

Lecture 8

Особенности тестирования Desktop, WEB and Mobile-приложений.

Марьяна Макарова
Senior QA, QA Lead

QA Course

Особенности тестирования Desktop, Web и Mobile приложений.

Привет. Меня зовут Марьяна, я Senior QA и QA Lead, в DataArt работаю 10 лет, и за свою карьеру успела поработать со всеми видами приложений, которые мы будем сегодня рассматривать. Краткое содержание:

Содержание



✓ Десктоп приложения

- Что такое десктоп приложение
- Технические особенности
- Особенности тестирования

✓ Web приложения

- Что такое web-приложение
- Архитектура web-приложения
- Технические особенности
- Особенности тестирования

✓ Мобильные приложения

- Виды мобильных приложений
- Что такое нативные приложения
- Технические особенности
- Особенности тестирования

Desktop приложение это:

Desktop приложение это:



- ✓ Клиентское программное обеспечение, реализующее Windows Forms интерфейс.
- ✓ Приложение устанавливается на рабочую станцию пользователя и запускается локально.
- ✓ Это приложение, работающее под управлением операционной системы и в общем случае не требующее доступа к сети Интернет.



5

Каждый из вас сталкивался с такими приложениями, это те приложения, которые мы устанавливаем на компьютер: фоторедакторы, видеоигры или , например, Microsoft Office.

Технические особенности Desktop приложений.

Технические особенности



- ✓ Требуют установки и настройки
- ✓ Требуют определённой ОС
- ✓ Скорость работы не зависит от внешних факторов, зависит только от мощности и свободных ресурсов ЭВМ
- ✓ Занимают память компьютера
- ✓ Имеют стандартные интерфейсы
- ✓ Как правило, не требуют подключения к интернету

6

- Требуют установки и настройки. Это означает что для того, чтобы это приложение работало, мы его должны скачать или купить, установить его, настроить.

- Требуют определённой ОС, т.к. приложение пишется под определённую систему: Windows, MacOS, Linux.
- Скорость работы не зависит от внешних факторов, зависит только от мощности и свободных ресурсов ЭВМ. Не зависит от скорости подключения.
- Занимает память компьютера, поэтому для нашего приложения мы должны указывать, какие необходимы характеристики и объем памяти, чтобы запустить приложение.
- Имеют стандартные интерфейсы (windows forms), например, панель меню расположена на верху окна нашего приложения.
- Как правило, не требуют подключения к интернету, например, линейка приложений Microsoft Office. Но бывают исключения, например десктопная версия Skype и другие мессенджеры.

Особенности тестирования Desktop приложений

Особенности тестирования



- ✓ Кроссплатформенное тестирование
- ✓ Тестирование установки, удаления и обновления
- ✓ Тестирование совместимости (версия ОС, аппаратные ресурсы, другие приложения)
- ✓ Тестирование стабильности, стрессовое тестирование (утечки памяти и проч.)



Windows



Mac OS



Linux



7

На основании технических особенностей, рассмотрим **особенности тестирования**. Рассмотрим не все, а только те, которые специфичны для десктопных приложений.

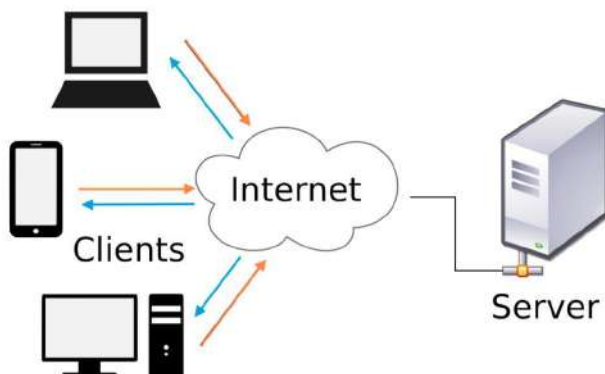
- Кроссплатформенное тестирование. Если мы выпускаем приложение под разные ОС, то на каждой из них мы должны проводить полный цикл тестирования.
- При тестировании установки проверяем, что оно корректно установилось и работает. При тестировании удаления смотрим, чтобы ничего не осталось, память компьютера не засоряется. При тестировании обновления смотрим, чтобы данные не были потеряны.
- Тестирование совместимости (версия ОС, аппаратные ресурсы, другие приложения). Помимо тестирования на каждой операционной системе, мы должны посмотреть на разных версиях ОС, т.к. обновления ОС выходят достаточно часто. Для тестирования нужно определить минимальную версию поддерживаемую системой и начинать тестировать с нее, желательно покрыть все версии, но если их много, то нужно выбирать. Должны быть требования по минимальным аппаратным ресурсам, при котором наше приложение будет работать корректно, ничего не будет зависеть, нужно

