# Обзорная статья по подбору тестовых устройств/браузеров

## Тестирование МП

Рано или поздно каждый мобильный тестировщик задаётся вопросом, на каком количестве устройств тестировать новый функционал, чтобы поймать максимальное количество девайсозависимых багов, потратив минимум времени. Казалось бы, для тестирования должно быть достаточно одного эталонного устройства. Однако, если тысячи человек пользуются вашим приложением каждый день, то вам придется задуматься о расширении «зоопарка» девайсов и лоб в лоб столкнуться с особенностями и проблемами, связанными с их многообразием. Так как же подобрать девайсы для тестирования?

### 1. Изучите целевую аудиторию

Часто этим пунктом пренебрегают. Но он может быть важен. Приложение элитного Барбершопа нацелит вас на новые модели смартфонов, флагманы с большим экраном; в женском салоне предположительно увеличится процент айфонов и уменьшится любовь к формату Plus. А если ваша целевая аудитория средний класс в регионах - тут будет большой разброс по производителям/устройствам, заметный процент старых моделей и Android в приоритете.

Особенно важно заметить про страны пользования — почти у всех стран есть свои “фишки”, так что аудитория приложения для условной Индии в среднем будет отличаться от пользователей из Скандинавии.

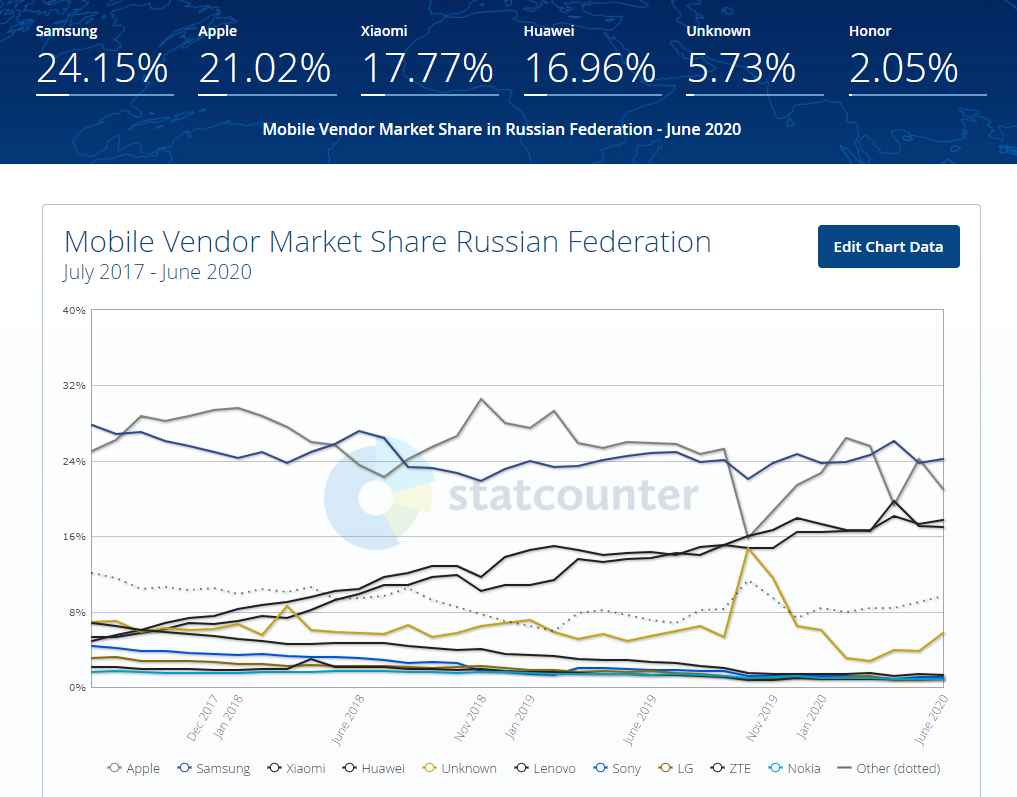
### 2. Внешняя аналитика

Тут нам на помощь могут прийти как международные подборки по самым популярным устройствам (такие, как, например [appbrain.com](https://www.appbrain.com/stats/top-android-phones-tablets-by-country?country=ru).), так и банальные варианты типа сортировки в яндекс.маркете по самым популярным устройствам. Главное — помнить про то, на какую страну и целевую аудиторию ориентирован ваш продукт, и основываться на этом условии.

