**Этапы тестирования веб-сайтов**

Тестирование сайта — один из важных жизненных этапов, после которого, предоставляется заказчику готовый проект без ошибок, с хорошей читабельностью, воспринимаемой легкостью, удобством и надежностью. Мы хотели бы рассказать о том какие этапы проходит сайт перед выходом в продакшн.

Тестирование — это отклонение фактического результата от ожидаемого, другими словами — это процесс поиска багов (ошибок).

Основные правила тестирования веб-сайтов — это шаги, которые показывают пользователю, насколько удобен и логичен будет проект, насколько просто и возможно найти ту или иную информацию. Хорошо ли воспринимается человеческому взгляду и правильно ли работает весь функционал данного сайта, который был поставлен по ТЗ или брифу — это основные показатели для тестировщика.

**Как тестировать сайт?**

Тестирование может происходить самыми разными способами, однако не стоит забывать о самом процессе и стратегии тестирования. От него зависит последовательность ваших действий. На сегодняшний день, специалисты по тестированию веб-сайтов применяют такие виды как:

* Функциональное тестирование
* Тестирования удобства пользования (юзабилити)
* Тестирование производительности
* Тестирование интерфейса пользователя (UI testing)
* Тестирование безопасности.

Рассмотрим более подробно эти виды тестирования:

**Функциональное тестирование**

Один из важных и незаменимых видов тестирование. Самое главное правило функционального тестирования является правильные расчеты функций. К примеру, возьмем интернет магазин, у которого есть не только скидки на товар, но и множество статусов при покупке, n кол-во товаров. Все эти варианты следует учитывать. Ведь если функционал проекта не работает в определенном браузере, то он не будет работать нигде. Если брать определенный функционал для веб проектов, то в основном проверяется:

* поиск и покупка товара, оформление заказа
* навигация
* формы аутентификация
* добавление, удаление, редактирование товара, заказа и т.д.

**Юзабилити тестирование сайта**

Тестирование удобства пользования (юзабилити) – это вид тестирования, который делает для сайта удобство и практичность в использовании. Основная цель показать пользователю:

* Понятен ли ваш сайт для окружающих и удобен ли?
* Удобная навигация?
* Какое впечатление создается у пользователя?
* Что может быть лишним или не нужным.

Основная задача тестироания юзабилити сайта — это дизайн того, что пользователь хочет найти и приобрести, что он ищет необходимую информацию и для этого ему ничего не мешает.

**Нагрузочное тестирование сайтов**

Тестирование производительности — в основном это нагрузочное тестирование. Нагрузочное тестирование сайта проверяется в большинстве случаев автоматом, то есть специальными программами. Это дает шанс проверить, насколько он будет работать под определенной нагрузкой.

Цель этого тестирования, заключается в кол-ве виртуальных пользователей, которые задают n кол-во запросов, в одно время (будь это секунды даже). Тем самым результат дает то, смог ли наш проект выдержать, к примеру, 100 пользователей, которые одновременно покупали товар или авторизовались на сайте, ответ показывает, реально ли выдержать сайт такую нагрузку.

**Тестирование пользовательского интерфейса**

UI testing- это тестирование графического интерфейса пользователя, которая предполагает проверить сайт на соответствия требованиям к графическому интерфейсу, профессионально ли оно выглядит, выполнено ли оно в едином стиле.

В большинстве случаев, тестирование интерфейса пользователя, осуществляется вместе со следующими видами тестирования(UI):

1. Тестирование на соответствие стандартам графических интерфейсов
2. Тестирование с различными разрешениями экрана
3. Тестирование кроссбраузерности, или совместимости с разными интернет браузерами и их версиями
4. Тестирование локализованных версий: точность перевода (мультиязычность, мультивалютность), проверка длины названий элементов интерфейса и т.д
5. Тестирование графического интерфейса пользователя на целевых устройствах (смартфоны, кпп, планшеты).

**Тестирование сайта на уязвимости**

Это ключ к надежности веб-сайтов. Основные правила этого тестирования — это проверка на уязвимость разных видов атак. Если это интернет-магазин, то, скорее всего, следует проверять запросы на Sql инъекцию (запросы к базе данных).

SQL-инъекции — это вредоносный код в запросах базы данных — наиболее опасный вид атак.  Если это различные формы общественные (гостевая книга), то проверка на XSS инъекции. Дает возможность внедрить произвольный код, и атаковать компьютер пользователей, просматривающих зараженные страницы.

