

# Mobile Testing

## Типы приложений

Мобильные приложения делятся на 3 типа:

1. **Нативное приложение** - 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.
2. **Веб-приложение** - открывается через браузер, а значит это просто веб-сайт.
3. **Гибридное приложение** - устанавливается через маркетплейс, а отображается внутри приложения как веб-сайт. Часто это приложения супермаркетов, недорогих доставок еды.

## Немного о плюсах и минусах типов для тестирования

Говорим о плюсах и минусах именно для тестирования. Если говорить в общем, то список будет намного длиннее. Например, нативное приложение очень дорогое в разработке и тестировании, что является минусом для бизнеса.

### Плюсы нативного приложения:

- 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.)

### Минусы нативного приложения:

- правки багов доезжают только при релизе следующей версии
- тестирование на каждой платформе
- занимают больше памяти

### Плюсы веб-приложений:

- правки багов приезжают быстрее
- тестирование проводится в браузере и не сильно завязано на ОС/модель телефона/платформу

- не требуется тестировать установку, удаление и обновление

#### **Минусы веб-приложений:**

- ограниченный доступ к функциональности девайса (FaceID, отпечаток пальца, камера и т.п.)
- не работают в оффлайне

#### **Плюсы гибридного приложения:**

- мультиплатформенные, т.е. реализация на всех ОС одна
- правки багов приезжаю быстрее, т.к. по сути функционал это встроенный веб-сайт в приложение
- могут использовать большинство функций девайса

#### **Минусы гибридного приложения:**

- не работают в офлайне
- ограниченный доступ к функциональности девайса

## **Material Design for Android**

**Material Design** – это дизайн-система, которую изобрели и представили в компании Google в 2014 году. Это не просто гайдлайн по единому визуальному оформлению – благодаря ему были унифицированы интерфейсы всех продуктов и сервисов корпорации, в частности ОС Android.

## **Основные принципы Material Design**

**В основе концепции Material Design лежат 4 принципа:**

1. **Тактильные поверхности.** Основой формирования пространства служит «цифровая бумага». Отдельные ее листы располагаются на разной высоте и при этом отбрасывают тени друг на друга. Помимо прочего, они также могут растягиваться, менять форму и цвет, также соединяться друг с другом. За счет этого пользователи лучше понимают, как устроена система и какова ее иерархия.
2. **Полиграфический дизайн.** Это означает, что при создании интерфейса для цифровых устройств используются традиционные средства и подходы из графического дизайна. Скажем так, на «цифровой бумаге» элементы выводятся с помощью «цифровых чернил».

3. **Осмысленная анимация.** Здесь анимация не просто выскакивает из ниоткуда, а появляется в соответствии с логикой системы. То есть один объект, реагируя на действия пользователя, плавно переходит в другой.
4. **Адаптивный дизайн.** Тут все просто – интерфейс должен быть оптимизирован на всех устройствах и экранах вне зависимости от того, какой продукт мы берем в качестве примера. При этом выглядеть, работать и реагировать все должно везде одинаково.

## Human Interface Guidelines

Приступая к разработке приложения или игры для iOS, начните с понимания следующих основных характеристик и шаблонов устройств, которые отличают интерфейс iOS. Использование этих характеристик и шаблонов для обоснования ваших дизайнерских решений может помочь вам создать приложение или игру, которые оценят пользователи iPhone.

- **Дисплей.** iPhone имеет дисплей среднего размера с высоким разрешением.
- **Эргономика.** Люди обычно держат свой iPhone в одной или обеих руках, когда они взаимодействуют с ним, переключаясь между альбомной и портретной ориентацией по мере необходимости. Пока люди взаимодействуют с устройством, расстояние до них обычно не превышает одного-двух футов.
- **Входные данные.** Мультитач-жесты, экранные клавиатуры и голосовое управление позволяют людям выполнять действия и выполнять значимые задачи на ходу. Кроме того, люди часто хотят, чтобы приложения использовали их местоположение и ввод данных с помощью акселерометра и гироскопа устройства, а также могут захотеть участвовать в пространственных взаимодействиях.
- **Взаимодействие с приложениями.** Иногда люди тратят всего минуту или две на проверку обновлений событий или социальных сетей, отслеживание данных или отправку сообщений. В других случаях люди могут потратить час или больше на просмотр веб-страниц, игры или просмотр мультимедиа. Обычно у людей одновременно открыто несколько приложений, и им нравится часто переключаться между ними.
- **Особенности системы.** iOS предоставляет несколько функций, которые помогают людям взаимодействовать с системой и их приложениями привычными и согласованными способами.

