## Тестирование верстки

Ныне чтобы протестировать верстку, нужно учесть ряд деталей. А самое главное не забыть о них. Чтобы этого не допустить составили чек лист тестирования верстки. Само тестирование разбито на несколько под этапов.

**Этап 1. Визуальная часть.**

Тестируем в каком-то адекватном браузере (например, в Firefox).

* нет ли заметных глазу косяков: поломанные блоки, не состыковки цвета, некорректное отображение текста вокруг изображений;
* точность соответствия макета: делаем накладку в photoshop;
* проверяем сетку (вертикальные/горизонтальные выравнивания). Часто это упускается в формах. Если сетка кривая, но верстка соответствует дизайну, это вопрос можно обсудить с дизайнером/заказчиком;
* проверяем в разных разрешениях:

Список классических разрешений:

1024x600

1024x768

1152x864

1280x800

1280x1024

1440x900

1680x1050

1920x1080

- не должно ничего ломаться;

- не должна появляться горизонтальная прокрутка для оговоренных в ТЗ разрешениях;

- не должны резко обрываться фоны при больших разрешениях (что размера картинки хватило на большое разрешение).

Можно переводить экран в разные разрешения или можно использовать инструменты тестирования (например, в Firefox Меню -> Инструменты -> Веб-разработка -> Адаптивный дизайн или Resolution Test plugin в Chrome);

* уменьшаем размер окна меньше минимального по ТЗ - не должно ничего ломаться, фоны не должны "плыть";
* проверяем масштабирование страницы. В разумных масштабах (мы ограничиваемся диапазоном 75-150%) страница должна выглядеть без визуальных косяков;
* ресайз text area не должен ломать вёрстку;
* нормально ли подсвечиваются поля в фокусе;
* нормально ли подсвечиваются поля с ошибками.

**Этап 2. Доступность.**

Продолжаем в том же браузере.

* выделяется ли текст в текстовых блоках;
* кликаются ли кликабельные элементы (ссылки/кнопки);
* устанавливается ли фокус в поля форм;
* кликабельные элементы должны иметь указатель "курсор", перетаскивающиеся - "лапка" или "ресайз", активные/недоступные - курсор default;
* в идеале все активные элементы должны реагировать на наведение, не доступные/неактивные - не должны;
* кликабельные элементы, назначение которых не очевидно должны быть снабжены подсказками (title);
* лого на главной ссылкой быть не должно, на всех внутренних должно;
* проверка печати страницы (если было в ТЗ);
* Надписи (особенно логотип и главное меню сайта) должны оставаться читабельными, у всех информационных картинок должны быть подписи аккуратным небольшим серым шрифтом (да, для img можно задавать font – это внешний вид alt-текста, что выводится вместо картинки). Картинки часто отключают при использовании дорогого инета, тарифицируемого по траффику (GPRS, роуминг). Проверяется в FF через плагин Web Developer→Images→Replace Images With Alt Attributes.
* Работоспособность при выключенном JavaScript. JS может быть выключен согласно корпоративных требований безопастности. А в Opera Mini он работает только методом перезагрузки страницы. Весь критически важный функционал сайта должен быть доступен без JS. Проверяется в FF через плагин Web Developer→Disable→Disable JavaScript→All JavaScript;
* Должны быть favicon.ico

**Этап 3. Корректная работа при вбивании реального текста, надёжность вёрстки.**

Продолжаем в том же браузере.

Вёрстка должна тянутся, не разваливаться и не терять дизайнерский вид при изменении контента на странице. Его может быть больше или меньше чем на макете, он может быть обёрнут во всякие <p> из визига и т.п.  
Обязательно нужно проверять удаление заголовков! Бывает, что отступы между блоками после этого пропадают, это частая ошибка, причина — что отступы были заданы не для блоков, а для внутренних элементов — заголовков.

1. **Проверка ввода и удаления данных.**  
   Проверяется: на странице с контентом, пробуем добавлять и удалять содержимое – «что будет, когда текста много?», «а когда мало?».   
   Обязательно пробовать менять расположение элементов, чтоб после того как ты поменял блоки местами не развалилось оформление (из-за каскада).
2. **Проверка корректности работы стилей.**  
   Проверяется: на страницы с контентом вбиваем текст с абзацами и без абзацев (важно! бывает горе-верстальщики прописывают стили только для абзацев), со списками и картинками, таблицами и заголовками разных уровней.

Это нужно чтоб на живом сайте потом не полезли проблемы при заполнении реальными данными.

Правки содержимого страницы делаются в FF через плагин Firebug: HTML→Edit – меняем/добавляем/удаляем текст. Хороший пример проверочного текста находится в normalize.css в test.html между <body> и </body>.

**Этап 4. 404-е запросы.**

Нет ли в верстке 404-х запросов (firebug).

## Кросс-браузерность.

До этого этапа мы проверяли только в одном браузере на одной платформе. Т.к. перечень браузеров может быть достаточно большим, особенно с учетом разных платформ, проходить полный список в каждом из них, особого смысла нет (бюджет и сроки проекта). Достаточно ограничится такими поверками:

* визуальная часть;
* доступность;
* функциональность.

Точный список браузеров уточняется у менеджера или заказчика.

Для линейки IE дополнительно:

* масштаб страницы;
* расширяемость;
* рамки у элементов в фокусе;
* не должно быть Javascript ошибок (левый нижний угол).

Проверка в IE7 делается переключением IE8 в режим IE7 (F12→Режим обозревателя→Internet Explorer 7). Можно в IE Tester десктопной программе, но качество проверки и достоверности хуже, или в реальных браузерах на виртуальной машине.