Вот, пожалуй, основные виды, которые используют для тестирования сайта.

Главная задача тестирования заключается в том, чтобы заказчику, как и пользователю, был предоставлен качественный продукт. Для этого перед началом тестирования любого сайта следует составить чек-лист для тестирования:

* проверки веб-форм
* проверки правильности данных
* тестирования интерфейса пользователя
* тестирования оплаты
* тестирования версий для печати
* тестирования отчетов.

**Основные виды тестирования сайта (веб-приложения)**

1. Тестирование функциональности;
2. Тестирование удобства использования;
3. Тестирование интерфейса;
4. Тестирование совместимости;
5. Тестирование производительности и скорости загрузки сайта;
6. Тестирование безопасности.

**Тестирование функциональности**

Проверьте все ссылки, присутствующие на веб-странице, а также ссылки на базы данных, формы, используемые для подтверждения действий и получения информации от пользователей, файлы **Cookie** и т.д.

**Проверьте все ссылки**

* Проверьте ссылки, исходящие от всех страниц к конкретному домену.
* Внутренние ссылки.
* Ссылки на другие элементы, расположенные внутри страниц.
* Ссылки для отправления электронной почты администратору или другим пользователям веб-страниц.
* Проверьте, нет ли ссылок на изолированные страницы.

**Проверьте формы**

Формы используются для получения информации от пользователей и взаимодействия с ними.

Что нужно проверить в формах:

* Правильность работы валидации в каждом поле формы.
* Значения полей, используемые по умолчанию.
* Опции для создания форм, удаления, просмотра и редактирования форм (*если такие имеются*).

Рассмотрим пример проекта поисковой системы, над которым я сейчас работаю. В проекте есть этапы регистрации рекламодателей и партнеров. Каждый шаг регистрации отличается от других, но зависит от остальных этапов. Поэтому весь процесс регистрации должен проходить правильно.

Есть различные виды валидации, например, проверка электронной почты, финансовой информации пользователя и т.д. Все поля с валидацией нужно протестировать в ручном или автоматическом режиме.

**Тестирование файлов cookie**

**Cookie** — это небольшие файлы, хранящиеся на компьютере пользователя. Чаще всего они используются для поддержки сеансов с авторизацией. Проверьте приложение, выключая и включая **cookies** в опциях браузера.

Проверьте, шифруются ли **Cookie** перед записью на компьютере. Протестируйте сеансы регистрации и статистику пользователя, когда сеанс посещения сайта закончится. Проверьте, влияет ли на безопасность приложения удаление файлов **cookie**.

**Проверьте HTML/CSS**

Если вы оптимизируете сайт для поисковых систем, то валидация **HTML/CSS** особенно важна. Первым делом проверьте сайт на наличие синтаксических ошибок в **HTML-коде**. Проверьте, доступен ли сайт для различных поисковых систем.

**Тестирование базы данных**

Взаимодействие веб-приложения с базой данных является очень важным моментом. Проверьте целостность данных и проведите **тестирование сайта на наличие ошибок** при редактировании, удалении, изменении форм или других действиях, имеющих отношение к базе данных.

Проверьте, все ли запросы к базе данных выполняются правильно, данные извлекаются и обновляются должным образом.

При тестировании функциональности сайтов нужно проверить:

**Ссылки**

1. Внутренние ссылки;
2. Внешние ссылки;
3. Ссылки на электронную почту;
4. Битые ссылки.

**Формы**

1. Валидация полей;
2. Сообщения об ошибке при неверном вводе;
3. Обязательные и необязательные к заполнению поля.

**База данных**

Следует проверить целостность базы данных.

**Тестирование удобства использования (юзабилити сайта)**

Тестирование юзабилити — это анализ взаимодействия пользователя и сайта, поиск ошибок и их устранение.

При этом проверяется:

* Легкость обучения;
* Навигация;
* Субъективная удовлетворенность пользователей;
* Общий вид.

**Проверка навигации**

Под навигацией подразумеваются средства для просмотра страниц пользователем. Это кнопки, блоки. А также то, как посетитель сайта использует ссылки на другие страницы.

Проверка юзабилити:

* Сайт должен быть простым в использовании;
* Инструкции должны быть очень четкими;
* Проверьте, достигают ли предоставленные инструкции поставленной цели;
* Главное меню должно быть доступно на каждой странице;
* Главное меню должно быть построено в логической последовательности.