**Отличительные возможности iPhone** объединяют платформу и возможности устройства, которые люди ценят больше всего. Чтобы ваш дизайн чувствовал себя в iOS как дома, расставьте приоритеты в следующих способах включения этих функций и возможностей.

- Помогите людям сосредоточиться на основных задачах и содержимом, ограничив количество экранных элементов управления, одновременно делая второстепенные детали и действия доступными для обнаружения при минимальном взаимодействии.
- Легко адаптироваться к изменениям внешнего вида, таким как ориентация устройства, темный режим и динамический тип, позволяя пользователям выбирать наиболее подходящие для них конфигурации.
- Включите взаимодействия, которые поддерживают то, как люди обычно держат свое устройство. Например, людям, как правило, проще и удобнее обращаться к элементу управления, когда он расположен в средней или нижней части дисплея, поэтому особенно важно, чтобы пользователи могли перемещаться назад или выполнять действия в строке списка.
- С разрешения пользователей интегрируйте информацию, доступную через возможности платформы, таким образом, чтобы улучшить взаимодействие, не требуя от людей ввода данных. Например, вы можете принимать платежи, обеспечивать безопасность с помощью биометрической аутентификации или предлагать функции, использующие местоположение устройства.

## Guidelines для Windows Phone

Это операционная система с очень низкой популярностью, не то что Windows для компьютера, но все равно обладает собственной документацией с рекомендациями к разработке приложений.

Гайдлайны от Windows невозможно сравнить с гайдами от Google и Apple. «Топорный вид» создания этой документации — первое, что бросается в глаза. Эти гайды не так подробно расписаны, как предыдущие два, однако в них описано все, что нужно знать и понимать для разработки приложений на «телефонную» Windows.

Главный плюс гайдлайна — русский язык. Информация изложена подробно, но недостаточно удобно структурирована, многовато «воды» для подобного руководства. Однако, если рассматривать гайдлайн не как учебник, который надо сначала прочитать, а потом приступить к разработке, а как карманный справочник — тогда можно признать, что подобное излишество полезно.

# Нефункциональное тестирование мобильных приложений

Нефункциональное тестирование проводится для проверки нефункциональных требований приложения, таких как производительность, безопасность, совместимость, надежность, удобство использования и т. д. В большинстве случаев это выполняется методом black box testing. Оно проверяет, соответствует ли поведение системы требованиям по всем аспектам, не охваченные функциональным тестированием.

## На что обратить внимание при нефункциональном тестировании мобильных приложений:

- Входящие и исходящие SMS, звонки, оповещения других приложений;
- Перегревание батареи;
- Отключение и подключение провода;
- Отключение и включение сети, Bluetooth, авиарежима, GPS;
- Отключение и подключение SD-карты, дополнительных устройств вроде физической клавиатуры или гарнитуры;
- Потеря связи с сервером или прокси.

Также важным этапом в тестировании, направленном на поддержку и корректное отображение программного продукта является **кросс-браузерность**. В конечном итоге продукт должен правильно отображаться на любом современном устройстве.

Проверка кросс-браузерной совместимости предполагает не только тесты на реальных браузерах, но и использование удобных онлайн-сервисов и утилит. Таких как BrowserStack, Browsershots, Webshot by Mobile Ready и другие.

## Виды нефункционального тестирования:

### Тестирование производительности

Проверяет, как программный продукт работает под определенной нагрузкой.

Задачей тестирования производительности является определение масштабируемости приложения под нагрузкой, при этом происходит:

- измерение времени выполнения выбранных операций при определенных интенсивностях выполнения этих операций,
- определение количества пользователей, одновременно работающих с приложением,
- определение границ приемлемой производительности при увеличении нагрузки (при увеличении интенсивности выполнения этих операций),
- исследование производительности на высоких, предельных, стрессовых нагрузках.

## **Тестирование производительности делится на:**

### **1. Нагрузочное тестирование (Performance and Load Testing)**

Иногда одновременная работа нескольких приложений не дает возможности открыть новое или загрузить корректно уже открытое. Приложение должно учитывать эти ситуации.

#### **Что проверяем?**

- Работа приложения в условиях нехватки памяти;
- Поведение приложения при открытии в условиях нехватки памяти;
- Поведение приложения в условиях нехватки свободного места для установки;
- Поведение приложения в условиях отсутствия какой-либо функции на устройстве (3G, SD-карта и т. п.);
- Установка, перенос и удаление приложения (установка или перенос приложения на карту SD).

### **2. Стрессовое тестирование (Stress Testing)**

Определяет работоспособность программного продукта при максимальной нагрузке.

