

## Полный чек-лист тестирования веб-приложения

Оригинал статьи: <http://www.guru99.com/complete-web-application-testing-checklist.html>

Автор: Guru99.com

Перевод: Ольга Алифанова

Во время тестирования веб-приложения нужно обращать внимание на нижеперечисленные пункты. Этот чеклист применим практически к любому типу веб-приложений в зависимости от бизнес-требований.

Чек-лист для тестирования веб-приложений состоит из:

- Тестирования удобства использования.
- Функционального тестирования.
- Тестирования совместимости.
- Тестирования баз данных.
- Тестирования безопасности.
- Тестирования производительности.

Теперь давайте рассмотрим каждый пункт по отдельности.

### Тестирование удобства использования

Что это такое?

- Это не что иное, как тестирование дружелюбности приложения для пользователя.
- При тестировании удобства использования проверяется, легко ли новому пользователю разобраться в приложении.
- В целом при тестировании удобства использования тестируется системная навигация.

Какова цель этого тестирования?

Тест удобства использования удостоверяется в простоте и эффективности использования продукта при использовании стандартных практик тестирования удобства использования.

### Сценарии тестирования удобства использования:

- Содержание веб-страницы верное, без грамматических и орфографических ошибок.
- Все шрифты соответствуют требованиям.
- Все тексты правильно выровнены.
- Все сообщения об ошибках верные, без орфографических и грамматических ошибок, и соответствуют заголовку окна.
- Подсказки существуют для всех полей.
- Все поля правильно выровнены.

- Между полями, колонками, рядами и сообщениями об ошибках оставлено достаточно свободного места.
- Все кнопки должны иметь стандартный формат и размер.
- Ссылка на домашнюю страницу должна быть на каждой странице сайта.
- Неактивные поля должны быть серыми.
- Проверьте, что на сайте нет битых ссылок и изображений.
- Подтверждающие сообщения должны отображаться для всех операций обновления и удаления.
- Проверьте сайт при разных разрешениях экрана ((640 x 480, 600x800 и т. д.)
- Проверьте, что пользователь может пользоваться системой без раздражения.
- Проверьте, что TAB правильно работает.
- Панель скrolла должна появляться только тогда, когда она требуется.
- Если при отправке формы есть сообщения об ошибке, в нем должна содержаться информация, переданная пользователем.
- Заголовок должен отображаться на каждой странице.
- Все поля (текстовые, выпадающие меню, радио-кнопки и т. д.) и кнопки должны быть доступны с клавиатуры, и пользователь должен быть в состоянии пользоваться сайтом, используя только клавиатуру.
- Убедитесь, что данные в выпадающих списках не обрезаются из-за размеров поля, и проверьте, защищены ли данные в код или управляются администратором.

### **Функциональное тестирование**

Что это?

- Тестирование функциональностей и операционного поведения продукта с целью убедиться, что они соответствуют спецификациям.
- Тестирование, игнорирующее внутренние механизмы системы или компонента. Оно концентрируется исключительно на выходных данных, полученных в ответ на пользовательский ввод и условия исполнения сценариев.

### **Какова цель функционального тестирования?**

Цель функционального тестирования – убедиться, что ваш продукт соответствует нужной функциональной спецификации, упомянутой в вашей документации по разработке.

### **Сценарии функционального тестирования:**

- Протестируйте валидацию всех обязательных полей
- Убедитесь, что знак звездочки отображается у всех обязательных полей

