**Контрольный список для тестирования веб-приложений: примеры тестов для веб-сайта**

При тестировании веб-приложений следует учитывать указанный ниже шаблон. Приведенный ниже контрольный список практически применим для всех типов веб-приложений в зависимости от бизнес-требований.

Контрольный список тестирования веб-приложений состоит из:

* [Юзабилити-тестирование](https://www.guru99.com/complete-web-application-testing-checklist.html#1)
* [Функциональное тестирование](https://www.guru99.com/complete-web-application-testing-checklist.html#2)
* [Тестирование на совместимость](https://www.guru99.com/complete-web-application-testing-checklist.html#3)
* [Тестирование базы данных](https://www.guru99.com/complete-web-application-testing-checklist.html#4)
* [Тестирование безопасности](https://www.guru99.com/complete-web-application-testing-checklist.html#5)
* [Тестирование производительности](https://www.guru99.com/complete-web-application-testing-checklist.html#6)

Теперь давайте подробно рассмотрим каждый чек-лист:

**Юзабилити-тестирование**

**Что такое юзабилити-тестирование?**

* Юзабилити-тестирование - это не что иное, как проверка удобства использования.
* При юзабилити-тестировании поток приложения тестируется, чтобы новый пользователь мог легко понять приложение.
* В основном, системная навигация проверяется в Юзабилити-тестировании.

**Какова цель юзабилити-тестирования?**

Тест юзабилити устанавливает простоту использования и эффективность продукта с использованием стандартных практик тестирования юзабилити.

**Примеры тестов юзабилити**

* Содержание веб-страницы должно быть правильным, без орфографических или грамматических ошибок.
* Все шрифты должны быть одинаковыми в соответствии с требованиями.
* Весь текст должен быть правильно выровнен.
* Все сообщения об ошибках должны быть правильными, без орфографических или грамматических ошибок, а сообщение об ошибке должно совпадать с меткой поля.
* Текст всплывающей подсказки должен присутствовать в каждом поле.
* Все поля должны быть правильно выровнены.
* Между метками полей, столбцами, строками и сообщениями об ошибках должно быть достаточно места.
* Все кнопки должны быть стандартного формата и размера.
* Ссылка на главную должна быть на каждой странице.
* Отключенные поля должны быть выделены серым цветом.
* Проверьте неработающие ссылки и изображения.
* Подтверждающее сообщение должно отображаться для любого типа операции обновления и удаления.
* Проверить сайт на разных разрешениях (640 х 480, 600х800 и тд?)
* Убедитесь, что конечный пользователь может запустить систему без разочарований.
* Убедитесь, что вкладка должна работать правильно.
* Полоса прокрутки должна появляться только при необходимости.
* Если при отправке появляется сообщение об ошибке, там должна быть информация, введенная пользователем.
* Заголовок должен отображаться на каждой веб-странице
* Все поля (текстовое поле, раскрывающееся меню, переключатель и т. Д.) И кнопки должны быть доступны с помощью сочетаний клавиш, и пользователь должен иметь возможность выполнять все операции с помощью клавиатуры.
* Проверьте, не усекаются ли данные раскрывающегося списка из-за размера поля. Также проверьте, жестко ли запрограммированы данные или управляются ли они через администратора.

**Функциональное тестирование:**

**Что такое функциональное тестирование?**

* Тестирование функций и рабочих характеристик продукта на соответствие его спецификациям.
* Тестирование, которое игнорирует внутренний механизм системы или компонента и сосредотачивается исключительно на выходных данных, генерируемых в ответ на выбранные входные данные и условия выполнения.

**Какова цель или цель функционального тестирования?**

* Цель [функционального тестирования](https://www.guru99.com/functional-testing.html) - проверить, соответствует ли ваш продукт предполагаемым функциональным спецификациям, указанным в документации по разработке.

