкладка build for billions - оптимизация приложения при плохом соединении, нестандартных экранах, для экономии трафика и заряда батареи;
- Ссылка на отдельную страницу с чек-листами для игр.

iOS разработка и требования к приложениям:

iOS разработка быстрее и дешевле в плане фрагментации. Android имеет много версий и много устройств (24000 различных Android-устройств от более чем 1200 производителей). Это усложняет разработку и тестирование приложения, что приводит к затратам времени и денег. Только Apple использует ОС iOS и имеет ограниченное количество смартфонов и планшетов. Материал взят с официального сайта <https://developer.apple.com/design/human-interface-guidelines/foundations/>



Accessibility (Удобство, доступность):

- **Лучшие практики:**

- Дизайн с учётом доступности и удобства. Приоритет на простоту и восприятие каждого дизайнерского решения независимо от того какими возможностями обладает пользователь и как использует устройство;
- Простота. Сделайте сложные задачи простыми;
- Восприимчивость. Весь контент должен быть воспринят, независимо от того, что использует пользователь, зрение слух или осязание;
- Поддержка персонализации;
- Аудиты и тестирования вашего приложения на удобство;
- Взаимодействия. Поддержка технологий как VoiceOver, специальных возможностей и прочее;

- **Жесты;**

- Не используйте жесты, используемые ОС для стандартной работы;

- Простые жесты предпочтительны, для обычных взаимодействий;
- У жестов должен быть альтернативный способ выполнения;
- Если есть возможность, используйте более одного способа физического взаимодействия;
- Добавьте функцию Drag and drop (перетаскивание);
- **Кнопки и элементы управления;**
 - Целевой размер любого элемента управления не должен быть менее 44x44 пикселя;
 - Охарактеризовать доступность пользовательских элементов;
 - Использовать заливку кнопки, которая выполняет наиболее вероятное действие;
 - Предпочтительно использование системных переключателей;
 - Отображение ссылок выделяется синим цветом с подчеркнутым шрифтом;
- **Пользовательский ввод;**
 - Позвольте пользователю наговаривать текст, а не печатать;
 - Организуйте поддержку Сири;
 - Поддержка выделения текста;
 - Поддержка тактильной обратной связи (вибрации);
 - Поддержка VoiceOver (читает содержимое экрана);
 - Описание контента (сделайте описания изображения, для тех, кто не может его видеть и использует VoiceOver);
 - Краткое описание инфографики (для тех, кто не может его видеть и использует VoiceOver);
 - Если изображение декоративное скройте его от вспомогательных технологий;
 - Каждый экран должен быть с уникальным заголовком;
- **Навигация;**
 - Убедитесь, что к каждому элементу можно перейти, используя VoiceOver;

- При переходе на другую страницу, пользователь должен быть уведомлён;
- Создать альтернативные текстовые метки, для всех важных функций;
- **Текст;**
 - Текст должен динамически адаптироваться, ко всем размерам шрифта;
 - Избегайте обрезания части текста, при увеличении шрифта;
 - Настроить макет, для большого размера шрифта;
 - Увеличение размера значимых элементов, по мере увеличения шрифта;
 - Предпочтительно использование стандартного шрифта;
 - Избегайте использования курсива или текста заглавными буквами, при длинных фрагментах текста;
- **Цвета и эффекты;**
 - Не полагаться исключительно на цвет, чтобы выделить важную информацию;
 - Предпочтительно использование системных цветов текста;
 - Настроить цветовую схему для инверсии цветов;
 - Использовать сильно контрастные цвета, для лучшей читаемости;
 - С осторожностью добавляйте быстро движущие, мерцающие объекты;

App icon (Иконки приложений):

- Простота;
- Дизайн иконки должен отображаться одинаково хорошо на разных платформах;
- Использование текста в иконке, предпочтительно только в том случае, если он используется в вашем бренде;
- Оптимизировать значёк, для разных приложений (например поиск, будет отображаться маленький значек в результатах);
- Иконка создается квадратной, без полей (разные устройства автоматически изменяют вид иконки, для правильного отображения);