тестировать на минимальном наборе. Также другие приложения могут влиять на наше приложение, обычно это драйвера.

- **Тестирование стабильности, стрессовое тестирование** происходит при максимальной загрузке оперативной памяти. Ничего не должно зависать и падать, но если даже упало, то при следующем запуске приложения, всё должно быть сохранено и работать корректно.

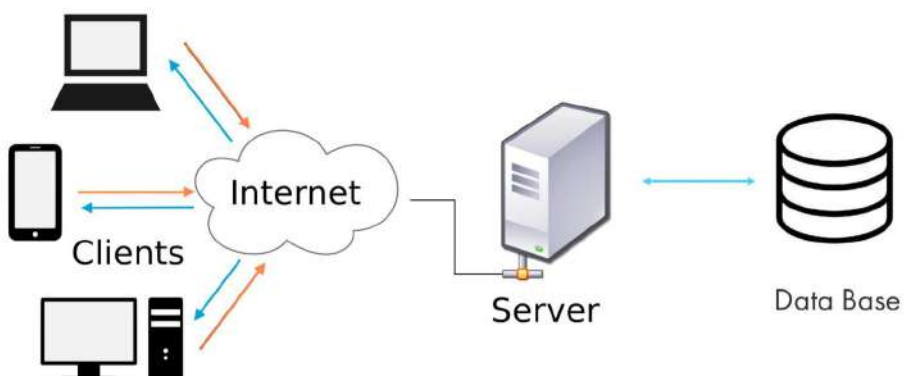
Архитектура Web-приложения.

Архитектура Web-приложения



10

Архитектура Web-приложения



11

- Клиент-серверное приложение, в котором клиентом выступает браузер, а сервером - веб-сервер;
- Логика веб-приложения распределена между сервером и клиентом;
- Хранения данных осуществляется, преимущественно, на сервере;
- Обмен информацией происходит по сети;

Мы все пользуемся веб-приложениями, например, это может быть интернет-магазин, онлайн банк или сервис покупки билетов.

Клиент - это браузер, клиент общается с сервером по сети, отправляет запрос на сервер, сервер ему отвечает, и юзер видит изменения в интерфейсе. Зачастую в этой схеме добавляется база данных (БД), где хранятся все данные. Клиент запрашивает информацию, сервер отправляет запрос в БД. Полученную из БД информацию сервер отправляет на клиент, где её видит юзер.

Пример



The image shows three sequential screenshots of a login form, illustrating the state of the interface during a login attempt.

- First Screenshot:** The form has two input fields: "Логин" (Login) and "Пароль" (Password). The "Логин" field has a placeholder "вход по номеру телефона" (login by phone number). Below the password field is a checkbox labeled "Запомнить меня" (Remember me). The "Продолжить" (Continue) button is present but has a greyed-out, inactive appearance.
- Second Screenshot:** The "Логин" field now contains the text "test". The "Пароль" field contains four dots, indicating masked input. The "Продолжить" button is now a solid green color, indicating it is active.
- Third Screenshot:** The "Логин" field still contains "test". The "Пароль" field is empty. Below the password field, an error message is displayed in red text: "Неверный логин или пароль. Если не можете войти, восстановите доступ." (Incorrect login or password. If you cannot log in, recover access). The "Продолжить" button remains active (green).

