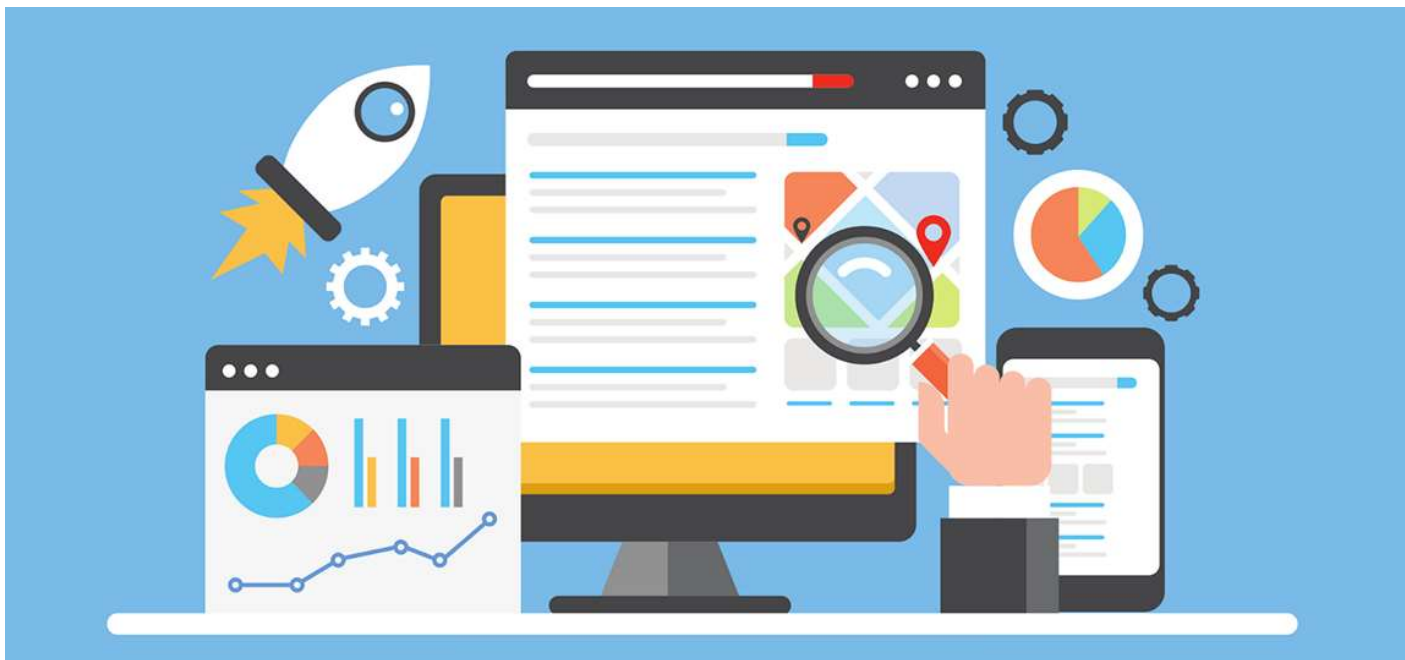


Чек-лист тестирования WEB приложений

🕒 5 мин 👁 162K

Тестирование веб-сервисов*

Привет! После публикации статьи «Чек-лист тестирования мобильных приложений», поступило большое количество сообщений про такой же чек-лист, только для WEB приложений. Чтобы ответить на этот вопрос была подготовлена универсальная шпаргалка, которую можно использовать при тестировании практически любого WEB приложения.



В данный чек-лист вошли только общие характеристики. Естественно, в тестируемом приложении может быть функциональность, для которой нужно применять отдельный подход и создать отдельные сценарии. То же самое верно для производительности, удобства использования, безопасности и прочего тестирования, которое необходимо вашему приложению.

Чек-лист для тестирования WEB приложений состоит из шести разделов:

- Функциональное тестирование
- Интеграционное тестирование
- Тестирование безопасности
- Тестирование локализации и глобализации
- Тестирование удобства использования
- Кросс-платформенное тестирование

Функциональное тестирование

В данном пункте нам важно убедиться, что наш продукт соответствует нужной функциональной спецификации, упомянутой в документации по разработке.

Что проверяем?

1. Тестирование форм

1. Регистрация

1. Пользователь с данными существует в системе.
2. Пользователь с данными не существует в системе.
3. Пользователь, заблокированный в системе, не может пройти повторную регистрацию.

2. Авторизация

1. Пользователь существует в системе с введенным логином и паролем.
2. Пользователь с введенным логином не существует в системе.
3. Пользователь с введенным логином существует в системе, но пароль неверный.
4. Пользователь с введенным логином и паролем существует в системе, но заблокирован модерацией (страница заморожена).
5. Валидация полей ввода.

3. Протестируйте валидацию всех обязательных полей

1. Максимальная и минимальная длина.
2. Диапазон допустимых символов, спецсимволы.
3. Обязательность к заполнению.
4. Убедитесь, что астериск (знак звездочки) отображается у всех обязательных полей.
5. Убедитесь, что система не отображает окно ошибки при незаполненных необязательных полях.

4. Формы обратной связи

5. Ссылки на пользовательские соглашения

2. Поиск

1. Результаты существуют/не существуют.
2. Корректное сообщение о пустом результате.
3. Пустой поисковой запрос.
4. Поиск по эмодзи.

3. Поля

1. Числовые поля: они не должны принимать буквы, в этом случае должно отображаться соответствующее сообщение об ошибке.
2. Дробные значения, например, как система валидирует 1.1 и 1,1.
3. Отрицательные значения в числовых полях, если они разрешены.
4. Деление на ноль корректно обрабатывается.
5. Протестируйте максимальную длину каждого поля, чтобы убедиться, что данные не обрезаются или скрываются под многоточие.
6. Протестируйте все поля ввода на спецсимволы.
7. Проверить что текст не выезжает за границы поля.