- Не используйте значки (элементы значков) компании Apple;

App icon attributes

App icons in all platforms use the PNG format and support the following color spaces:

- Display P3 (wide-gamut color)
- sRGB (color)
- Gray Gamma 2.2 (grayscale)

The layers, transparency, and corner radius of an app icon can vary per platform. Specifically:

Platform	Layers	Transparency	Asset shape
iOS, iPadOS	Single	No	Square
macOS	Single	Yes, as appropriate	Square with rounded corners
tvOS	Multiple	No	Rectangle
watchOS	Single	No	Square

App icon sizes

iOS, iPadOS app icon sizes

You need to provide a large version of your app icon, measuring 1024x1024 pixels, to display in the App Store. You can let the system automatically scale down your large app icon to produce all other sizes, or — if you want to customize the appearance of the icon at specific sizes — you can supply multiple versions.

@2x (pixels)	@3x (pixels) iPhone only	Usage
120x120	180x180	Home Screen on iPhone
167x167	—	Home Screen on iPad Pro
152x152	—	Home Screen on iPad, iPad mini
80x80	120x120	Spotlight on iPhone, iPad Pro, iPad, iPad mini
58x58	87x87	Settings on iPhone, iPad Pro, iPad, iPad mini
76x76	114x114	Notifications on iPhone, iPad Pro, iPad, iPad mini

Брендинг:

- Выберите концептный цвет;
- Брендинг должен зависеть от контента;
- Соблюдайте правила товарных знаков Apple;

Цвета:


- Инклюзивность цветов;
 - Не полагаться только на отображение цветом важной информации (использовать метки или глифы, для людей с дальтонизмом);
 - Избегать цветов, которые затрудняют восприятию контента (например серый текст на белом фоне);
 - Обратить внимание, как будут восприниматься сочетания цветов в других странах;
- Системные цвета;
 - Избегать жесткой привязки цвета к объектам, используйте динамический (например при смене на темную тему);
 - Избегать жесткого копирования системных цветов;
- Управление цветами;
 - Применение всех цветовых профилей к элементам;

- Использовать широкий диапазон цвета;

System colors (iOS)

Default	Accessible		Name
Light	Dark		
 R 205 G 28 B 28	 R 205 G 59 B 59		Red
 R 255 G 140 B 50	 R 255 G 100 B 50		Orange
 R 255 G 255 B 50	 R 255 G 211 B 50		Yellow
 R 142 G 189 B 59	 R 142 G 189 B 59		Green
 R 199 G 199 B 199	 R 199 G 199 B 199		Mint
 R 142 G 170 B 199	 R 142 G 170 B 199		Teal
 R 99 G 173 B 199	 R 99 G 173 B 199		Cyan
 R 99 G 122 B 199	 R 99 G 122 B 199		Blue
 R 99 G 85 B 174	 R 99 G 85 B 174		Indigo
 R 179 G 122 B 174	 R 179 G 122 B 174		Purple
 R 255 G 140 B 140	 R 255 G 140 B 140		Pink
 R 142 G 122 B 99	 R 142 G 122 B 99		Brown

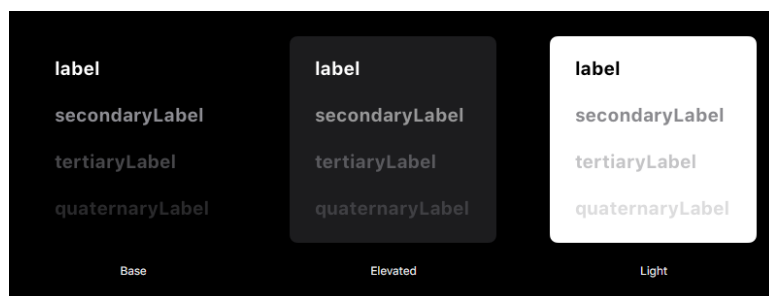
System gray colors (iOS)