Пример: стандартные поля логин и пароль, кнопка продолжить - не активна. На второй картинке: после введения логина и пароля, кнопка становится активной - ярко-зелёной, сейчас всё ещё на стороне клиента, на сервер мы ничего не передавали. После нажатия кнопки Продолжить, клиент отправляет запрос на сервер, сервер проверяет в БД, есть ли такой логин и соответствуют ли ему пароль. На третьей картинке мы видим, что сервер ответил, что нет такого юзера или неверный пароль.

Пример



Логин

вход по номеру телефона

Пароль

☐ Запомнить меня

Продолжить

Логин

вход по номеру телефона

Пароль

☐ Запомнить меня

Продолжить



13

Если юзер ввел верный логин и пароль, то перед ним появляется, например, его юзер-профиль.

Технические особенности веб-приложений:

Технические особенности



- ✓ В общем случае платформонезависимые
- ✓ Требуют подключения к сети
- ✓ Зависят от конкретного браузера
- ✓ Зависят от скорости сети
- ✓ Скорость работы может зависеть от других пользователей
- ✓ Возможность влиять на работу других пользователей (изменение данных на сервере)

14

- В общем случае платформонезависимые, значит, что написанное приложение будет работать на любой ОС;

- Зависят от конкретного браузера, т.к. все браузеры очень разные, есть свои особенности, отображение и логика может работать по-разному;
- Требуют подключения к сети, т.к. вся передача данных происходит по сети, без неё они не смогут работать;
- Скорость работы может зависеть от других пользователей. Зависит от скорости сети. Все запросы обрабатываются параллельно, но есть лимит запросов. Если лимит пользователей превышен, то они выстраиваются в очередь, увеличивается время ожидания, замедляется скорость работы, что влияет на других юзеров;
- Возможность влиять на работу других пользователей (изменение данных на сервере). Например, один товар в интернет-магазине рассматривают сразу два пользователя. Один его покупает, другой только пытается купить, и видит, что товара уже нет. И это нужно учитывать при тестировании тоже.

Особенности тестирования Web-приложений

Особенности тестирования

DataArt

- ✓ Кросс-платформенное тестирование
- ✓ Нагрузочное тестирование
- ✓ Тестирование безопасности



15

Какие виды тестирования свойственны для веб-приложений:

- Кросс-платформенное тестирование. Должны обязательно проверить на выбранном наборе браузеров, на каждом сайт может отображаться по-разному, также может логика(front end) неправильно работать;
- Нагрузочное тестирование. Т.к. количество пользователей неограниченно, мы должны оценить потенциальное количество пользователей, которое будет пользоваться нашим приложением, и проверить наше приложение с данной нагрузкой. Часто нагрузочное тестирование является автоматизированным;
- Тестирование безопасности. Проводится отдельной командой тестировщиков, но до этого основная команда тестировщиков должна проверить базовые вещи, например, скрытие

пароля (чтобы он нигде не был в открытом виде, не передаваться в запросах), ролевой доступ (если есть приложение с разными ролями, например, админ и обычный юзер, что обычному юзеру недоступны функции админа), переход по прямой ссылке в раздел, который требует пароля (т.е. при переходе должно появиться поле для ввода логина и пароля).

Рассмотрим статистику популярности браузеров.

Статистика популярности браузеров по странам



Chrome	Safari	Edge	Firefox	Samsung Internet	Opera
65.21%	18.42%	3.77%	3.67%	2.9%	2.32%
Desktop, Mobile & Tablet Browser Market Share Worldwide - September 2021					
Chrome	Safari	Edge	Firefox	Samsung Internet	Android
50.46%	34.7%	5.96%	3.47%	1.97%	0.84%
Desktop, Mobile & Tablet Browser Market Share in United States Of America - September 2021					
Chrome	UC Browser	Safari	QQ Browser	360 Safe	Edge
48.78%	12.26%	11.12%	8.44%	8.2%	3.53%
Desktop, Mobile & Tablet Browser Market Share in China - September 2021					
Chrome	Yandex Browser	Safari	Opera	Samsung Internet	Firefox
63.25%	11.81%	7.5%	5.38%	4.45%	4.1%
Desktop, Mobile & Tablet Browser Market Share in Russian Federation - September 2021					

16

Исходя из статистики выбирается набор браузеров, на которых мы будем тестировать.