- Убедитесь, что система не отображает окно ошибки при незаполненных необязательных полях.
- Убедитесь, что високосные коды корректно валидируются и не вызывают ошибок в расчетах.
- Протестируйте числовые поля: они не должны принимать буквы, в этом случае должно отображаться соответствующее сообщение об ошибке.
- Протестируйте отрицательные значения в числовых полях, если они разрешены.
- Протестируйте, что деление на ноль верно обсчитывается.
- Протестируйте максимальную длину каждого поля, чтобы убедиться, что данные не обрезаются.
- Протестируйте всплывающее сообщение ("Это поле ограничено 500 знаками"), которое должно отобразиться, если введенные данные превышают разрешенный размер поля.
- Убедитесь, что подтверждающее сообщение отображается для операций обновления и удаления.
- Убедитесь, что значения стоимости отображаются в нужной валюте.
- Протестируйте все поля ввода на спецсимволы.
- Протестируйте функциональность тайм-аута.
- Протестируйте функциональность сортировки.
- Протестируйте функциональность доступных кнопок.
- Протестируйте условия использования и часто задаваемые вопросы: они должны быть внятными и доступными пользователю.
- Протестируйте, что при отказе функциональности пользователь перенаправляется на специальную страницу ошибки.
- Протестируйте, что все загруженные документы правильно открываются.
- Протестируйте, что пользователь может скачать загруженные файлы.
- Протестируйте почтовую функциональность системы.
- Протестируйте, что Java Script верно работает в разных браузерах (IE, Firefox, Chrome, Safari, Opera).
- Посмотрите, что будет, если пользователь удалит куки, находясь на сайте.
- Посмотрите, что будет, если пользователь удалит куки после посещения сайта.
- Протестируйте все данные в выпадающих списках: они должны быть расположены в хронологическом порядке.

## **Тестирование совместимости**

Что это?

- Тестирование совместимости используется, чтобы убедиться, что ваше приложение совместимо с другими элементами системы, в которой оно работает – например, браузерами, операционными системами или железом.

#### **Какова цель тестирования совместимости?**

- Цель тестирования совместимости – оценка того, насколько хорошо ПО работает в определенном браузере, под определенной ОС, с другим ПО или железом.

#### **Сценарии тестирования совместимости:**

- Протестируйте сайт в различных браузерах (IE, Firefox, Chrome, Safari, Opera) и убедитесь, что сайт правильно отображается.
- Убедитесь, что используемая версия HTML совместима с соответствующими версиями браузеров.
- Убедитесь, что картинки корректно отображаются в разных браузерах.
- Убедитесь, что шрифты верно отображаются в разных браузерах.
- Убедитесь, что Java Script код работает в разных браузерах.
- Проверьте анимированные GIF в разных браузерах.

#### **Инструмент для тестирования совместимости**

Spoon.net: Spoon.net предоставляет доступ к тысячам приложений (браузеров), не требующих установки. Этот инструмент помогает вам тестировать приложение в разных браузерах на одной и той же машине.

#### **Тестирование баз данных**

Что это?

- При тестировании баз данных проверяются бэкэнд-записи, введенные через веб или десктоп-приложение. Данные, которые отображаются в приложении, должны совпадать с данными, хранящимися в базе данных.

#### **Чтобы тестировать базы данных, тестировщик должен знать следующее:**

- Тестировщик должен понимать функциональные требования, бизнес-логику, основной сценарий приложения и дизайн базы данных.
- Тестировщик должен разбираться в таблицах, триггерах, процедурах хранения, способах отображения и указателях, используемых для приложения.
- Тестировщик должен понимать логику триггеров, процедур хранения, способов отображения и указателей.
- Тестировщик должен понимать, какие таблицы затрагиваются, когда операции вставки, обновления и удаления выполняются в приложении.