Default	Accessible		Name
Light	Dark		
 R 142 G 142 B 147	 R 142 G 142 B 147		Gray
 R 174 G 174 B 178	 R 99 G 99 B 102		Gray (2)
 R 199 G 199 B 204	 R 72 G 72 B 74		Gray (3)
 R 209 G 209 B 214	 R 58 G 58 B 60		Gray (4)
 R 229 G 229 B 234	 R 44 G 44 B 46		Gray (5)
 R 242 G 242 B 247	 R 28 G 28 B 30		Gray (6)

Dark mode (Темный режим):

- Цвета в темном режиме;
 - Более тусклые цвета фона;
 - Более яркие цвета переднего плана;

- Не все цвета инвертируются;
- Цвета должны быть более контрастны;
- Светлый задний фон должен быть смягчен;
- Иконки и изображения;
 - При необходимости создать значки для темной темы;
- Текст;
 - Использовать системные метки;



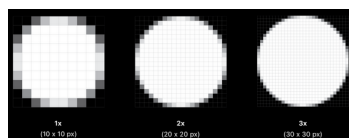
- Предпочтительно использование системных фонов;

Иконки:

- Предоставить текстовые метки для VoiceOver;
- Избегать дублирования значков Apple;
- При создании значка используйте формат PDF или SVG;
- Избегать текста в значках;

Изображения:

- Скалирование



Platform	Scale factors
iOS	@2x and @3x
iPadOS	@2x
macOS, tvOS	@1x and @2x
watchOS	@2x

- Для всех изображений должна быть версия в высоком разрешении;

- Добавлять цветовой профиль к каждому изображению;
- Всегда тестировать изображение на ряде реальных устройств;

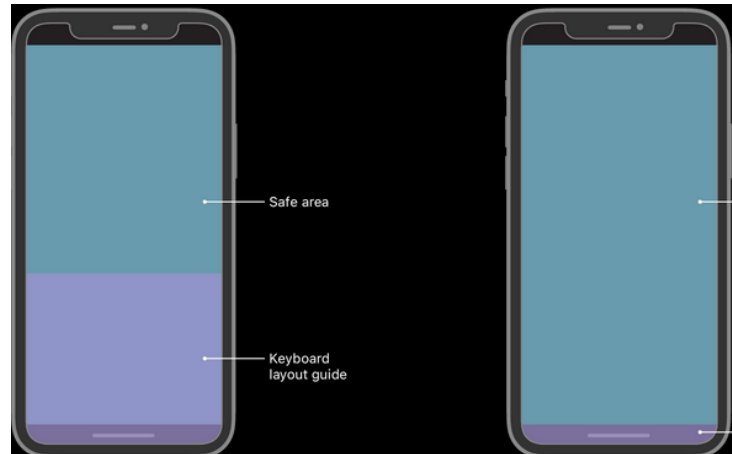
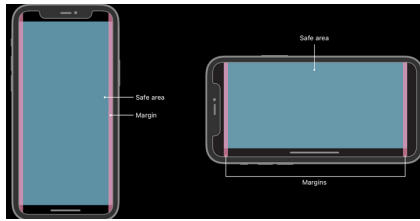
ИНКЛЮЗИВНОСТЬ:

Я просто оставлю это здесь =)

- Age
- Gender and gender identity
- Race and ethnicity
- Sexuality
- Physical attributes
- Cognitive attributes
- Permanent, temporary, and situational disabilities
- Language and culture
- Religion
- Education
- Political or philosophical opinions
- Social and economic context

Layout (Схема расположения):

- Поддержка книжной и альбомной ориентации;
- Приложение должно работать при разных разрешениях экрана (разные модели iPhone);
- Соблюдать поля, по краям полноразмерных кнопок;
- Фон должен покрывать весь задний фон;
- Избегать добавления интерактивных элементов в нижней части экрана;
- Убирать строку состояния, только при необходимости;