Требования к набору браузеров и их версий обычно выдвигает заказчик, но в реальности это бывает достаточно редко, и команда тестировщиков определяет их сама.

На картинке верхняя строка - мировая статистика, ниже по странам. По странам необходимо смотреть, т.к. приложение пишется под определенную аудиторию. На первом месте везде стоит Chrome. И это очень хорошая ситуация, т.к. основное тестирование мы всё можем проводить в Chrome. Но если мы обратим внимание на второе место, то по Штатам зачастую достаточно проверять по Chrome и Safari, и, допустим, покрыть Edge. В Китае ситуация очень отличается, проценты между 2-5 местом приблизительно одинаковые, и там довольно специфичный набор браузеров, и их все придется тестировать. В России мы должны протестировать Chrome, Yandex Browser и Safari, остальные в зависимости от ситуации, если есть ресурсы и время. И такую статистику мы можем посмотреть не только по Desktop.

Виды мобильных приложений:

Виды мобильных приложений



- ✓ Нативные приложения
- ✓ Мобильные web-приложения
- ✓ Гибридные приложения



18

- Нативные приложения пишутся под конкретную операционную систему;
- Мобильные web-приложения - это те же самые приложения, которые мы рассмотрели, только верстка адаптирована под экраны мобильных телефонов;
- Гибридные приложения совмещают в себе функции нативных и гибридных приложений. И тестируются они обычно так же, как нативные приложения.

Нативные мобильные приложения.

Нативные мобильные приложения

Это приложения, которые разрабатываются под конкретную аппаратно-программную платформу и пишутся на языках, созданных для данной платформы.

Нативные приложения создаются специально для запуска на целевой платформе - с поддержкой всех нативных технологий и аппаратных возможностей конкретной платформы.



19

Это приложения, которые разрабатываются под конкретную аппаратно-программную платформу и пишутся на языках, созданных для данной платформы. Нативные приложения создаются специально для запуска на целевой платформе - с поддержкой всех нативных технологий и аппаратных возможностей конкретной платформы.

Плюсы и минусы данных приложений:

Плюсы и минусы



- В приложении легко реализовать доступ ко всем функциям устройства (камера, GPS, Touch/Face ID и др.)
- Оптимизированы под ОС: работают быстро, потребляют меньше энергии, памяти и т.п.
- Нативный интерфейс
- Проверяются магазинами (AppStore, Google Play)
- Не обязателен интернет



- Дороговизна разработки и поддержки
- Необходимость установки и обновления
- Наличие ограничений при распространении через App Store/Google Play
- Занимают память телефона



20

“Не обязателен интернет” может касаться не всех приложений.

Пункт, что проверяется магазинами, есть в плюсах и минусах. Плюс в том, что при проверке в магазинах могут быть выявлены уязвимости, вероятность, что пользователь встретит проблему уменьшается. Но чтобы нам выпустить приложение, оно должно пройти модерацию в сторсах, на это уходит время, и много критериев, которым должно соответствовать приложение. Зачастую так бывает с App Store, что они могут отправить приложение на доработку. И например, если вы нашли баг в приложении и быстро пофиксили, зарелизить его не получится быстро, т.к. нужно пройти через проверку App Store.

Основной минус это дороговизна разработки и поддержки, т.к. для каждой операционной системы нужна отдельная команда разработчиков, под каждую систему пишется отдельное приложение. Соответственно, на тестирование требуется больше времени, повышается стоимость приложения. Также, необходимость установки и обновления, например, если пользователь обновил ОС, но не обновил приложение, оно может перестать корректно работать, и если у пользователя не включены автоматические обновления, он может зайти, а приложение не работает.

Также приложение занимает память телефона, если оно достаточно большое, то юзер подумает над альтернативами.

Технические особенности мобильных приложений.

