1. Проверка всех текстов на соответствие правилам языка.

Будь то русский или любой другой язык, каждый текст, выложенный на сайте, должен соответствовать его правилам: орфографическим, пунктуационным, должны быть строго соблюдены грамматические и лексические нормы. Это касается всего текстового контента – не только больших статей, но и коротких заголовков страниц, надписей на кнопках и пр.

2. Тестирование форм

Предполагается, что формы будут заполнять простые пользователи, поэтому в процессе заполнения все должно быть предельно понятно. Редакторам, тестирующим сайт, нужно проверить эти формы на себе и решить несколько вопросов.

Понятно ли простому пользователю, что нужно делать для правильного заполнения формы? Точно ли описаны этапы ее заполнения и требования к итоговому варианту?

С какими трудностями можно столкнуться при заполнении? Что может быть непонятно или истолковано двояко?

Как улучшить форму, как сделать ее более понятной и простой?

Принимается ли форма по указанному адресу после отправки?

3. Проверка фотографий

Каждая картинка на сайте должна быть оптимизирована. Также нужно проверить, сколько она весит – слишком большие неадаптированные для сети изображения будут сильно затруднять загрузку страниц. Также следует проверить подписи ко всем фотографиям, их описания, заголовки, альтернативный текст.

4. Проверка контекста на смысл

Здесь следует ответить на вопросы о смысле:

Зачем конкретная рассматриваемая страница может понадобиться пользователю?

Заполнена ли страница таким образом, чтобы пользователь мог сразу получить всю необходимую информацию по заявленной теме?

Насколько контент страницы повлияет на целевую аудиторию и повлияет ли вообще, привлечет ли?

Для этой работы привлекаются менеджеры, которые работают в роли тестировщиков.

5. Проверка скорости загрузки сайта

Размер страниц сайта не должен быть слишком большим, а скорость загрузки обязана быть максимальной. От того, насколько быстро загружается сайт, во многом зависит его посещаемость – никому не нравится ждать подолгу. Проверять скорость загрузки можно с помощью инструмента Google. Она относится к факторам ранжирования, поэтому обращайте внимание не рекомендации Google – поисковик сам подскажет, как можно улучшить ситуацию.

6. Адаптация сайта для просмотра на мобильных устройствах

Количество мобильного интернет-трафика все только растет, а это значит, что любой продвинутый современный сайт должен легко открываться на планшетах и смартфонах разных моделей. Кроме того, верстка сайта должна подходить под разные форматы – важно, чтобы пользователи могли прочитать всю информацию со своих мобильных устройств.

7.Проверка сайта в разных браузерах

Проверьте, как сайт открывается в популярных браузерах и открывается ли вообще.

8. Проверка шрифтов

На этом этапе идет простая проверка форматирования. Некоторые слова или символы могут выделяться в тексте из-за того, что заданное форматирование не соблюдается.

9. Навигация и юзабилити

Проверка удобства перемещения пользователя по сайту. Важно обратить внимание на то, действительно ли ссылки ведут на указанные места.

Веб-разработчики могут выступать в роли тестировщиков.

10. Поиск битых ссылок

Все они должны работать и вести на правильные страницы. Если сайт маленький, проверить ссылки можно самостоятельно. Если же на сайте около пятисот ссылок, можно вооружиться такими инструментами, как [Screaming Frog SEO Spider](https://www.screamingfrog.co.uk/seo-spider/). Если ссылок более пятисот, понадобиться купить расширенную версию или воспользоваться сервисом онлайн.

11. Проверка в Google Search Console

Search Console дает возможность узнать про все существенные проблемы сайта: появление страниц с ошибкой 404, появление несанкционированного ПО и пр. С помощью такого инструмента веб-разработчик может следить за ходом работы ресурса, а также регулировать индексацию, собирать статистику запросов и т.д. Еще один вариант для монторинга сайта – регистрация в Bing Webmaster Tools, Яндекс.вебмастер.