Материалы:

На сайте присутствуют шаблоны разных элементов, для упрощения разработки;

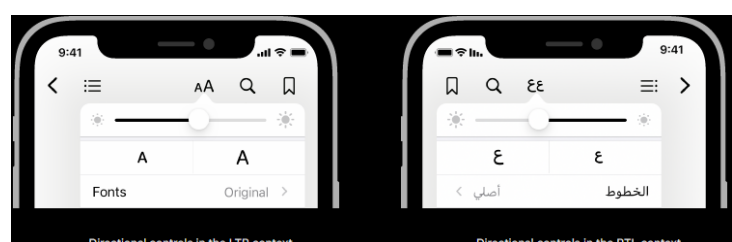
- `label` (default)
- `secondaryLabel`
- `tertiaryLabel`
- `quaternaryLabel`

You can use the following vibrancy values for fills on all materials.

- `fill` (default)
- `secondaryFill`
- `tertiaryFill`

Right to left (Право на лево):

В арабских странах читают и пишут справа на лево, должна быть возможность выравнивания текста по правому краю и перенос элементов управления соответственно;



Некоторые элементы могут отображаться некорректно или заезжать друг на друга из-за специфики арабского письма;



ВИНДОУС МОБАЙЛ? СЕРЬЁЗНО? ОНА В 2013 - 2017 ПРЕКРАТИЛА СВОЁ СУЩЕСТВОВАНИЕ

Согласно отчету аналитической компании International Data Corporation, в 2013 году Windows Phone вошел в тройку лидеров по выпуску операционных систем для смартфонов и был назван платформой с наиболее быстрыми показателями роста доли рынка (на момент публикации прирост составлял 91% в год). Windows Phone является третьей по распространенности операционной системой на 9 рынках [\[источник 7\]](#)
По данным аналитической компании IDC доля рынка устройств на базе Windows Phone, пройдя свой максимум в 2013 со значением 3.4%, снизилась до 2.5% во втором квартале 2014 года. В дальнейшем доля Windows Phone только снижается, достигнув значения 1.6% к концу 2015. К середине 2016 года доля Windows Phone сократилась до исторического минимума в 0.7% рынка [\[источник 8\]](#)

Нефункциональное тестирование мобильных приложений:

Тестирование прерываний (всплывающие подсказки, сворачивание/разворачивание приложения, разрядка/подзарядка, входящие/исходящие вызовы/SMS/MMS):

Прерывание - это смена фокуса с одного приложения на другое, которое требует немедленной реакции.

Пример: оформляешь перевод в мобильном банке, появляется поп-ап входящего звонка. Поп-ап вызова создает прерывание.

После завершения прерывания прерванный сценарий должен быть продолжен с момента остановки.

Почему важно тестировать прерывания

Прерывания часто становятся причиной того, что пользователь не может завершить сценарий прерванный другим приложением. И часто это причина blockers.

Случай из жизни: пользователь набрал корзину, закрыл приложение, вновь открыл его - корзина пустая. Это блокиер!

Типы прерывания

Это не исчерпывающий список, но можно использовать как чек-лист.

Аппаратные

- Разрядка аккумулятора
- Низкий заряд аккумулятора
- Перезагрузка
- Отключение
- Блокировка-разблокировка
- Горячая установка сим/флеш-карты
- Подключение/отключение зарядки
- Нехватка памяти

Сетевые

- Потеря/восстановление сети
- Смена типа сети
- Входящий, исходящий звонок
- Удаленное управление*Например, с помощью Apple Watch можно запустить таймер фото, соответственно фокус переключается на приложение "Камера"*
- Передача и получение информации с помощью периферийных устройств*Например, во время работы с приложением на iOS по AirDrop прилетает запрос принять фотографию*

Системные

- Пуши от других приложений, системные (будильник, таймер, напоминание)
- Диалоговые окна, предложение обновить ОС
- Смена фокуса на другое приложение
- Смена ориентации экрана

- Вызов нативных приложений. В приложении тапаешь на ссылку - открывается встроенный браузер внутри приложения. Или кликаешь написать на эл.почту - открывается нативное приложение "почта".