**Понимая вышеперечисленные пункты, тестировщик может легко написать сценарии для тестирования баз данных.**

**Сценарии тестирования баз данных:**

- Проверьте название базы данных: оно должно совпадать со спецификацией.
- Проверьте таблицы, колонки, типы колонок и значения по умолчанию: все это должно совпадать со спецификацией.
- Проверьте, позволяет ли колонка значение null.
- Проверьте первичный и внешний ключ каждой таблицы.
- Проверьте процедуры хранения.
- Протестируйте, установлена ли процедура хранения.
- Проверьте название процедуры хранения.
- Проверьте названия параметров, их типы и количество.
- Проверьте, обязательны параметры или нет.
- Проверьте процедуру хранения, удалив некоторые параметры.
- Проверьте базу данных, если на выходе ноль – записи с нулем должны быть задействованы.
- Проверьте процедуру хранения, задав простые SQL-запросы.
- Убедитесь, что процедура возвращает значения.
- Проверьте процедуру вводом тестовых данных.
- Проверьте поведение каждого флага в таблице.
- Убедитесь, что данные правильно сохраняются в базе данных после каждого ввода.
- Проверьте данные при каждой операции обновления, удаления и вставки.
- Проверьте длину каждого поля. Длина на бэкэнде и фронтэнде должны совпадать.
- Проверьте названия баз данных QA, UAT и прода. Имена должны быть уникальными.
- Проверьте зашифрованные данные в базе.
- Проверьте размер базы и время отклика на каждый запрос.
- Проверьте данные, отображающиеся на фронтэнде, и убедитесь, что они совпадают с бэкэндом.
- Проверьте целостность данных, вводя невалидные значения в базу.
- Проверьте триггеры.

## **Что такое тестирование безопасности?**

Тестирование безопасности нацелено на поиск недостатков и пробелов с точки зрения безопасности приложения.

### **Сценарии тестирования безопасности:**

1. Убедитесь, что страницы, содержащие важные данные (пароль, номер кредитной карты, ответы на секретные вопросы и т. п.) открываются через HTTPS (SSL).
2. Убедитесь, что важная информация (пароль, номер кредитной карты) отображается в зашифрованном виде.
3. Убедитесь, что правила создания паролей внедрены на всех страницах авторизации (регистрация, страница "забыли пароль", смена пароля).
4. Убедитесь, что если пароль изменен, пользователь не может зайти под старым.
5. Убедитесь, что сообщения об ошибках не содержат никакой секретной информации.
6. Убедитесь, что если пользователь вышел из системы или сессия завершена, он не может пользоваться сайтом.
7. Проверьте доступ к закрытым и открытым страницам сайта напрямую без авторизации.
8. Убедитесь, что опция "Просмотр исходного кода" отключена и не видна пользователю.
9. Убедитесь, что учетная запись пользователя блокируется, если он несколько раз ввел пароль неверно.
10. Убедитесь, что пароль не хранится в куки.
11. Убедитесь, что если какая-либо функциональность не работает, система не отображает информацию о приложении, сервере или базе данных. Вместо этого отображается соответствующее сообщение об ошибке.
12. Проверьте сайт на SQL-инъекции.
13. Проверьте права пользователей и их роли. К примеру, кандидат не должен быть способен получить доступ к странице администратора.
14. Убедитесь, что важные операции пишутся в логи, и информацию можно отследить.
15. Убедитесь, что значения сессий отображаются в адресной строке в зашифрованном виде.
16. Убедитесь, что куки хранятся в зашифрованном виде.
17. Проверьте приложение на устойчивость к брутфорс-атакам.

## **Что такое тестирование производительности?**

Тестирование производительности проводится для оценки соответствия системы или компонента специфичным требованиям к производительности.

### **Общие тестовые сценарии:**

- Определение производительности, стабильности и масштабируемости приложения под разной нагрузкой.
- Определение, может ли актуальная архитектура поддерживать приложение при пиковых нагрузках.
- Определение, какая конфигурация приводит к наилучшим показателям производительности.
- Определение бутылочного горла приложения и инфраструктуры.
- Определение, не изменилось ли время отклика у новой версии приложения.
- Оценка продукта и/или железа с целью удостовериться, что они выдержат прогнозируемые объемы нагрузки.

### **Как проводится тестирование производительности? Вручную или автоматически?**

В целом невозможно проводить тестирование производительности вручную по ряду причин:

- Понадобится большое количество ресурсов.
- Невозможно одновременно осуществлять ряд действий.
- Отсутствует подходящий способ отслеживания поведения системы.
- Сложность выполнения повторяющихся задач.

Чтобы справиться с вышеперечисленными проблемами, используются специальные инструменты тестирования производительности. Ниже перечислены некоторые из них:

- Apache JMeter
- Load Runner
- Borland Silk Performer.
- Rational Performance Tester
- WAPT
- NEO LOAD