12. Установка Minify

Эта библиотека на PHP5 нужна, чтобы объединить, кэшировать JavaScript и CSS-файлы, а также минимизировать их. Благодаря этому уменьшается время загрузки сайта. Подробнее про библиотеку написано на GitHub.

13. Проверка ошибки 404

Страницы с такой ошибкой должны давать посетителю сайта полезную информацию и запускать поиск по ресурсу.

14. Проверка наличия иконки

Значок сайта, который пользователи видят в своей вкладке и адресной строке. Оригинальный запоминающийся значок повышает доверие к сайту и делает его более запоминающимся.

Часто SEO-оптимизаторы в роли тестировщиков.

15. Переадресация - 301 редирект

Структура сайта будет неизбежно меняться, и при этом важно удостовериться, что старые ссылки дают 301 редирект на новые. Для сбора всех ссылок предназначен уже упомянутый Screaming Frog Spider. Полученную по ссылкам информацию выгружают в  Excel, где в разных колонках отображаются старые и новые URL.

16. Проверка SEO

На каждой странице должен быть заполнен Title, кроме того, нужны метаописания, h1, h2. Проверка robots.txt.

17. Карта сайта

Должна быть корректно видна и в XML, и в HTML. Проверка наличия ссылки на нее в robots.txt.

18. Тестирование целей, установка аналитики

Установленная система аналитики должна быть настроена до того, как сайт будет запущен. Это может быть Google Analytics или другая по вашему усмотрению. Кроме всего, лучше сразу настроить все необходимые цели.

19. Проверка разметки

Любая разметка данных (например, Schema.org) должна адекватно отобразиться в SERP. За помощью в устранении багов микроразметки можно обратиться к Search Console или в Яндекс вебмастер.

20. Ускоренные мобильные страницы (AMP)

Страницы в этом формате есть не на всех сайтах, но если у вас они внедрены, корректность их отображения нужно проверить перед запуском.

21. Интеграция с социальными сетями

В зависимости от целевой аудитории, может быть внедрена интеграция с разными соцсетями. Перед запуском сайта проверьте все их кнопки, убедитесь, что они ведут на нужные страницы.

22. SERP

Search Console поможет проанализировать, как страницы показываются в результатах поиска, адекватно ли отображение.

23. Реклама

Рекламные кампании обычно запускаются одновременно с ресурсом.

24. Мониторинг

Для его осуществления предназначены различные инструменты, выбор зависит от того, с чем вы привыкли работать. Перед запуском сайта проверьте исправность своих инструментов.

25. Системы резервного копирования

При «падении» сервера эти системы должны сохранить все необходимые вам данные, поэтому такие системы должны работать с самого запуска сайта.

26. Тестирование нагрузки на ресурс

Тест сайта на «выдержку» - как он поведет себя во время наплыва трафика. Для такого тестирования нужны инструменты, которые могут разниться в зависимости от ваших предпочтений.

27. Авторизация пользователей

Если пользователю необходимо авторизоваться на сайте, чтобы просматривать определенные страницы, проверьте, насколько корректно работает авторизация. Отдельно проверьте, не могут ли неавторизованные пользователи просматривать защищенные страницы.

28. Сертификат безопасности SSL

Перейдя на страницу с HTTPS-адресом, кликните на замочек в адресной строке. После этого вы должны получить уведомление с именем вашего ресурса. При отсутствии замка или неправильном названии, сообщите об этих проблемах хостинг-провайдеру.

Несмотря на низкий порог вхождения, вёрстка является важным этапом в создании любого интернет-проекта.

Зачастую вёрстке уделяется незаслуженно мало внимания. Несмотря на то, что при приёме результатов работы верстальщика особых вопросов не возникает, и всех всё устраивает, это вовсе не исключает того, что вёрстка выполнена некачественно. Поэтому к проверке вёрстки, с которой наверняка придётся иметь дело другим верстальщикам, программистам, сеошникам, а, главное, пользователям, нужно подойти не менее основательно, чем к дизайну.