Технические особенности



- ✓ Большое количество поддерживаемых платформ и устройств
- ✓ Низкие технические характеристики
- ✓ Высокие требования к безопасности
- ✓ Разнообразие взаимодействия с пользователем
- ✓ Зависимость от сети/канала связи
- ✓ Реакция на внешние прерывания

- Большое количество поддерживаемых платформ и устройств - это ОС, в основном iOS, Android. Также много поддерживаемых версий, много производителей на Android.
- Низкие технические характеристики, если мы сравним с десктопным компьютером, то характеристики телефона будут гораздо ниже. И мы обязательно должны учитывать, чтобы приложение не занимало много памяти, пользователь мог нормально ими пользоваться, а не думать, какое он может установить, потому что мало памяти.

- Высокие требования к безопасности. В телефонах хранится практически всё: личные данные, фото, банковские карты, пароли. Очень важно внимательно подходить к тестированию безопасности.
- Разнообразие взаимодействия с пользователем. В телефоне мы можем не только скроллить ленту, есть многообразие жестов: можно повернуть, потрясти и постучать - всё это нужно проверять, как реагирует наше приложение на жесты.
- Зависимость от сети/канала связи. Телефон может работать через wi-fi, мобильную сеть, все ситуации нужно проверять, чтобы приложение корректно работало.
- Реакция на внешние прерывание. В основном это звонки, push-уведомления от других приложений и SMS, подключения к другому устройству, зарядки, наушников, мы должны это учитывать в нашем приложении.

Особенности тестирования, которые свойственны для мобильных приложений.

Особенности тестирования



✓ Тестирование удобства использования

- ✓ Соответствие гайдлайнам (Material Design, Human Interface Guidelines)
- ✓ Размер и расположение элементов интерфейса
- ✓ Навигация между экранами, реакция на нативные жесты
- ✓ Обратная связь с пользователем
- ✓ Поворот экрана



22

- Usability testing- Приложение должно соответствовать guidelines (методическим рекомендациям). Для Android - это material design guidelines, для iOS - human interface guidelines. В них прописано, как должны выглядеть приложения, какие должны быть элементы и как работать. Разработчики, когда пишут приложения, должны всё учитывать, тестировщики - проверять, действительно ли наше приложение соответствует guidelines. Например, дизайнеры не могут нарисовать кнопку для удаления, это должна быть точно корзина. И например, нельзя использовать x - "крестик", потому что он используется для закрытия.
- Размер и расположение элементов интерфейса. Преимущественно все устройства с небольшими экранами, нам нужно это учитывать, чтобы приложением было удобно пользоваться, чтобы при нажатии на одни элементы, мы не задевали соседние.

- Навигация между экранами, реакция на жесты. Также переход между экранами может быть свайпом, по нажатию на какую-нибудь кнопку, оно может быть длинным или коротким, наше приложение должно корректно реагировать.
- Обратная связь с пользователем. Пользователь всегда должен понимать, что происходит, например, если грузится картинка, переход на другой экран - спиннер, progress bar (индикатор выполнения), т.е. понимание что идёт какой-то процесс, а не зависло приложение. Если не работает что-то или ошибка, то обязательно должно быть сообщение об ошибке.
- Поворот экрана. Казалось бы достаточно простая вещь, но иногда приложение может достаточно интересно реагировать на поворот экрана. Мы должны выяснить поддерживает ли приложение поворот экрана, или приложение только вертикальной ориентации. И также должны быть различные вариации, например, может быть при повороте экрана уведомление на телефоне.

Особенности тестирования совместимости.

Особенности тестирования



✓ Тестирование совместимости

- ✓ Совместимость с различными операционными системами
- ✓ Совместимость с различными устройствами
- ✓ Адаптируемость (размеры, разрешение экрана)
- ✓ Совместимость со сторонними сервисами



23

- Совместимость с различными операционными системами. Как уже говорили ранее, например, Android для каждого вендора может отличаться, и приложение может работать по-разному для каждого из этих устройств.
- Совместимость с различными устройствами. Должны выбрать набор устройств, тестировать на каждом из этих устройств, это достаточно сложно выполнимо, т.к. набор может быть достаточно большим. Поэтому, обычно пользуются эмуляторами, под Android это достаточно работоспособный вариант. Под iOS сложнее, эмулятор не будет полной копией того, что бы мы увидели на телефоне. Также можно использовать облачный сервис, который имеет в своей базе много различных телефонов и позволяет подключиться к ним удаленно для тестирования нашего приложения. Но это может быть не очень удобно, т.к. вы будете тестировать с экрана своего компьютера, не все кейсы можно проверить, и это платный сервис, не всегда проект на это идет.