Обновление

Например, открыли приложение интернет-ТВ, перешли в раздел "ТВ", выбрали канал "ТестированиеТВ" и по тапу на воспроизведение получили сообщение, что нужно обновить приложение. Правильный сценарий завершения прерывания на обновление - это вернуть в приложение в раздел "ТВ" и включить канал "ТестированиеТВ".

Лонгтапы и жесты

Это на грани тестирования логики приложения. Но все же пример: открыли банковское приложение, открыли форму пополнения баланса телефона, свернули приложение, лонгтапнули по иконке банка и выбрали пункт "Курс валют". Один из правильных сценариев работы с таким прерыванием - это после закрытия экрана валют открывать экран пополнения баланса.

Тестирование безопасности и конфиденциальности:

Тестирование безопасности означает верификацию безопасности системы и анализ рисков, связанный с целостным подходом к защите приложения, хакерским атакам, вирусам, несанкционированным доступом к конфиденциальной информации и т.д.

Главной целью тестирования безопасности является гарантия безопасности сети и информации. Тестирование безопасности часто связано с web-продуктами, так как многие люди могут получить доступ к web-приложению через Интернет.

К тестированию безопасности относится:

- **Конфиденциальность** - пользователи должны быть уверены, что их конфиденциальные данные эффективно защищены от третьих лиц и людей со злыми намерениями;
- **Целостность** - пользовательские данные не должны быть потеряны или повреждены;

- **Аутентификация** - пользовательские данные должны быть проверены на подлинность; в противном случае меры безопасности неприменимы;
- **Авторизация** - надлежащая защита данных может быть гарантирована только в том случае, если каждый пользователь идентифицирован и предоставил доступ к определенным данным;
- **Доступность** - пользователи должны легко проходить процедуру авторизации и получать доступ к своим данным;
- **Неопровержимость** - система должна обеспечивать отслеживание тех, кто отправляет сообщения и получает их. Пользователи не должны отказывать в отправке или получении сообщений, если они это сделали.

Распространённые угрозы безопасности ПО:

- Межсайтовый скриптинг - предполагает, что вредоносные скрипты запускаются на web-странице, сгенерированной сервером. С помощью сценариев злоумышленники могут направить пользователя на другой сайт, где может произойти серьезная атака, украсть файлы cookie пользователя, загрузить вредоносную программу на компьютере пользователя и т. д.;
- Подделка на межсайтовый запрос - недобросовестные люди размещают ссылку на свой сайт на надежном сайте или на web-странице. Когда пользователь переходит на сайт злоумышленника, вредоносный код запускает и крадет конфиденциальную информацию пользователя (логины и пароли или получает контроль над учетной записью пользователя и т. д.);
- Кодовые инъекции - злоумышленники добавляют код в программу, которая меняет его функционирование. Таким образом, они могут получить доступ к конфиденциальной информации пользователей, разбить приложение и так далее;
- Серверная сторона включает инъекцию - недобросовестные люди вводят вредоносные скрипты в код HTML или запускают свои скрипты с сервера;
- Автономный обход - позволяет злоумышленникам получить неавторизованный доступ к учетной записи или личным данным другого пользователя.

Ключевые действия для проверки безопасности мобильного приложения:

- Убедиться, что данные
- Проанализировать требования по

пользователей приложения (логины, пароли, номера кредитных карт) защищены от атак автоматических систем и не могут быть обнаружены путем отбора;

- Проверить, не предоставляет ли приложение доступ к секретному контенту или функционалу без должной аутентификации;
- Убедиться, что система безопасности приложения требует надежный пароль и не позволяет хакеру использовать пароли других пользователей;
- Проверить, является ли время проведения сессии приложения соответствующим;
- Найти динамические зависимости и принять меры для защиты этих уязвимостей от хакеров;
- Защитить приложение от атак SQL-инъекций;
- Найти случаи неуправляемого кода и устранить последствия;
- Убедиться, что срок действия сертификата не истек, если приложение использует Certificate Pinning;
- Защитить приложение и сеть от DoS-атак;
- Провести анализ разных потоков данных и обезопасить системы от