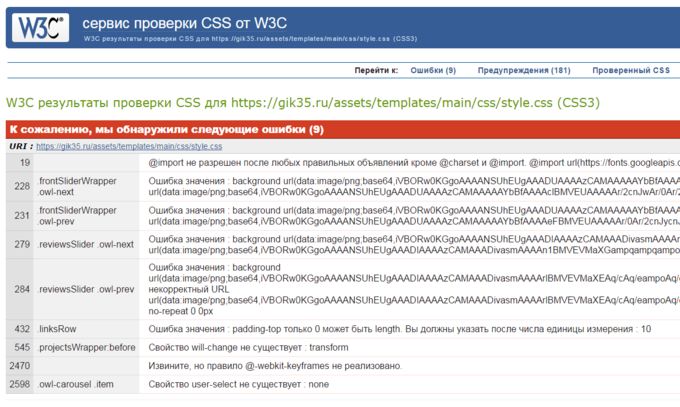
## 10 требований, которые предъявляются к тестированию верстки.

## 1 Проверка валидности

Следование стандартам гарантирует корректный разбор вёрстки в браузерах, следующих этим стандартам. И даже если то, как отрисовался сайт в вашем браузере, совпадает с макетом, то есть несколько нюансов, которые могут проявить себя в перспективе, ведь стандарты берутся не с потолка. К тому же, валидность вёрстки является одним из большого числа факторов ранжирования в поисковых системах.

Проверить валидность HTML-части: [http://validator.w3.org/nu/](https://quasi-art.ru/)

Проверить валидность CSS: [https://jigsaw.w3.org/css-validator/](https://quasi-art.ru/)



**2 Онлайн-сервисы**

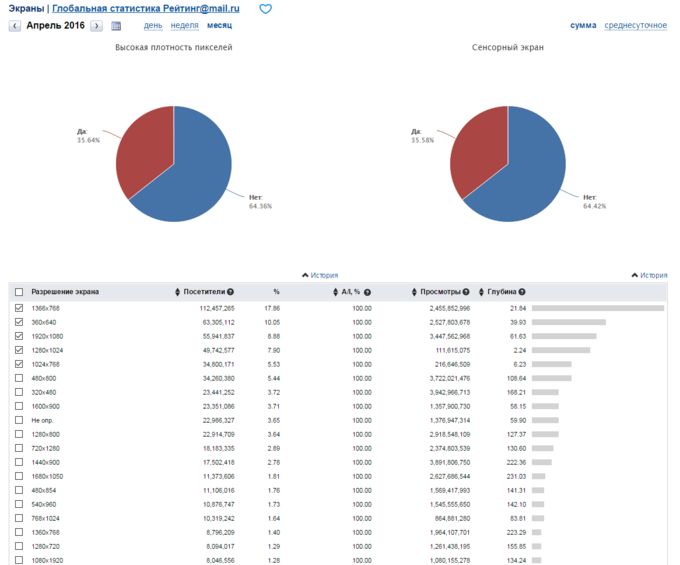
Кроме официальных валидаторов W3C, которые проверяют лишь соответствие стандартам, существуют также сервисы, которые произведут анализ CSS-файлов и дадут советы по их оптимизации и организации.

* [http://www.testmycss.com/](https://quasi-art.ru/)
* [http://csslint.net/](https://quasi-art.ru/)
* [https://www.onlinewebcheck.com/check.php](https://quasi-art.ru/)
* [http://www.cssanalyzer.com/](https://quasi-art.ru/)
* [http://www.quertime.com/article/top-20-online-tools-to-optimize-and-audit-css-codes/](https://quasi-art.ru/)

## 3 Реальные устройства

В настоящее время разнообразие разрешений экранов, их цветопередачи и плотности довольно внушительное, но возможности проверить сайт абсолютно на всех возможных устройствах и при всех комбинациях настроек очень сложно. Но альфа-тестирование вёрстки упрощается с возможностями Chrome WebDev Tools, где можно имитировать различные условия.