- Адаптируемость (размеры, разрешение экрана). Набор размеров экрана очень большой, нам нужно определить минимальный, максимальный, и на всех вариантах протестировать приложение.
- Совместимость со сторонними сервисами. Могут быть платёжные системы или шаринг в социальных сетях.

Также свойственно для мобильных приложений:

- Тестирования подключения. Т.к. большинство приложений используют Интернет для работы, нам нужно проверить wi-fi, мобильный интернет (4g, 3g, EDGE), переход между wi-fi и мобильным интернетом, при полном отсутствии Интернета. Всё это можно проверить, создавая реальные условия, или использовать программы, которые это имитируют.
- Тестирование производительности - проверка на отсутствие утечек памяти, что память не засоряется, и если мы запустили много приложений, то наше корректно работает.
- Тестирование безопасности, схоже с пунктом у веб-приложений, мы должны проверить все пароли, что наши персональные данные не теряются.
- Тестирование установки, удаления и обновления. То же самое, что и для Desktop приложений, для начала нам его нужно установить, проверить, что во время установки мы поставили на паузу, а после оно корректно продолжило свою установку. Что после удаления оно удалилось полностью. После обновления нужно проверить, что все данные целы. В мобильных приложениях зачастую бывают покупки, нам нужно убедиться, что они все сохранены и ничего не потерялось.

Особенности тестирования



- ✓ Тестирование подключения
- ✓ Тестирование производительности
- ✓ Тестирование безопасности
- ✓ Тестирование установки, удаления и обновления
- ✓ Проверка реакции приложения на внешние прерывания



24

- Проверка реакции приложения на внешние прерывания - звонки, SMS, push-уведомления, подключения других устройств к телефону. Если наше приложение поддерживает какие-то устройства, то мы тестируем с этими устройствами, если нет, то они просто не должны влиять на наше приложение.

Статистика популярности мобильных устройств по странам.

Аналогично тому, что мы рассматривали для браузеров, мы можем посмотреть на популярность мобильных устройств по странам. Как мы видим, у каждой страны свой фаворит. Соответственно, когда мы выбираем набор устройств, мы должны знать нашу целевую аудиторию, и исходя из этого выбирать устройства.

На графике по Штатам всё просто - Apple и Samsung, остальные имеют низкий процент по популярности и скорее мы не будем их тестировать, если у нас нет лишнего времени. В России разброс достаточно большой и нам нужно будет тестировать и на Samsung, Xiaomi, Apple и Huawei.

Такую же статистику мы можем посмотреть для ОС мобильных устройств, статистику по версиям ОС, можем выбрать отдельно планшеты или мобильники. И уже исходя из этой статистики, мы выберем набор девайсов, на которых мы будем тестировать наше приложение.

Статистика популярности мобильных устройств по странам



Samsung	Apple	Xiaomi	Huawei	Oppo	Vivo
27.55%	26.75%	11.51%	7.88%	5.65%	4.57%
Mobile Vendor Market Share Worldwide - September 2021					
Huawei	Apple	Unknown	Xiaomi	Oppo	Vivo
37.88%	20.47%	14.25%	7.86%	6.35%	6.33%
Mobile Vendor Market Share in China - September 2021					
Apple	Samsung	LG	Motorola	Google	Unknown
57.08%	27.05%	4.04%	3.82%	2.12%	1.06%
Mobile Vendor Market Share in United States Of America - September 2021					
Samsung	Xiaomi	Apple	Huawei	Realme	Unknown
32.57%	18.2%	17.99%	15.62%	2.19%	1.79%
Mobile Vendor Market Share in Russian Federation - September 2021					

Questions

Вопросы:

- 1. Как тестируются вырезы под камеру для мобильных телефонов, т.к. иногда важные кнопки помещаются на место под камерой?**

Это тестируется при выборе устройства, это относится не к размеру экрана, а к модели телефона, к девайсу. Например, раньше iPhone были с кнопкой, а потом кнопка пропала, и все приложения перестали корректно отображаться, и их пришлось переписывать под новый телефон.