**Примеры сценариев функционального тестирования:**

* Проверьте, все обязательные поля должны быть проверены.
* Проверьте, что знак звездочки должен отображаться для всех обязательных полей.
* Проверьте, что система не должна отображать сообщение об ошибке для необязательных полей.
* Убедитесь, что високосные годы подтверждены правильно и не вызывают ошибок / просчетов.
* Проверьте, что числовые поля не должны принимать алфавиты и должно отображаться правильное сообщение об ошибке.
* Проверьте наличие отрицательных чисел, если это разрешено для числовых полей.
* Для расчетов необходимо правильно обрабатывать тестовое деление на ноль.
* Проверьте максимальную длину каждого поля, чтобы убедиться, что данные не усекаются.
* Проверьте, что всплывающее сообщение («Это поле ограничено 500 символами») должно отображаться, если данные достигают максимального размера поля.
* Проверьте, должно ли отображаться подтверждающее сообщение для операций обновления и удаления.
* Проверьте, какие значения суммы должны отображаться в денежном формате.
* Проверьте все поля ввода на наличие специальных символов.
* Проверьте функциональность тайм-аута.
* Протестируйте функцию сортировки.
* Проверьте работоспособность имеющихся кнопок
* Проверить Политика конфиденциальности и часто задаваемые вопросы четко определены и должны быть доступны для пользователей.
* Проверьте, не работает ли какая-либо функция, и пользователь перенаправляется на страницу пользовательской ошибки.
* Проверьте, правильно ли открываются все загруженные документы.
* Проверьте, должен ли пользователь иметь возможность скачивать загруженные файлы.
* Проверьте работу электронной почты в системе.
* Проверьте, правильно ли работает [Java-](https://www.guru99.com/java-tutorial.html) скрипт в разных браузерах (IE, Firefox, Chrome, safari и Opera).
* Проверьте, что произойдет, если пользователь удалит файлы cookie, находясь на сайте.
* Проверьте, что произойдет, если пользователь удалит файлы cookie после посещения сайта.
* Проверить все данные в поле со списком / списком расположены в хронологическом порядке.

**Проверка совместимости:**

**Что такое тестирование на совместимость?**

* Тестирование совместимости используется для определения совместимости вашего программного обеспечения с другими элементами системы, с которыми оно должно работать, например, с браузерами, операционными системами или оборудованием.

**Какова цель или цель тестирования совместимости?**

* Цель тестирования совместимости - оценить, насколько хорошо программное обеспечение работает в конкретном браузере, операционных системах, оборудовании или программном обеспечении.

**Примеры сценариев тестирования совместимости:**

* Протестируйте веб-сайт в разных браузерах (IE, Firefox, Chrome, Safari и Opera) и убедитесь, что веб-сайт отображается правильно.
* Проверьте, совместима ли используемая версия HTML с соответствующими версиями браузера.
* Проверьте правильность отображения изображений в разных браузерах.
* Проверьте, можно ли использовать шрифты в разных браузерах.
* Проверьте, можно ли использовать код java-скрипта в разных браузерах.
* Протестируйте анимированные GIF-файлы в разных браузерах.

**Тестирование базы данных:**

**Что такое тестирование базы данных?**

* При тестировании базы данных тестируются серверные записи, которые были вставлены через веб-приложения или настольные приложения. Данные, отображаемые в веб-приложении, должны совпадать с данными, хранящимися в базе данных.

**Чтобы выполнить тестирование базы данных, тестировщик должен знать следующие моменты** :

* Тестировщик должен хорошо понимать функциональные требования, бизнес-логику, поток приложений и структуру базы данных.
* Тестировщик должен определить таблицы, триггеры, процедуры хранения, представления и курсоры, используемые для приложения.
* Тестировщик должен понимать логику созданных триггеров, процедур хранения, представлений и курсоров.
* Тестировщик должен выяснить, какие таблицы будут затронуты при выполнении операций вставки, обновления и удаления (DML) через веб-приложения или настольные приложения.