#### **Что проверяем?**

1. Высокая загрузка центрального процессора
2. Нехватка памяти
3. Загрузка батареи
4. Отказы

5. Низкая пропускная способность сети

6. Большое количество взаимодействий пользователя с приложением (для этого может понадобиться имитация реальных условий состояния сети)

Важно не путать стрессовое и нагрузочное тестирование. Нагрузочное тестирование — это все же тестирование средних нагрузок. А стрессовое — это тестирование запредельных нагрузок

### 3. Тестирование стабильности/надежности (Stability / Reliability Testing)

Проверяет, выдержит ли программный продукт длительную нагрузку.

Время на выполнения операций может играть в данном виде тестирования второстепенную роль. При этом на первое место выходит отсутствие утечек памяти, перезапусков серверов под нагрузкой и другие аспекты, влияющие именно на стабильность работы.

### 4. Объемное тестирование (Volume Testing)

Для получения оценки производительности при увеличении объёмов данных в базе данных приложения. При этом происходит:

- измерение времени выполнения выбранных операций при определенных интенсивностях выполнения этих операций,
- может производиться определение количества пользователей, одновременно работающих с приложением.

Итак, закрепим. Какую же работоспособность проверяем в каждом из видов тестирования производительности? Нагрузочное тестирование — при **нормальных** условиях. Стрессовое тестирование — при **экстремальных** нагрузках. Тестирование стабильности — при **длительной** работе. Объемное тестирование — при **увеличенных объемах** обрабатываемых данных.

### 5. Тестирование подключения

Так же к тестированию производительности можно отнести **тестирование подключения**

Проверка при нестабильном сигнале Wi-Fi или с нулевым балансом на счету в сети 3G. Необходимо тестировать приложение как минимум 4-мя способами:

- Позитивный кейс (наличие отличной постоянной связи);
- Наличие постоянной неотличной связи;
- Отсутствие связи;
- Потеря связи.

### **Что проверяем?**

- Тестирование при подключенном Wi-Fi, 4G/3G/E/etc
- Разрыв и восстановление сети
- Переключение с одного типа сети к другому
- Оффлайн
- Смена, отключение геопозиции

## **Инсталляционное тестирование/ Тестирование установки (Installation Testing)**

Проверяет, не возникает ли проблем при установке, удалении, а также обновлении программного продукта. Они должны быть простыми и быстрыми для пользователя. Не должны требовать дополнительных знаний. После обновления, приложение не должно терять накопленную о пользователе информацию. Обновления должны устанавливаться при помощи любого соединения (Wi-Fi, 3G, установка с ПК, на SD).

**Поэтому тестирование установки можно назвать одной из важнейших задач по обеспечению качества программного обеспечения.**

### **Что проверяем?**

- Соответствие функционалов новой и старой версии;
- Адекватность обновления (все ли данные сохраняются).



# Тестирование удобства использования/Юзабилити тестирование (usability testing)

Это метод тестирования, направленный на установление степени удобства использования, «обучаемости», понятности и привлекательности для пользователей разрабатываемого продукта в контексте заданных условий.

По сути, проверяет, удобен ли программный продукт в использовании.

## Что проверяем?

- Свайпы, лонгтапы, двойные тапы, тап и т.п. (**важно:** проверяй не на самых быстрых девайсах)
- Элементы должны быть кликабельными. Не должно быть ложных срабатываний по тапу на соседний элемент
- Многократные нажати, скроллы в самый низ/верх списка
- Вызов функций по лонгтапу по иконке на рабочем столе
- Взаимодействие с виджетами на рабочем столе
- Проверка жизненного цикла приложения (об этом поговорим в отдельной статье). Например редактируем анкету пользователя в тестируемом приложении, поступает звонок или открываем сообщение по тапу на нотификацию, возвращаемся в тестируемое приложение - все на прежнем месте и работает.
- Не должно быть пустых экранов
- Любое действие должно приводить к какому-то результату и пользователь должен понимать что он получил
- За действием пользователя следует реакция: вибрация, смена цвета ссылок, звук и т.п.
- Если на действие несколько реакций, то они должны быть синхроны
- Соответствие нативным гайдам ОС. Проще говоря у каждой ОС есть свои требования к используемым элементам управления. Ссылки на гайдлайн [Apple](#) и [Android](#).

Стоит обратить внимание, что данный метод не имеет ничего общего с тестированием пользовательского интерфейса.

## Тестирование пользовательского интерфейса (Graphical user interface, GUI)

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.

### Что проверяем?

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

## Конфигурационное тестирование (Configuration Testing)

Определяет, как программный продукт будет работать в условиях смены конфигурации. Приложение должно правильно отображаться на всех типах

устройств, для которых разрабатывалось.