**Проверка контента**

Контент должен быть логичным и простым для понимания. Проверьте текст на наличие ошибок. Применение темных цветов раздражает пользователей, не нужно использовать их в теме оформления.

Для контента и фона страницы лучше применять общепринятые стандарты, чтобы цвет шрифта, рамок и т.д. не раздражал пользователей.

Контент должен быть содержательным, ссылки работать надлежащим образом, изображения соответствующего размера. Это основные стандарты, соблюдаемые при веб-разработке. Ваша задача — проверить все в рамках тестирования пользовательского интерфейса.

**Другая информация для пользователей**

Варианты поиска, карта сайта, справочные материалы и т.д. Проверьте работу всех ссылок в карте сайта. Функция «*Поиск по сайту*» должна помогать легко находить нужный контент.

**Тестирование пользовательского интерфейса**

Нужно проверить, правильно ли осуществляется связь с сервером. Следует проверить совместимость сервера с используемым программным обеспечением, аппаратными средствами, сетью и базой данных.

Основные интерфейсы:

* Интерфейсы веб-сервера и приложения.
* Интерфейсы сервера базы данных и сервера приложения.

Если база данных или веб-сервер для какого-либо запроса, исходящего от сервера приложения, возвращает сообщение об ошибке, сервер приложения должен фиксировать его и отображать пользователю.

Проверьте, что происходит, когда пользователь прерывает какое-либо действие. А также, что происходит при повторном подключении к серверу в ходе выполнения какой-либо операции.

**Проверка совместимости**

Нужно проверить:

* Совместимость с браузерами;
* Совместимость с операционными системами;
* Просмотр на мобильных устройствах;
* Параметры печати.

**Совместимость с браузерами**

Работа некоторых веб-приложений зависит от типа браузера. Сайт должен быть совместим с различной конфигурацией и параметрами разнообразных браузеров.

Верстка сайта должна быть кроссбраузерной. При использовании **Java-скриптов** и **AJAX**, обеспечивающего функциональность пользовательского интерфейса, проверки безопасности или валидации создают большую нагрузку на систему.

Проверьте работу веб-приложения в браузерах **Internet Explorer**, **Firefox**, **Netscape Navigator**, **AOL**, **Safari**, **Opera** разных версий.

**Совместимость с операционными системами**

Некоторые функции веб-приложения могут быть несовместимы с определенными операционными системами. Не во всех из них поддерживаются новые технологии, используемые в веб-разработке. Поэтому проверьте работу приложения в **Windows**, **Unix**, **MAC**, **Linux**, **Solaris** и их различных версиях.

**Просмотр на мобильных устройствах**

Проведите **тестирование сайта на мобильных устройствах**и проверьте, как просматриваются веб-страницы с помощью мобильных браузеров. Проблемы с совместимостью также могут возникнуть из-за мобильных устройств. Также не стоит забывать о **тестировании сайта на разных разрешениях.**

**Параметры печати**

Если вы предусматриваете возможность печати страницы, удостоверьтесь, что шрифты, выравнивание, графика и т. д. отображаются на бумаге должным образом. Страницы должны подходить под размеры, которые устанавливаются в опциях печати.

**Тестирование производительности сайта**

Тестирование производительности сайта или веб-приложения должно включать в себя:

* Нагрузочное тестирование.
* Стрессовое тестирование.

Проверьте производительность приложения на различной скорости интернета.

**Нагрузочное тестирование сайта**(*веб-приложения*) — это тестирование, при котором большое количество пользователей одновременно выполняют запрос к одной и той же странице. Выдерживает ли система пиковые нагрузки?

Стрессовое тестирование — нагрузка системы, выходящая за пределы установленных лимитов. Стрессовое тестирование выполняется с целью достичь сбоя в работе сайта или веб-приложения путем увеличения нагрузки. А также проверить, как система реагирует на стресс, и как она восстанавливается после сбоев. Стрессовой нагрузке подвергают поля для ввода информации, входа и регистрации.

**ab тестирование**функциональности также включает в себя проверку на ошибки, связанные с оперативной памяти.

Тест производительности можно применять для проверки масштабируемости сайта или оценки продуктивности при использовании стороннего программного обеспечения.

