

Адаптивный дизайн

Зачем все это нужно?

Начнём с истории. Если в 1980 экраны наиболее распространенных устройств обладали дисплеями с разрешением в 320×200 пикселей, то к концу 1990 наиболее распространённым было разрешение экрана в 1024×768 пикселей. И всем всего хватало. Более того, применение термина "наиболее распространенные" трудно применять, когда по факту было всего 3 - 4 доступных размера экранов. Проблема в том, что прогресс не стоит на месте, а 1990 год был 30 лет назад. Да, до сих пор можно найти разрешение экрана, которое можно считать стандартом для большинства (1920×1080), но само количество потенциальных вариантов возросло. Не говоря уже про то, что помимо некогда стандартного соотношения сторон 4:3, появились 5:4, 16:9, 2,37:1 (3440×1440) ко всему этому добавилось море устройств, каждое со своим соотношением сторон и разрешением. Немаловажным фактором было и увеличение интернет трафика с мобильных устройств, так как в значительной мере возросла необходимость вывода существующих сайтов на новые размеры экранов.

Так как же в условиях всего этого хаоса убедиться, что пользователь увидит именно тот интерфейс, который вы проектировали, а не первозданное зло?

Нам на помощь приходит адаптив.

Под адаптивным дизайном принято понимать процесс, направленный на оптимизацию работоспособности и внешнего вида интерфейса для широкого спектра устройств и экранов.

При изменении размеров экранов необходимо перестраивать не только структуру страницы, но и контент, а порой и функционал. Помимо удобства, вменяемо настроенный адаптив гарантирует более выгодные позиции при выдаче результатов в запросах поисковиков, что делает данный процесс неотъемлемой частью работы над проектом.

В конечном счете то, какими методами вы будете руководствоваться в подготовке адаптива для системы отвечают детали проекта.

Так какой же он бывает?

1. **Статичный.** Игнорируем возможные изменения размера экрана. Делаем все только в одном размере с фиксированными значениями отступов. Данный подход позволяет в значительной степени сэкономить время, силы и прочие ресурсы в ходе разработки в обмен на полное отсутствие возможности хоть как-то подстраиваться под разные устройства, если места на экране не хватает, то увы, не судьба и все такое. Как результат, подобный подход применим в ситуациях, когда речь идет о поддержке устройств с одним единственным размером экрана или о приложениях, которые не меняют размер своего окна и гарантированно поместятся на экране. (Калькулятор, как пример. Возможность изменять размер экрана

калькулятора никак не повлияет на работоспособность приложения в позитивном ключе, получается, что если мы знаем минимальный допустимый размер для приложения, то стоит его придерживаться именно его.) Ну и в редких случаях, когда вам уже все равно и вы смирились с отчаянием. В таких случаях можно сделать интерфейс в единственном размере, а все ситуации, которые взрослые люди решают при помощи адаптива, решать скроллом.

2. **Адаптивный** (привязка к контейнерам фиксированного размера с перестроением дизайна и функционала) Определяем интересующие нас размеры и настраиваем точки излома таким образом, чтобы обеспечить максимально адекватное отображение интерфейса на выбранных разрешениях экрана. По факту в ходе работы выбираются размеры экранов, на которых продукт должен адекватно смотреться. На них и делается упор при разработке. Имея данные о наиболее распространенных устройствах и их параметрах, мы можем сформулировать достаточно вменяемое предположение о том, под какие размеры необходимо подготовить интерфейс для того, чтобы покрыть большую часть существующих экранов. По сути, при подобном подходе готовится несколько вариантов одного и того же экрана по одной для каждой точки излома. При изменении размера экрана, превосходящем диапазон для конкретной точки, получаем переход на следующую. Все значения в промежутках дополняются свободным местом по бокам.
3. **Отзывчивый** (Подстраивание под любую ширину экрана с упором на плавное изменение внешнего вида в зависимости от содержимого) Выставляем пропорции, размеры в процентах. Задачей такого подхода является предоставление максимально равномерного опыта взаимодействия независимо от размера экрана. Можно поспорить, что размер экрана тут даже не имеет значения, так как интерфейс будет подстраиваться под любой размер экрана и любые пропорции. В таком варианте немного сложнее гарантировать, что все будет работать, как задумано. Сложность заключается в том, что вместо общего вида интерфейса здесь приходится думать о правилах поведения для тех или иных элементов, максимальных и минимальных значениях и о том, как обеспечить адекватное отображение для любого размера экрана, без каких-то четких критериев и значений размеров. Суть не в том, чтобы при растягивании интерфейса получалась красивая анимация, суть в том, чтобы избежать необходимости учитывать и прорабатывать все частные случаи и размеры экранов, а просто оптимизировать поведение экранов для любых условий. Дополнительным бонусом получаем тот факт, что подготовленный по данной стратегии продукт будет адекватно отображаться как на уже существующих экранах, так и на тех, которые

еще не появились. Другими словами, меньше риск необходимости обновления в будущем.