хранению и проверке данных;

- Обеспечить управление сессий для защиты информации от несанкционированных пользователей;
- Изучить все криптографические коды и исправить ошибки, если необходимо;
- Убедиться, что бизнес-логика приложения защищена и не подвержена воздействию внешних атак;
- Проанализировать взаимодействие системных файлов, выявить и исправить уязвимости;
- Проверить блоки управления протоколами (к примеру, не подвергается ли страница сбросу по умолчанию с помощью вредоносных iFrames);
- Защитить приложение от вредоносных атак;
- Обезопасить систему от вредоносных вторжений во время работы программы;
- Предотвратить небезопасное хранение данных в памяти устройства;
- Исключить возможные вредные действия куки;
- Обеспечить регулярный контроль безопасности информации;

- потенциального вредоносного влияния;
- Освободить систему от случаев переполнения буфера или от нарушения целостности памяти;
- Изучить пользовательские файлы и предотвратить их возможное вредоносное влияние.

Тестирование зависимости от сетей и каналов связи (тип соединения 3G, 4G, 5G, Wi-Fi)

Подключение через Wifi:

Данный тип подключения является достаточно быстрым. На нем обычно тестирую приложения в нормальных условиях. Тут также можно протестировать приложение при потере связи во время передачи данных.

Рассмотрим какие кейсы следует проверить при Wifi подключении:

Поведение приложения при отключении сети Wifi во время передачи данных. Приложение делает долгую сетевую операцию, мы отключаем интернет во время данной операции. Приложение должно выдать ошибку о отсутствии подключения к интернет, отменить или приостановить операцию в зависимости от алгоритма. После включения Wifi все функции приложения должны работать.

Поведение приложения при отсутствии подключения к интернет.

Отключаем Wifi и пытаемся выполнить сетевые операции. При выполнении сетевой операции приложение должно выдавать сообщение об отсутствии интернета.

Поведение приложения, при наличии Wifi сети, не подключенной к интернет. Или когда чтобы выйти в интернет, нужно ввести пароль. Такие сети бывают в аэропортах. Проэмулировать сеть, не подключенную к интернет просто. Берем роутер, к которому подключаются устройства по Wifi, и отключаем у роутера интернет. У телефона остается подключение по Wifi, но выйти в интернет нельзя. При таком типе подключения при выполнении сетевых операций приложения должно выдавать сообщение о отсутствии интернет Источник: https://mobile-testing.ru/different_connection_inernet_types/

Подключение через 3g(Edge), 4G, 5G

Данный тип подключения является достаточно медленным. Сотовая сеть может появляться и исчезать в поездке, например. Здесь нужно обратить внимание на следующие кейсы.

Пропадание сотовой сети во время передачи данных мобильным приложением. Пропадание можно эмулировать, положив телефон с металлическую коробку, например. Приложение должно правильно реагировать на пропадание сигнала и выдавать соотв. сообщение о отсутствии интернет подключения.

Длительные запросы по сети. Как известно Edge подключение — медленное подключение. Если ваше мобильное приложение использует Арі, то запросы также будут медленными. Здесь возможен такой кейс: Нажимаем на кнопку, к примеру отправить — идет запрос. Далее пока идет запрос, снова нажимаем на ту же кнопку. Или делаем быстро несколько параллельных запросов. Обычно в таком случае приложение должно показывать активити бар. Иначе при отправке второго запроса, когда еще не прошел первый приложение начинает некорректно работать и крешиться.

Замеры сетевого трафика. Как известно, интернет трафик для многих типов трафика является платным. Для замеров сетевого трафика обычно можно пользоваться подключением типа.