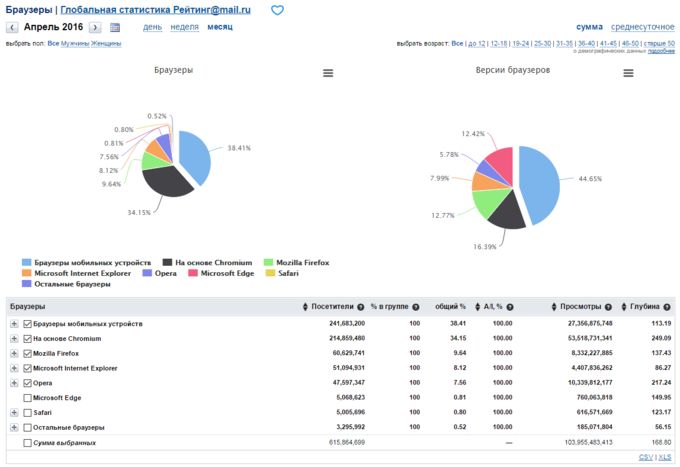
Лучше учитывать реальную статистику, которая есть в открытом доступе.



## 4 Зоопарк браузеров

Традиционно у верстальщиков установлены все популярные браузеры, каждый из которых имеет своеобразные особенности. Ситуация упрощается хотя бы тем, что большинство популярных браузеров работают на меньшем количестве движков, поэтому нет особой разницы между тестированием в Chromium и Яндекс.Браузере. Но вот Firefox и Safari требуют отдельного тестирования.

Статистика:



**5 Динамичный контент**

Зачастую при разработке дизайна сайта упускается важный момент — содержимое может быть различного объёма (разрешение изображения, количество символов текста и т.д.). Анонс новости может быть не только в 4 строки, как красиво нарисовал дизайнер, а в 3 или 5. А может и все 100, если редактор постарается. Изображение в слайдере, загруженное контент-менеджером, может вдруг оказаться не эталонного размера или не с теми пропорциями, как нарисовано в макете, а вообще повёрнутым на 90 градусов. Или вообще не существовать.

Именно такие моменты и должен предусмотреть верстальщик. Конечно, некоторые вещи обычно принято решать в том числе и на стороне сервера, например, генерация миниатюр и вывод изображения по умолчанию, но хорошим тоном считается предусматривать это и на уровне вёрстки.

Также верстальщик должен предусматривать такие ситуации, когда целые блоки сайта (шапка, подвал, новости, слайдер и пр.) могут скрываться или меняться между собой местами. В данном случае крайне желательно, чтобы, например, обмен местами двух блоков не привёл к каким-либо неожиданностям, например, значительное изменение вертикальных отступов между этими или соседними блоками.

**7 Выбор фреймворка**

Фреймворки служат для ускорения разработки. Самым популярным фреймворком является Bootstrap, который содержит множество готовых типовых решений (сетка, кнопки, типографика, формы, модальные окна и пр.).

Но у этого удобства есть обратная сторона — объём данных. Использование фреймворков может замедлить работу сайта, что особенно заметно на мобильных устройствах или при медленном соединении.

**8 Исходники**

Отлично если верстальщик использует препроцессоры и прочие средства автоматизации. Сопровождение проекта в таком случае становится проще.

Но если проект будет переходить из рук в руки, то стоит предупреждать последующих исполнителей об этом.

**9 Производительность**

Просмотр диафильма вместо плавно прокручиваемой страницы может испортить впечатление от сайта. Так как верстальщик обычно следует макету, то стоит с умом подойти к разработке дизайна и не увлекаться параллаксами, анимациями и прочими визуальными эффектами, которые могут произвести на посетителя не то впечатление, которое ожидается.

**10 Семантика**

Если верстальщик реализует всё на одних div-ах, то это плохой стиль. Интерактивные элементы вроде кнопок тоже стоит реализовывать с помощью тега button, а не div или span. В таком случае будет проще осуществлять навигацию посредством клавиатуры или программ для слепых или слабовидящих.