

WEB-Testing_checklist

- собрать основные виды тестирования веб-приложений (функциональные и нефункциональные) и их особенности (например Web-формы, кроссбраузерность, функциональное тестирование).

Составляющие веб-приложения

- **Клиент**

Как правило, **клиент** – это браузер, но встречаются и исключения (в тех случаях, когда один веб-сервер (BC1) выполняет запрос к другому (BC2), роль клиента играет веб-сервер BC1).

- **Сервер**

Веб-сервер – это сервер, принимающий HTTP-запросы от клиентов и выдающий им HTTP-ответы.

*Отметим, что веб-сервером называют как программное обеспечение, выполняющее функции веб-сервера, так и непосредственно компьютер, на котором это программное обеспечение работает. Наиболее распространенными видами ПО веб-серверов являются Apache, IIS и NGINX.

- **База данных**

– информационная модель, позволяющая упорядоченно хранить данные об объекте или группе объектов, обладающих набором свойств, которые можно категоризировать.

Фактически, **база данных** не является частью веб-сервера, но большинство приложений просто не могут выполнять все возложенные на них функции без нее, так как именно в базе данных хранится вся динамическая информация приложения (учетные, пользовательские данные и пр).

Основные виды тестирования веб-приложений

Функциональные

Функциональное тестирование является крайне важным для любого приложения, в том числе веб-приложения. Функциональное тестирование гарантирует, что веб-приложение работает правильно и корректно.

- **Модульное тестирование (*Unit Testing*)**
– проверка небольших отдельных областей приложения на ранних стадиях разработки, чтобы снизить вероятность перехода к более серьезным ошибкам в дальнейшем.

Тесты приемки (*аналоги друг друга*):

- **Дымовое тестирование (*Smoke Testing*)**
– проверка веб-приложения на стабильность и наличие явных ошибок.
 - **Тестирование сборки (*Build Verification Testing*)**
– тестирование направленное на определение соответствия, выпущенной версии, критериям качества для начала тестирования
-
- **Санитарное тестирование (*Sanity Testing*)**
– специфическая проверка работоспособности отдельных функциональных элементов, систем, веб-архитектур и расчетов.
 - **Тестирование локализации (*Localization Testing*)**
– оценка правильности версии программного продукта (языковой и культурный аспекты).
 - **Регрессионное тестирование (*Regression Testing*)**
– проверка веб-приложения на корректность дополнений, соответствие установленным требованиям после изменений и успешное взаимодействие с другими системами.
 - **Тестирование совместимости** – тестирование работы программного продукта в определённом окружении.
 - **Интеграционное тестирование (*Integration Testing*)**
– проверка, что приложение совместимо со сторонними сервисами.

- **Юзабилити-тестирование (*Usability Testing*)**
– проверка удобства пользования интерфейсом веб-приложения.

Функциональное тестирование также проверяет формы, чтобы убедиться, что они работают.

Нефункциональные

проверка корректности работы нефункциональных требований. Оценивается, КАК программный продукт работает. Эта проверка включает в себя следующие виды:

- **Тестирование пользовательского интерфейса (*UI testing*)**
– это тестирование графического интерфейса пользователя, которое предполагает проверить сайт на соответствия требованиям к графическому интерфейсу, профессионально ли оно выглядит, выполнено ли оно в едином стиле.
- **Тестирование производительности**
– работа приложения под определённой нагрузкой.
Нагрузочное тестирование сайта проверяется в большинстве случаев автоматом, то есть специальными программами. Это дает шанс проверить, насколько он будет работать под определенной нагрузкой.
- **Тестирование защищенности** – определение безопасности ПО: защищено ли оно от атак хакеров, несанкционированного доступа к данным и т. д.

Особенности тестирования веб-приложений

Особенности архитектуры:

- **Кроссбраузерность: разнообразие клиентов**

Тестирование на кроссбраузерность - это проверка на правильность (соответствие требованиям и стандартам) отображения и функционирования веб-приложения в разных браузерах и на разных операционных системах.

Благодаря стандартизации большинство популярных браузеров одинаково обрабатывают код. При этом необходимость в кроссбраузерном тестировании не исчезает, так как далеко не все проблемы решаются стандартизацией.

- **Веб-формы на стороне клиента**

Одной из важных составляющих интернет-приложений являются формы для заполнения, взаимодействие с которыми пользователь осуществляет с помощью клиента.

Однако, данные формы очень часто служат источником дефектов, которые, обосновавшись в «продакшене», могут принести большие финансовые и репутационные убытки компании.

- **Текст, как основной источник информации при работе через клиент**

Часть информации сети Интернет представлена в виде текстов, с которыми, опять же, пользователь взаимодействует посредством клиента. Большинство веб-ресурсов в том или ином объеме требуют проверки текстов на предмет отсутствия грамматических ошибок и опечаток.

Веб-сервер

- **Инсталляция на веб-сервер**

Прежде чем приступить к тестированию, нужно установить (инсталлировать) веб-приложение на веб-сервер как и при проверке десктоп-приложений. Но существует и различие в нюансах, которые необходимо учесть и протестировать, особенно если это касается ПО, распространяемого для локальной инсталляции на веб-серверы пользователей.

- **Нагрузочное тестирование**

Нагрузочное тестирование имитирует работу с приложением определенного количества пользователей. Этот вид тестирования осуществляется при помощи специальных инструментов (например, jMeter), главная цель которых – определить профили нагрузки и искусственно создать для них нагрузку, выявляющую граничные возможности приложения (или сервера) в условиях работы с ним того или иного количества пользователей.

База данных

В базе данных приложение хранит всю необходимую информацию. Для того, чтобы база данных служила достойным хранилищем информации для вашего приложения, при тестировании необходимо так же обращать внимание некоторые нюансы.