### 3. Особенности самого приложения тоже могут влиять на выбор

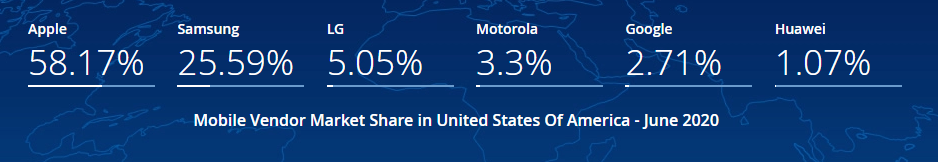
Приложение в принципе может не обслуживать девайсы ниже какой-то оси. Пообщайтесь с менеджером или разработчиками (как вариант редкий, но существующий – изучите документацию), чтобы потом не оказалось, что в приложении графическом вы не можете протестировать поведение Pencil 2, потому что купили девайсы только с первым. Или ваше мобильное приложение сильно зависит от железа, а вы этот момент не учли и у всех ваших девайсов схожие характеристики. Узнайте и выпишите отдельно требования.

### 4. Производители

На Android довольно высока девайсозависимость, производители стремятся привлечь покупателя фишками «только у нас», добавляя проблем разработчикам, поэтому важно протестировать приложения на устройствах разных **вендоров**. Особенно внимательно надо работать с теми, которые отключают по умолчанию сервисы Google Play, в том числе и Магазин приложений. Старайтесь держать в парке разные варианты, в том числе и “голый” андроид. Посмотреть лидирующих (по трафику) вендоров можно на [**Statcounter**](https://gs.statcounter.com/vendor-market-share/mobile/russian-federation).  
  
 

Вверху мы видим актуальные данные за предшествующий месяц по выбранному региону, возможны варианты мир/Европа/Страна. На скриншоте РФ - лидирует Samsung, а у соседей впереди уже Xiaomi.  


В США заметное отличие:

- большой отрыв у Apple, заметная доля Samsung, далее LG, Motorola, Google, Huawei.  
  


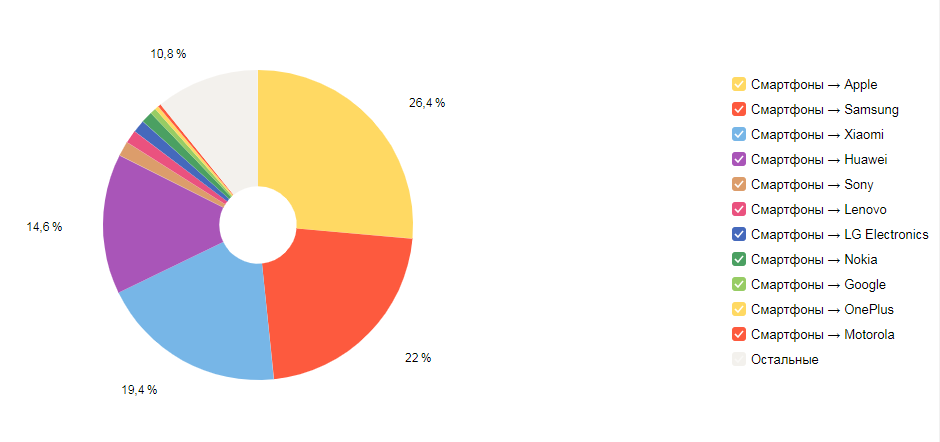
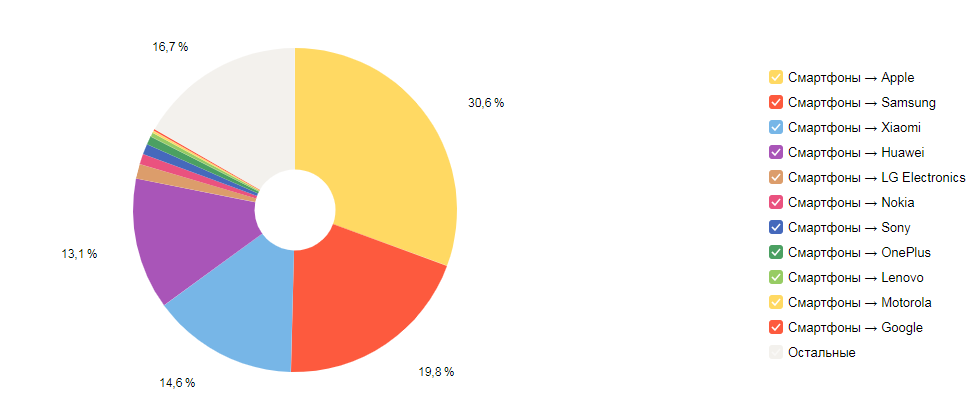
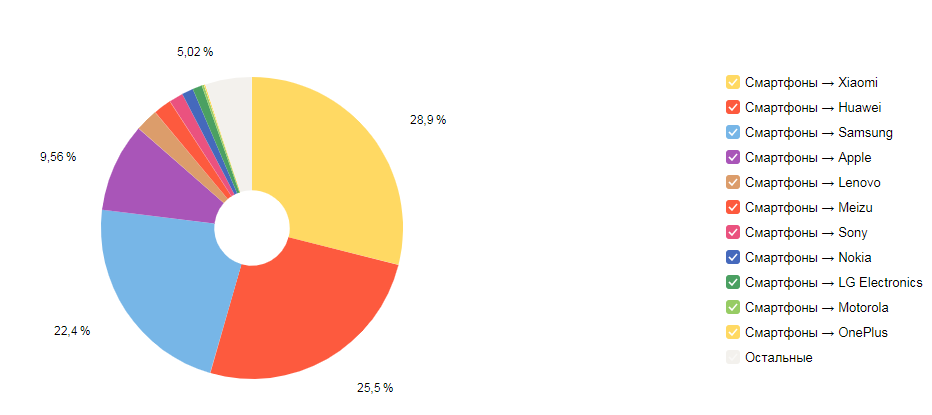
Очевидно, что геолокация приложения - важна.