**С помощью вышеупомянутых пунктов тестировщик может легко написать тестовые сценарии для тестирования базы данных.**

**Примеры тестовых случаев для тестирования базы данных:**

* Проверьте имя базы данных: имя базы данных должно соответствовать спецификациям.
* Проверьте таблицы, столбцы, типы столбцов и значения по умолчанию: все должно соответствовать спецификациям.
* Проверьте, допускает ли столбец значение NULL.
* Проверьте первичный и внешний ключ каждой таблицы.
* Проверьте сохраненную процедуру:
* Проверьте, установлена ​​ли хранимая процедура.
* Проверьте имя хранимой процедуры
* Проверьте имена параметров, типы и количество параметров.
* Проверьте параметры, требуются они или нет.
* Протестируйте хранимую процедуру, удалив некоторые параметры
* Проверьте, когда выход равен нулю, нулевые записи должны быть затронуты.
* Протестируйте хранимую процедуру, написав простые [SQL-](https://www.guru99.com/sql.html) запросы.
* Проверьте, возвращает ли хранимая процедура значения
* Протестируйте хранимую процедуру с образцом входных данных.
* Проверьте поведение каждого флага в таблице.
* Убедитесь, что данные правильно сохраняются в базе данных после каждой отправки страницы.
* Проверьте данные, если выполняются операции DML (обновление, удаление и вставка).
* Проверьте длину каждого поля: длина поля в задней и передней частях должна быть одинаковой.
* Проверьте имена базы данных QA, UAT и production. Имена должны быть уникальными.
* Проверьте зашифрованные данные в базе данных.
* Проверьте размер базы данных. Также проверьте время отклика каждого выполненного запроса.
* Проверьте данные, отображаемые в интерфейсе, и убедитесь, что они совпадают в серверной части.
* Проверьте достоверность данных, вставив недопустимые данные в базу данных.
* Проверьте триггеры.

**Что такое тестирование безопасности?**

Тестирование безопасности включает в себя тестирование для выявления любых недостатков и пробелов с точки зрения безопасности.

**Примеры тестовых сценариев для тестирования безопасности:**

* Убедитесь, что веб-страница, содержащая важные данные, такие как пароль, номера кредитных карт, секретные ответы на контрольный вопрос и т. Д., Должна быть отправлена ​​через HTTPS (SSL).
* Убедитесь, что важная информация, такая как пароль, номера кредитных карт и т. Д., Должна отображаться в зашифрованном виде.
* Правила проверки пароля применяются на всех страницах аутентификации, таких как Регистрация, забытый пароль, изменение пароля.
* Убедитесь, что при изменении пароля пользователь не сможет войти со старым паролем.
* Убедитесь, что сообщения об ошибках не должны отображать важную информацию.
* Убедитесь, что пользователь вышел из системы или сеанс пользователя истек, пользователь не должен иметь возможность перемещаться по сайту.
* Подтвердите доступ к защищенным и незащищенным веб-страницам напрямую без входа в систему.
* Убедитесь, что опция «Просмотр исходного кода» отключена и не должна быть видна пользователю.
* Убедитесь, что учетная запись пользователя заблокирована, если пользователь несколько раз вводит неправильный пароль.
* Убедитесь, что файлы cookie не должны хранить пароли.
* Убедитесь, что какие-либо функции не работают, система не должна отображать информацию о приложении, сервере или базе данных. Вместо этого он должен отображать настраиваемую страницу ошибки.
* Проверьте атаки с использованием SQL-инъекций.
* Проверьте роли пользователей и их права. Например, отправитель запроса не должен иметь доступ к странице администратора.
* Убедитесь, что важные операции записаны в файлы журналов и эту информацию можно отслеживать.
* Убедитесь, что значения сеанса находятся в зашифрованном формате в адресной строке.
* Убедитесь, что информация cookie хранится в зашифрованном формате.
* Проверить приложение на наличие атак грубой силы

**Что такое тестирование производительности?**

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

**Общие сценарии тестирования:**

* Для определения производительности, стабильности и масштабируемости приложения в различных условиях нагрузки.
* Чтобы определить, может ли текущая архитектура поддерживать приложение на пиковом пользовательском уровне.
* Чтобы определить, какой размер конфигурации обеспечивает наилучший уровень производительности.
* Выявление узких мест в приложениях и инфраструктуре.
* Чтобы определить, отрицательно ли повлияла новая версия программного обеспечения на время отклика.
* Чтобы оценить продукт и / или оборудование, чтобы определить, может ли оно выдержать прогнозируемые объемы нагрузки.

**Как провести тестирование производительности? Путем ручного тестирования или автоматизации**

Практически невозможно выполнить тестирование производительности вручную из-за некоторых недостатков, таких как:

* Потребуется большее количество ресурсов.
* Одновременные действия невозможны.
* Правильный системный мониторинг недоступен.
* Нелегко выполнять повторяющуюся задачу.

Следовательно, чтобы преодолеть вышеуказанные проблемы, мы должны использовать инструмент тестирования производительности. Ниже приведен список некоторых популярных инструментов тестирования.

* Apache JMeter
* Загрузите Runner
* Borland Silk Performer.
* Rational Performance Tester
* XXXV
* НЕО НАГРУЗКА