- 2. Например, есть десктопная игрушка, которая сперва ищет матчи во внутреннем браузере, а потом браузер запускает десктопное приложение. Какой это тип приложения?**

Если в игре можно играть по сети, то соответственно она работает через Интернет, и это будет гибридным приложением. Если оно использует сеть, то часть данных хранится на сервере, и будет то же самое, что с веб-приложениями, какие-то запросы отправляет на сервер и получать ответ. Это комбинация десктоп и веб-приложения.

- 3. Ко всем этим тестированиям пишутся тест-кейсы?**

В идеале да, но могут быть и чек-листы. Для мобильных устройств можно взять чек-лист, где должны проверить все комбинации, влияния внешних факторов.

- 4. Тестируем внешний вид веб-приложения (кнопки, панель ввода) на мобильном устройстве. Это относится к функциональному тестированию или не к функциональному тестированию?**

Если мы говорим о внешнем виде: размер (правильный ли размер), цвет, место расположения, - это usability testing, а оно относится к нефункциональному.

- 5. С какими приложениями больше всего вы сталкивались на практике?**

Самые популярные веб и мобильные приложения. Чаще всего это веб и опционально к нему мобильное приложение. Было пару проектов, где была только мобильная версия.

6. Вы приводили пример, когда при изменении дизайна iPhone пострадали многие приложения. Отсюда вопрос, могут ли разработчики узнавать об изменениях заранее или только по факту?

До релиза девайса всё хранится в секрете. До него компания может рассказать о каких-то новых особенностях. Но на практике только тогда, когда появляется девайс на руках, мы начнем тестировать, находить баги и фиксить. Что-то мы можем только предположить, вроде изменений размера экрана.

7. Вы говорили, что тестируется одновременно веб и мобильное приложение, в этом случае надо ли весь набор тест-кейсов прогонять в двух местах?

Да, обязательно. Например, если количество человек в команде позволяет, то можно разделить, часть тестирует веб, часть мобильное. И в идеале один тестирует Android, другой iOS. Обязательно все кейсы должны прогонять на каждой среде. Например, мы не будем прогонять все кейсы на всех наборах телефонов, мы выберем из приоритета.

8. Расскажите о личном опыте, внедрения фичей или ограничений от Store какой-нибудь страны.

У нас было приложение, которое ориентированно на китайскую аудиторию, и было много ограничений с доступом в Интернет. Были технические особенности, где должны располагаться сервера, чтобы пользователи могли туда достучаться.

9. Как тестировщики могут найти ошибки, которые создали разработчики при создании http-запросов?

Это тестирования API. Мы можем использовать DevTools, который можно открыть в каждом браузере. И мы увидим все запросы, которые наш клиент отправляет на сервер. Можем отследить, что он отправляет и правильно ли. Например, мы отправляет логин-пароль, но нам отвечают, что его нет. И мы можем зайти в DevTools, чтобы посмотреть, какой запрос клиент отправляет на сервер, передаёт ли он правильные параметры. Он должен передать наш логин и зашифрованный пароль.

Также можно напрямую тестировать http-запросы через Postman, без использования клиента будем напрямую отправлять запросы на сервер и смотреть, что отвечает сервер. Например, если через Postman запрос без ошибок, а через клиента он неверный, то возможно неправильно написан фронтенд.

10. При тестировании веб-приложений мы делаем запрос на сервер, и ожидаем, что он нам вернёт, к примеру 404 ответ, но он возвращает 500. Это тоже является багом?

Конечно. 404 ответ - это может быть, что страница не существует, а 500 ошибка - это сервер не смог выполнить запрос, и не факт, что это ошибка будет обрабатываться.

Вопросы для самостоятельного изучения:

1. Что такое Cash и Cookies?
2. Что такое http и https, чем они отличаются?
3. Что такое API? (Рассмотреть примеры запросов Get, Post)
4. Рассмотреть гибридные и нативные кроссплатформенные мобильные приложения.