Итак, мы (условно) выбираем девайсы для русскоязычного приложения на три страны: Россия (~2/3 аудитории), Украина, Беларусь.

- Samsung и Xiaomi включаем как обязательные.

- Huawei упорно не сдает позиции. Обсудите с командой, поддерживаете ли (если не в теме, погуглите «Huawei поддержка Google Play»). Сюда же идет и Honor – по сути это тот же Huawei с маркетингом на молодежную аудиторию.

В качестве наглядной иллюстрации три диаграммы (Яндекс.Метрика) по мобильному трафику за полгода на трех русскоязычных сайтах с разной целевой аудиторией.

Однозначно везде заметный охват у главной четверки:**Apple**, **Samsung**, **Xiaomi**, **Huawei**.  
  
  


При этом пятый вариант производителя везде столь незначителен, что в «среднем по палате», при ограниченном бюджете и человеко-часах на тестирование, его можно оставить на потом.

### **5. Соотношение сторон экрана**

Важный параметр, про который иногда забывают, привязываясь по старой памяти в первую очередь к разрешению экрана, где сейчас уже зашкаливающее многообразие (Android). При этом проверять надо обязательно. На GUI тут живет много багов, и не редкость, когда кнопка, расположившаяся «не там» может в итоге привести к криту, заблокировав возможность использовать функционал приложения.

При выборе параметров для смартфона постарайтесь захватить разнообразное соотношение сторон.

### 6. Размер

По размерам пытаемся подобрать на min, среднее и max.

### 7. Новизна

Разбивку по моделям/годам удобно смотреть на [DeviceSpecifications](https://www.devicespecifications.com/ru/brand/cefa26).

### **8. Особенности устройства**

На iOS могут быть нюансы работы нативной «Назад» у моделей с “челкой”, хотя физически она и расположена в зоне статус-бара. Любое приложение с ландшафтной ориентацией и полным использованием экрана (например плеер) также желательно посмотреть на моделях с “бровью”. Если приложение использует камеру, обязательно проверять и на фронтальной. Обратите внимание, чтобы в список попали устройства как с отсутствием на передней панели аппаратных кнопок так и с наличием. Для iOS это кнопка Home, для Android три сенсорных: Назад, Домой, Меню.

### 9. В список претендентов занесите эталонное устройство

Для разных проектов и задач, могут быть выбраны разные эталонные устройства.

Список эталонных устройств может быть оговорен с заказчиком, на основе его предпочтений. Если макеты по приложению предоставляет заказчик, то можно подобрать девайсы с разрешением, кратным ширине макетов.