**Скорость соединения**

**Сплит тестирование сайта** при использовании различных вариантов интернет-соединения: через модем, **ISDN** и т.д.

**Нагрузка**

1. Количество пользователей, одновременно посещающих сайт;
2. Проверьте работу системы при пиковых нагрузках;
3. Пользователь осуществляет доступ к большому количеству данных.

**Стрессовая нагрузка**

1. Непрерывная нагрузка;
2. Производительность памяти, процессора, обработки файлов и т. д.

**Тестирование безопасности**

Ниже приведены некоторые наборы для тестирования веб-безопасности:

* Проверка с помощью вставки внутреннего **URL** в адресную строку браузера без авторизации. Внутренние страницы при этом не должны открываться.
* После авторизации с помощью логина и пароля, а также просмотра внутренних страниц попробуйте изменять **URL**. Например, вы проверяете какую-то статистику сайта под идентификатором **ID= 123**. Попробуйте изменить **ID URL** на другой **ID** сайта, который не имеет отношения к авторизованному пользователю. В любом случае доступ этого пользователя к просмотру других показателей должен быть запрещен.
* Попробуйте ввести неверные данные в поля формы для авторизации. Выясните, как система реагирует на ввод недопустимых данных.
* Каталоги или файлы не должны быть доступны напрямую, если для них не предусмотрена возможность скачивания.
* Проверьте работу капчи для защиты от автоматического входа с помощью программного кода.
* Проверьте, используется ли в целях безопасности **SSL**. Если да, то должно отображаться сообщение при переходе пользователя с незащищенных **HTTP-страниц** к защищенным и наоборот.
* Все операции, сообщения об ошибках, нарушения безопасности должны записываться в файл журнала на веб-сервере.

Основной причиной тестирования безопасности сайта является поиск потенциальных уязвимостей и их последующее устранение.

* Сетевое сканирование;
* Сканирование уязвимостей;
* Возможность потенциального взлома паролей;
* Обзор журнала;
* Средства для проверки целостности;
* Обнаружение вирусов.

**Моменты, которые следует учитывать при тестировании сайта**

Следует обратить внимание на взаимодействие **HTML-страниц**, интернет-подключение, брандмауэры, приложения, запускаемые на веб-страницах (*апплеты,****JavaScript****, модульные приложения*), а также приложения, работающие на стороне сервера (*скрипты****CGI****, интерфейсы баз данных, генераторы динамических веб-страниц*).

Есть множество типов серверов и браузеров различных версий. Между ними есть небольшие, но значимые различия.

**Пример сценариев тестирования сайта**

Дополнительные факторы, которые следует учесть при тестировании сайта:

* Какова ожидаемая нагрузка на сервер (*например, количество запросов за единицу времени*)?
* Какая производительность требуется при различных видах нагрузки (*время ответа веб-сервера, время отклика базы данных на запрос*)?
* Какие инструменты потребуются для тестирования производительности?
* Кто является целевой аудиторией? Какие браузеры будут использовать пользователи? Какова скорость подключения? Предназначен ли сайт для использования внутри организации или будет доступен в интернете для широкого круга пользователей?
* Какую производительность ожидает получить клиент (*насколько быстро должны загружаться страницы, как должны себя вести анимации, апплеты, нагрузка и запуск*)?
* Будут ли разрешены простои сервера и техническое обслуживание, а также обновление контента? Если да, в каком количестве?
* Какие средства безопасности требуются (*файерволы, шифрование, пароли и т.д.*), и какую работу они будут выполнять? Как их можно проверять?
* Насколько надежным должно быть интернет-соединение? Как оно будет влиять на резервное копирование системы?
* Как будет выполняться управление обновлением контента сайта?
* Требования для технического обслуживания, отслеживания и контроля содержимого веб-страниц, графических элементов, ссылок и т.д.
* Какая спецификация **HTML** будет соблюдаться? Насколько точно?
* Как будут проверяться и обновляться внутренние и внешние ссылки? Насколько часто?
* Как будет происходить управление и проверка **CGI** апплетов, сценариев **JavaScript**, компонентов **ActiveX** и т.д.?
* Максимальный размер веб-страницы не должен превышать 3-5 экранов, кроме случаев, когда контент сосредоточен на одной теме. Если размер веб-страницы больше, предоставьте внутренние ссылки для навигации по ней.
* Разметка веб-страницы и элементы дизайна должны быть последовательными и логично связанными.
* Отображение веб-страниц должно быть независимо от типа браузера.
* На каждой странице следует указать ссылку для связи.