4. Всплывающие сообщения

1. Протестируйте всплывающие сообщения («Это поле ограничено N знаками»).
2. Подтверждающие сообщения отображается для операций обновления и удаления.
3. Сообщения об ошибках ввода.

5. Фильтры

1. Протестируйте функциональность сортировки (по возрастанию, по убыванию, по новизне).
2. Задать фильтры с выдачей.
3. Задать фильтры, по которым нет выдачи.
4. Фильтры по категориям/подкатегориям.
5. Фильтры с радиусом поиска.
6. Данные в выпадающих списках.

6. Протестируйте функциональность доступных кнопок.
7. Наличие favicon.
8. Проверка обработки различных ошибок (страница не найдена, тайм-аут, ошибка сервера и т.д.).
9. Протестируйте, что все загруженные документы правильно открываются.
10. Пользователь может скачать/прикрепить/загрузить файлы/медиа (картинки, видео и т.д.).
А также удалить эти файлы из вложений. Убедитесь, что файлы уходят на сервер только после нажатия соответствующей кнопки
11. Протестируйте почтовую функциональность системы.
12. Кеш, cookie и сессии

1. Пользователь очистил кэш браузера
2. Посмотрите, что будет, если пользователь удалит куки, находясь на сайте.
3. Посмотрите, что будет, если пользователь удалит куки после посещения сайта.

13. DevTools

1. Ошибки в Console.
2. Все стили загружаются.
3. Картинки загружаются.

Интеграционное тестирование

Интеграционное тестирование проводится для того, чтобы убедиться, что ваше приложение совместимо со сторонними сервисами.

Что проверяем?

1. Проверяем работу сторонних модулей: оплата, шаринг, карты.
2. Реклама (просмотр, переходы по рекламе, аналитика).
3. Метрики (переходы по страницам, показы элементов, клики).

Тестирование безопасности

Данная проверка нацелена на поиск недостатков и пробелов с точки зрения безопасности нашего приложения.

Что проверяем?

1. Пользователь не может авторизоваться: под старым паролем, заблокирован в сервисе, достиг лимита авторизаций, ввел чужой код верификации.
2. Страницы, содержащие важные данные (пароль, номер кредитной карты и CVC, ответы на секретные вопросы и т. п.) открываются через HTTPS (SSL).
3. Пароль скрыт астерисками на страницах: регистрация, «забыли пароль», «смена пароля».
4. Корректное отображение сообщений об ошибках.
5. Завершение сессии после разлогина.
6. Доступ к закрытым разделам сайта.
7. SQL-инъекции.
8. Cross-Site Scripting (XSS) уязвимости.
9. HTML-инъекции.
10. Cookie должны храниться в зашифрованном виде.
11. Роли пользователей и доступ к контенту.

Тестирование локализации и глобализации

Тестирование интернационализации/глобализации WEB приложения включает тестирование приложения для различных местоположений, форматов дат, чисел и валют. Тестирование локализации включает тестирование WEB приложения с локализованными строками, изображениями и рабочими процессами для определенного региона.

Что проверяем?

1. Дата и время. Например отображение времени, даты в соответствии с часовым поясом пользователя.
2. Смена языка и проверка перевода всех элементов WEB приложения исходя из выбранного языка.
3. Выбор номера телефона с разными кодами стран.

4. Определение местоположения пользователя и отображение соответствующего пермишена ГЕО.
5. Отображение соответствующих символов валюты.

Тестирование удобства использования

Тестирование удобства использования подразумевает проверку навигации, контента, другая информация для пользователя.

Что проверяем?

1. Отсутствие орфографических и грамматических ошибок, все страницы имеют корректные заголовки.
2. Выравнивание картинок, шрифтов, текстов.
3. Информативные ошибки, подсказки.
4. Подсказки существуют для всех полей.
5. Отступы между полями, колонками, рядами и сообщениями об ошибках.
6. Кнопки имеют стандартный размер, цвет.
7. На сайте нет битых ссылок и изображений.
8. Неактивные поля отображаются серым цветом.
9. Проверьте сайт при разных разрешениях экрана.
10. Скролл должен появляться только тогда, когда он требуется.
11. Отображение чекбоксов и радио-кнопок, кнопки должны быть доступны с клавиатуры, и пользователь должен быть в состоянии пользоваться сайтом, используя только клавиатуру.
12. Отображение выпадающих списков.
13. Длинный текст скрывается под многоточие.
14. Корректный выбор даты.
15. Наличие плейсхолдеров в полях.
16. Логотип ведет на главную страницу сайта.
17. Переходы и навигация между страницами и разделами меню.

Кросс-платформенное тестирование

Кросс-платформенное тестирование проводится, чтобы убедиться, что ваше приложение

совместимо с другими браузерами, различными оболочками, аппаратным обеспечением устройства.

Что проверяем?

1. Тестирование в различных браузерах (Firefox, Chrome, Safari — это минимальный набор): анимация, верстка, шрифты, уведомления и т.д.
2. Тестирование в различных версиях ОС: Windows, Mac, Linux.
3. Java Script код работает в разных браузерах.
4. Просмотр на мобильных устройствах.

Резюме

Мы ознакомились с универсальной шпаргалкой по тестированию WEB приложений. Не забывайте читать документацию и дополнять чек-лист проверками, характерными для вашего сервиса. А если остались вопросы — скорее пишите в телеграм-канал [@qa_chillout](#).