Также мы можем выбрать девайс из совокупности параметров (популярность девайса для нашей целевой аудитории, самое выгодное соотношение сторон, подходящая ось для МП, для Android – чистый (голый) андроид).

Иногда в команде этот пункт может быть заменен на «любимый девайс разработчика», вполне возможно это будет большой гиковский OnePlus или Samsung на чистом Android.

В ходе приема рекламаций от заказчика можете обратить внимание то, какие устройства используются при приемке, и тоже включить их для своего тестирования.

### **10. Дополнительно**

Возьмите в привычку всегда держать на руках

* слабенький, старенький девайс с маленькой диагональю на старой оси;
* среднее устройство по этим показателям;
* новый, мощный, на последней оси, с большим дисплеем.

### 11. Как часто стоит пересматривать свой личный набор тестовых девайсов?

При затяжном проекте для избежания багов зависимых от девайсов - свой личный набор нужно менять раз в три месяца, а также при появлении каких то новых задач/фичей для которых требуется определенное устройство.

## Тестирование WEB

***Кросс-браузерное тестирование*** – это тестирование фичи в различных релевантных приложению браузерах. Это важная часть тестирования: несмотря на существование веб-стандартов, разные браузеры внедряют их различным образом. На глубоком уровне разные элементы страницы ведут себя по-разному в разных браузерах. Поведение фичи в Safari, к примеру, может сильно отличаться от ее работы в Chrome.

Если есть вероятность функциональных или косметических отличий в поведении системы в разных браузерах, без кросс-браузерного тестирования просто не обойтись. Неважно, новый это код или уже существующая фича: приложение должно единообразно работать в разных браузерах.

### **Выбор браузеров**

Конечно, тестировать абсолютно все существующие в мире браузеры – это утопия. Их чересчур много, чтобы это тестирование имело какой-то смысл. Разумнее сэкономить время, выбрав конкретные браузеры, подходящие для тестирования.

Условно всё кросс-тестирование делится на следующие направления:

* Браузеры: Chrome, Firefox, Safari, Edge, UC Browser, Opera и другие.
* Операционные системы: Windows, Android, iOS, Mac OS, Linux.
* Устройства: ПК (десктопы и ноутбуки), смартфоны, планшеты, телевизоры и другие.

Для эффективного тестирования требуется скомбинировать эти направления и составить список устройств, операционных систем и браузеров, в которых надо проводить тестирование.

Этот список должен содержать информацию о приоритете того или иного набора:

Например - высокоприоритетными могут быть связки «Ноутбук + Windows 10 + Chrome» или «iPhone 8 + iOS 13 + Safari», а низкоприоритетными, например, «PC + Linux + Firefox».

Если рассматривать кросс-браузерное тестирование со стороны работы приложения, то надо обратить внимание на две вещи:

1. Отображение страниц, то есть вёрстка (HTML + CSS).
2. Работоспособность различного интерактивного инструментария (JavaScript).

Так как правильно подбирать браузер?

### Браузеры и движки

За отрисовку HTML-документов и интерпретацию JavaScript-кода в браузере отвечают специальные компоненты так называемые движки браузера (browser engine). У современных браузеров обычно выделяется два: один отвечает за HTML и CSS, второй — за интерпретацию скриптов, написанных на JavaScript.

#### На сегодняшний день есть несколько актуальных движков:

1. **Gecko** - сегодня используется в Firefox и других продуктах, входящих в Mozilla, например, в почтовом клиенте Thunderbird. Соответствующий JS-движок — SpiderMonkey. Разные версии движка носят слегка различные наименования, так что мы можем встретить такие обозначения, как IonMonkey или OdinMonkey, но, по сути, это относится к одному и тому же продукту.
2. **Blink** - сейчас это наиболее популярный движок. Он основа браузера с открытым кодом Chromium, на базе которого, в свою очередь, разрабатывается браузер Chrome и многочисленные сторонние браузеры: Opera, «Яндекс.Браузер», Atom от «Майл.Ру» и другие. За основу Blink взят другой движок - WebKit, но сейчас они уже сильно отличаются. За JS-часть отвечает движок V8, который разработала компания Google.
3. **WebKit** - движок для браузера Safari - стандартный браузер в macOS. WebKit основан на другом движке KHTML, на котором построен браузер Konqueror в Linux/KDE. JavaScript в браузере Safari обрабатывается движком Nitro.
4. Не так давно в эту компанию входили движки от Internet Explorer:
   1. **Microsoft Trident** — используется в Internet Explorer 11 и 12. Вместе с ним работает JS-движок Chakra.
   2. **EdgeHTML** — основа для Microsoft Edge, браузера в Windows 10.