4. **Гибридный.** При данном подходе мы совмещаем как плюсы, так и минусы от двух предыдущих вариантов. Для части ключевых экранов делается адаптивный дизайн, а для определенного диапазона задается отзывчивый дизайн. В результате мы не только гарантируем оптимальный вид системы на ключевых экранах, но и заботимся о частных ситуациях. Чаще всего это реализуется в виде адаптива для десктопа и отзывчивого дизайна для мобильных устройств, так как там чуть больше хаоса с размерами. Такой подход занимает больше времени, так как совмещает в себе трудозатраты от двух подходов, но позволяет добиться лучшего результата на всех или по крайней мере большинстве экранов. А не в этом ли суть всего процесса?

Mobile first. Что это такое и с чем его едят?

Данный подход, как это ни странно, очень похож на золотое сечение. Чего же у них общего? Все просто. Все любят говорить, что применяют его, но на деле не могут описать, в чем он заключается и дать хоть в какой-то мере адекватную формулировку. На деле подход сводится к тому, что работа над интерфейсом начинается с проектирования опыта взаимодействия пользователя с мобильным устройством, а потом уже расширяется для более крупных экранов. На словах звучит просто, если забыть про то, что зачастую при классическом подходе с уменьшением размера экрана мы отрезаем и выбрасываем все то, что не помещается. При обратном же процессе мы начинаем с проектирования максимально адекватного мобильного интерфейса с учетом всех ограничений, а потом расширяем его. Это как разница между лепкой и скульптурой. Стоит отметить, что трудно считать какой-то из этих подходов лучшим, так как у каждого есть свое назначение, плюсы и минусы. Принимать решение о применении того или иного подхода стоит, опираясь на контекст в рамках проекта. Возникновение MF можно объяснить тем, что между экранами десктопных систем и систем мобильных устройств есть сильный перепад по размеру и доступным ресурсам, а практически полное отсутствие пользователей с планшетами делает этот разрыв еще более критичным. Получаем две противоположные позиции, между которыми приходится выбирать.

Направляющие и сетка

В зависимости от того, чего мы пытаемся добиться, контент в пределах страницы можно располагать с привязкой по сетке или же направляющим.

Сетка делит страницу на ряд колонок одинаковой или почти одинаковой ширины с фиксированными отступами между ними и гибкими либо фиксированными отступами по краям страницы. Разное количество колонок позволяет добиваться разного набора групп элементов на экране. Так, к примеру классические 12

колонок позволяют разместить 1, 2, 3, 4, 6 и 12 равных элементов и настроить их вменяемый адаптив для всех экранов. Основное преимущество данного метода кроет в себе и его основную проблему. Одинаковые отступы и четкая структура сильно ограничивают визуальное разнообразие. Для наиболее приятного внешнего вида важно понимать, где стоит отойти от заданной системы колонок и прибегнуть к простому набору отступов внутри компонентов.

Направляющие. Основным плюсом данного метода можно считать свободу в реализации структуры. С другой стороны, без должного внимания очень легко получить менее чем удовлетворительное качество результата. Отсутствие четкой привязки к колонкам дает возможность более гибко подойти к реализации интерфейса, при условии, что у вас есть понимание того, что вообще стоит делать и где отход от четкой структуры колонок будет преимуществом для проекта в целом.

Плитка. Часть систем пробуют реализовать интерфейс через набор ячеек, каждая с фиксированными отступами внутри, которые подобно кирпичикам складываются в целостный экран. Данный подход позволяет подстраивать общий вид под огромное количество пользователей, тогда, когда мы не можем гарантировать, какие блоки надо демонстрировать в какой ситуации и невозможно предугадать, что будет входить в состав экрана. получаем максимальный адаптив в том плане, что подстраивается уже не ширина или контент, а сама структура и функционал страницы. Независимо от наполнения получаем рабочий интерфейс.

Что имеем в итоге?

На сегодняшний день при работе с интерфейсами вопрос заключается уже не в том, делаем мы адаптив или нет, а в том, каким подходом руководствуемся? Необходимо знать разные методы и понимать разницу между ними для того, чтобы сделать обоснованный выбор в пользу того или иного решения.