Можно выделить 2 **цели** конфигурационного тестирования:

- Определить **оптимальную конфигурацию** оборудования, обеспечивающую требуемые характеристики производительности и времени реакции тестируемой системы.
- Проверить объект тестирования на **совместимость** с объявленным в спецификации оборудованием, операционными системами и программными продуктами третьих фирм.

### Что проверяем?

- Работа на retina-экранах элементов интерфейса.
- Переход приложения от горизонтального к вертикальному положению.
- Функционал проверки нужной версии операционной системы. Приложение не должно устанавливаться на неподдерживаемые устройства. Обязательна проверка на всех доступных из поддерживаемых девайсов.
- Адекватная работа всех медиа файлов на разных устройствах и операционных системах.

## Тестирование совместимости (Compatibility testing)

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

### Что проверяем?

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

## Тестирование на внештатные ситуации

Исследование программной системы на предмет восстановления после ошибок, сбоев. Оценивание реакции защитных свойств приложения.

Приложение должно адекватно реагировать на получение хаотичной информации: попадание в карман без блокировки экрана, нажатие на несколько пунктов одновременно и т.д.

### Что проверяем?

- Удобство навигации по приложению.
- Отсутствие пустых экранов в приложении – пользователь не должен оказываться в ситуации, в которой не очевидно, что сейчас происходит и что делать.
- Многократное быстрое нажатие на кнопку – часто при этом может случиться падение приложения. Одновременное нажатие на все клавиши.
- Нативные жесты, предусмотренные функционалом.

При моделировании ситуации сбоя, оценивается как степень потери данных (находится ли она в пределах допустимого), так и способность системы журналировать все транзакции и их статус выполнения.

## Тестирование защищенности/безопасности (Security testing)

Определяет, насколько безопасно использование программного продукта, т.е. защищен ли программный продукт от атак хакеров, несанкционированного доступа к данным и т.д.

Общая стратегия безопасности основывается на трех основных **принципах**:

### 1. Конфиденциальность

Конфиденциальность — это сокрытие определенных ресурсов или информации. Под конфиденциальностью можно понимать ограничение доступа к ресурсу некоторой категории пользователей, или другими словами, при каких условиях пользователь авторизован получить доступ к данному ресурсу.

### 2. Целостность

Существует два основных критерия при определении понятия целостности:

1. **Доверие.** Ожидается, что ресурс будет изменен только соответствующим способом определенной группой пользователей.
2. **Повреждение и восстановление.** В случае когда данные повреждаются или неправильно меняются авторизованным или не авторизованным пользователем, вы должны определить на сколько важной является процедура восстановления данных.

### 3. Доступность

Доступность представляет собой требования о том, что ресурсы должны быть доступны авторизованному пользователю, внутреннему объекту или устройству. Как правило, чем более критичен ресурс тем выше уровень доступности должен быть.

#### Что проверяем?

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

## Тестирование локализации и глобализации (Localization testing)

Тестирование локализованной версии программного продукта. Локализация — это процесс адаптации интерфейса ПО под разные регионы, культуры, языки.

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.

#### Что проверяем?

1. Все элементы в приложении переведены на соответствующий язык
2. Тексты защиты внутри приложения и пользователь в настройках приложения может выставить необходимый язык
3. Тексты зависят от языка в системных настройках

4. Тексты приходят с сервера
5. Корректное отображение форматов дат (ГОД — МЕСЯЦ — ДЕНЬ или ДЕНЬ — МЕСЯЦ — ГОД.)
6. Корректное отображение времени в зависимости от часового пояса

## Интеграционное тестирование (integration testing)

Приложение должно адекватно реагировать на прерывания работы.

### Что проверяем?

- Входящие и исходящие SMS, MMS, звонки, оповещения, работу других приложений.
- Выключение устройства, изъятие аккумулятора, разрядка устройства.
- Переход в режим ожидания (в том числе и с защитой паролем). Смена ориентации устройства в режиме ожидания.
- Отключение и подключение провода.
- Отключение и включение сети, Bluetooth, авиарежима, GPS.
- Потеря связи с сервером или прокси (подключение есть, но пакеты не доходят).
- Отключение и подключение SD-карты, дополнительных устройств вроде физической клавиатуры или гарнитуры.
- Зарядка устройства.
- Работа с акселерометром.
- Работа с физической клавиатурой (если в списке поддерживаемых моделей есть такие).
- Закрывать приложение для того, чтобы открыть какое-то другое на некоторое время и вернуться к вашему приложению позже;
- Послать девайс в сон на некоторое время.

**Все вышеописанные тесты необходимы для контролирования безопасности, производительности и надёжности программы. От**

**этих требований напрямую зависит качество, а значит, рейтинг и популярность приложения.**