Но Trident уже давно не поддерживается самой компанией Microsoft, а EdgeHTML просуществовал очень недолго и сейчас заменён на Blink, то есть внутри Internet Explorer находится Google Chrome. С полным списком браузеров, которые основаны на Blink/Chromium, можно ознакомиться, например, в [«Википедии»](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80%D1%8B_%D0%BD%D0%B0_%D0%B1%D0%B0%D0%B7%D0%B5_Chromium).

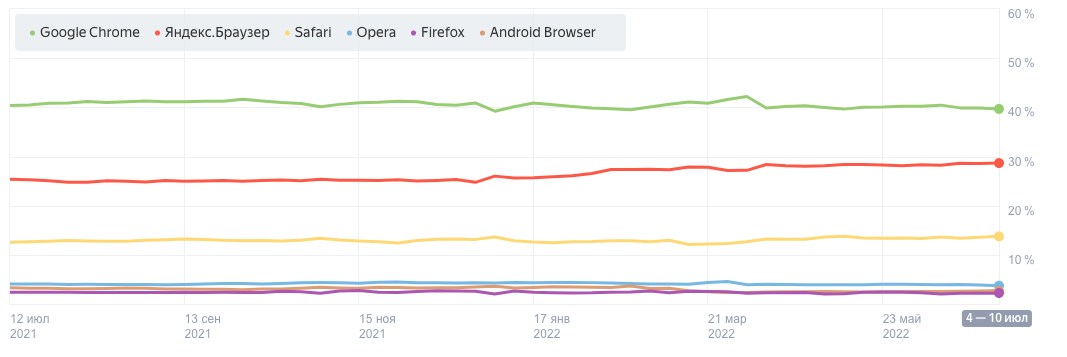
Таким образом, при кажущемся многообразии браузеров на рынке, существует всего 3 актуальных технологии рендеринга веб-страниц, на которых проверяются веб-приложения.

### Статистика использования браузеров

Для эффективного тестирования надо посмотреть, какими устройствами и браузерами пользуются посетители сайта, реальные или предполагаемые. На скриншотах представлена статистика использования браузеров по данным [Liveinternet.ru](https://www.liveinternet.ru/stat/ru/browsers.html)



Еще одним полезным сервисом для анализа популярности браузеров в русскоязычном интернете считается [«Яндекс.Радар».](https://radar.yandex.ru/browsers)



### Операционные системы

#### ОС на ПК и браузеры

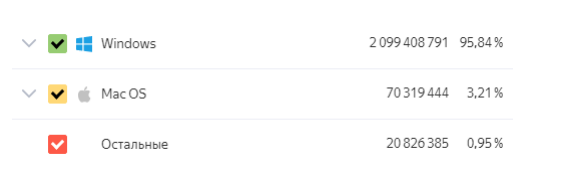
Сейчас широко распространены три операционные системы для ПК:

1. Windows.

2. macOS.

3. Linux.

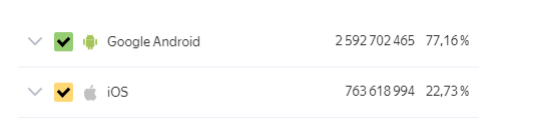
Подавляющее большинство компьютеров работает на Windows, чуть меньшее количество — хотя оно постоянно растёт — на macOS, а вот Linux используется редко. Зачастую это не персональные, а рабочие ПК, выход в интернет с которых ограничен. По данным Яндекса, с Windows заходит 95,84% всех пользователей ПК, с macOS — 3,21%, и с остальных операционных систем (все Linux) — порядка 1%.



Стоит отметить, что только Windows и macOS обладают уникальными браузерами: для Windows это Internet Explorer и Edge, для macOS — Safari. Для Linux нет широко используемых уникальных браузеров, на подавляющем большинстве Linux-систем установлены Chrome и/или Firefox.