Тестирование инсталляции: установка\удаление\переустановка\обновление программного обеспечения.

Под инсталляционным тестированием (тестированием установки) подразумевают уровень корректности установки некоего программного продукта в искусственно созданной среде с целью выявления степени ее готовности к эксплуатации.

Тестирование инсталляции (установки) направленно на проверку успешной инсталляции и настройки, а также обновления или удаления программного обеспечения.

Данный вид тестирования используется для проверки установки приложений и программ, как десктопных, так и мобильных.

Тестирование инсталляции в большинстве своем не входит в Веб-тестирование, являясь специализированным тестированием установки приложений на различные операционные системы.

Тестирование установки

Тестирование установки подтверждает, что процесс установки проходит без каких-либо проблем.

Пример тестовых сценариев –

- Убедитесь, что процесс установки проходит гладко и не займет много времени.
- Убедитесь, что установка прошла успешно через корпоративный магазин приложений.

Тестирование удаления

Основы тестирования деинсталляции могут быть сведены в одну строку: «Деинсталляция должна охватывать данные, связанные с приложением, всего за один раз».

Пример тестовых сценариев –

- Убедитесь, что все файлы, связанные с приложением, успешно удалены после удаления.
- Если это приложение, в котором хранятся мультимедийные файлы (например, Whatsapp или Facebook), сохраните файлы даже после удаления приложения.

Убедитесь, что все файлы, связанные с приложением, успешно удалены после удаления.

Если это приложение, в котором хранятся мультимедийные файлы (например, Whatsapp или Facebook), сохраните файлы даже после удаления приложения.

Тестирование обновлений

Мы должны быть очень осторожны с обновлениями мобильных приложений. Люди часто жалуются на то, что приложения не работают удовлетворительно после обновления. Поэтому очень важно, чтобы при тестировании обновлений мы определяли, что приложение будет работать так, как оно работало ранее. В двух словах, это ничего не должно сломать. Обновления мобильных приложений могут осуществляться двумя способами – **автоматическим обновлением** и **ручным обновлением**.

Пример тестовых сценариев –

- Убедитесь, что приложение успешно работает после автоматического обновления.
- Убедитесь, что процесс обновления отображается правильно.

Тестирование совместимости с функциями телефона

Тестирование совместимости используется, чтобы убедиться, что ваше приложение совместимо с функциями телефона:

Примеры:

- Жесты/смахивания
- Расширение/ориентация/размер экрана
- Видео/фото
- GPS
- Отображение на заблокированном экране
- Добавление в календарь/wallet/локальные папки

Тестирование производительности

Мобильный тест производительности охватывает производительность клиентских приложений, производительность сервера и производительность сети. Важно убедиться, что сценарии тестирования производительности охватывают все эти области. С помощью инструментов тестирования производительности нетрудно идентифицировать существующие сети, серверы и узкие места серверных приложений, учитывая предопределенную нагрузку и сочетание транзакций.

Пример тестовых сценариев –

- Убедитесь, что проверка доступного рейса занимает только разумное количество времени.
- Убедитесь, что во время проверки доступности рейса мобильный телефон работает нормально и не зависает.

Тестирование локализации

В настоящее время большинство приложений предназначено для глобального использования, и очень важно заботиться о региональных средах, таких как языки, часовые пояса и т. д. Важно проверить функциональность приложения,

когда кто-то меняет часовой пояс. Необходимо учитывать, что иногда западные дизайны могут не работать с аудиторией из восточных стран или наоборот.

Пример тестовых сценариев –

- Убедитесь в отсутствии проблем с пользовательским интерфейсом или усечением данных, когда мы используем мобильное приложение на разных языках (или, скажем, не на английском языке).
- Убедитесь, что изменения часового пояса корректно обрабатываются для вашего мобильного приложения.

Тестирование конфигурации

Это специальный вид тестирования, направленный на проверку работы программного обеспечения при различных конфигурациях системы, при различных версиях системы.

Пример: протестировать на разных версиях ОС