#### Мобильные ОС и браузеры

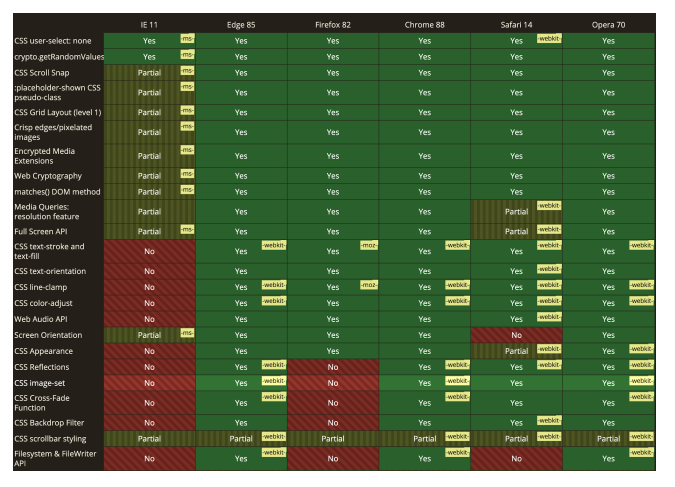
На сегодняшний день есть две популярные мобильные ОС — Android и iOS.



И, соответственно, два браузера под них — Mobile Chrome и Mobile Safari. Учитывая, что оба мобильных браузера входят в топ самых используемых браузеров, надо тестировать сайты под мобильные устройства. Для полноценного мобильного тестирования требуются как минимум четыре комплекта: Android и iOS, смартфон и планшет. От версии операционной системы зависит конкретная версия браузера, так как старые версии могут иметь устаревшие браузеры и плохую производительность.

### Цель и необходимость кросс-браузерной проверки:

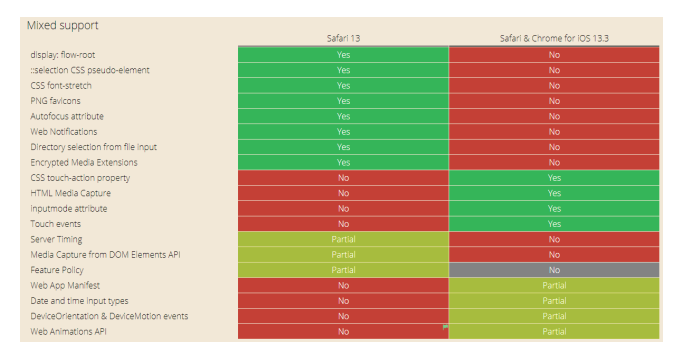
Хотя составляющие веб-страницы (HTML, CSS, JS) стандартизированы, итоговый результат отображения зависит от браузера. Браузер может не поддерживать часть стандартов, например, устаревшие модификаторы, обрабатывать стили особым образом, интерпретировать JavaScript со своими особенностями. На скриншоте ниже видно, что довольно большой пласт свойств поддерживается браузерами совершенно по-разному:



Полная таблица - на сайте [Caniuse.com](https://caniuse.com/?compare=firefox+74,chrome+83,ios_saf+13.3,and_chr+79&compareCats=all). Этот сайт позволяет выбрать браузеры и области - CSS, HTML5, JS и так далее - для сравнения и по выбранным критериям строит таблицу.

Кроссплатформенное тестирование важно потому, что, например, мобильный Safari и Safari под Mac, Chrome под Windows и Chrome под Android - это разные сборки браузера. Соответственно, между ними есть различия.

Этот же сайт даёт такую таблицу функций, которые поддерживаются по-разному, для обычного и мобильного Safari:



### Надо ли тестировать все комбинации браузеров, ОС и разрешений экрана?

Общий ответ: нет, не надо.

Важно исходить из имеющихся ресурсов и разумной достаточности. Например, если у вас нет 12-дюймового iPad Pro, а есть только обычный iPad Air, то, конечно, этого планшета будет достаточно для проверки вашего веб-приложения. Тестировать в разрешениях 1280x1024, 1366x768, 1920x1080 и так далее на каждом браузере также излишне. В этом случае эффективен комбинаторный подход к составлению тестового окружения.

### Надо ли тестировать на старых версиях и на самых-самых новых ОС и браузеров?

На самых-самых новых — надо, но во вторую очередь, когда всё проверено на основных связках ОС + браузер. На старых тестировать стоит только в том случае, если ими по какой-то причине пользуется значительное количество ваших клиентов. Это могут быть корпоративные устройства, встроенные устройства и так далее.