

ИДЕАЛЬНО!

*как создать
и переделать*

СВОЙ САЙТ

*Правильный подход
и передовые
техники разработки*

PUSHBOOKS
КНИГИ КОТОРЫЕ ДОЧИТЫВАЮТ

Книга
02

Элиот Джей Стокс
Ли Веру
Рэйчел Эндрю
Дмитрий Фадеев
Арэл Балкан
Кристиан Хейлманн
Пол Боуг
Марк Эдвардс
Аарон Уолтер
Бен Шварц
Энди Кларк
Стивен Хей
Дэвид Стори

**Идеально! Как создать и переделать
свой сайт. Правильный подход и
передовые техники разработки**

Текст предоставлен правообладателем

http://www.litres.ru/pages/biblio_book/?art=8721903

*Идеально! Как создать и переделать свой сайт. Правильный подход и передовые техники
разработки: СилаУма – Паблишер; Москва; 2013*

ISBN 978-5-906084-03-3

Аннотация

Книга — мостик между управленцем, маркетологом, веб-аналитиком и веб-разработчиком, одинаково полезная и интересная всем, вне зависимости от уровня квалификации и опыта.

Книга, прочитав которую, вы сможете понять современный подход к интернет-технологиям!

Закончив читать эту книгу, вы пересмотрите свои взгляды на работу, код и методы ведения проектов! Все покажется вам гораздо проще, гармоничнее и интереснее.

На русском языке издается впервые.

Содержание

Предисловие от партнера российского издания	7
Вступительное слово научного редактора	8
Введение	9
Бизнес-подход в переделывании сайтов	10
Пришло ли время переделывать сайт?	11
Когда полная переработка сайта является расточительством	11
Хорошие дизайнеры создают новое, великие дизайнеры переделывают старое	11
Когда сделать заново предпочтительней, чем улучшать постепенно	12
Сигналы к масштабным изменениям	13
Причины для изменений	14
Как избежать ловушек в проекте	15
Борьба с расширением рамок проекта	15
Принесение в жертву стиля и тренда	16
Разработка сайта без учета возврата инвестиций	16
Что делать с негативными отзывами	17
К исследованиям будь готов!	19
Зачем вам делать домашнее задание?	19
Какое исследование я должен провести?	19
Ценность опроса заинтересованных лиц	20
Исследуем то, что имеем в настоящий момент	21
Принцип совместной работы при переделывании сайта	22
Как правильно работать над проектом	22
Презентация проекта	23
Обратная связь через тестирование	25
Продажа тестирования заказчику	25
Какой вид тестирования проводить	26
Как получить отзывы о дизайне	26
Юзабилити-тестирование	27
Когда и кого тестировать	27
Как быть с отзывами заказчика	28
Вечная переделка	30
Как создать программу непрерывной работы	30
Использование технологий для обеспечения будущего сайту	30
Планирование мобильного будущего сайта	31
Что дальше?	33
Об авторе	34
О рецензенте	35
Выбор платформы: технические решения для переделки вашего сайта	36
Для кого этот раздел?	37
Узнать все о действующей платформе	38
Сбор технических требований	39

Управление контентом	39
Всем ли редакторам (контент-менеджерам) нужны одинаковые права доступа?	40
Будут ли управлять контентом специальные сотрудники, или это станет частью работы других людей?	40
Нужно ли поддерживать многоязычность?	40
Какой тип контента будет редактироваться?	41
А/В-тестирование	41
Электронная коммерция	42
Платежный шлюз	44
PCI DSS	46
О доставке	47
Цифровые продукты	47
Отчет и другие функции	48
Интеграция с другими системами	48
Ограничения	50
Бюджет	50
Выбор языка программирования	50
Технические ограничения	51
Политические ограничения	51
Писать новый код или улучшать старый?	52
Собственная разработка или коробочное ПО?	53
Зачем разрабатывать свое?	53
Выбираем коробочную платформу?	53
Облачные решения	54
Бойтесь «евангелистов»!	55
Хостинг	56
Дешевый хостинг – дорогой хостинг, или Скупой платит дважды	56
Как выбрать хост	56
Принимаем решение	58
Типы хостинга	59
Среда разработки	61
Вспомогательный сервер	61
Контроль версий	62
Перенос вашего сайта	63
Установка нового оборудования хостинга	63
Перемещение файлов на новый сервер	63
Перенос базы данных на новый сервер	63
Настройка почтовых ящиков	63
Изменение сервера доменных имен, или переназначение записи домена	63
Перенос сайта, работающего на базе данных	64
Куда дальше?	65
Об авторе	66
О рецензентах	67
Погружение в HTML5	68
Мы откуда и куда?	70

WHATWG, W3C и «Компания»	70
Найдите, какими функциями пользоваться	71
Конец ознакомительного фрагмента.	72

Идеально! Как создать и переделать свой сайт. Правильный подход и передовые техники разработки

Издано с разрешения Smashing Media GmbH.

Под редакцией Александра Сарычева

© The Smashing Book #3 – Redesign the Web, 2012

© Smashing Media GmbH, 2012

© О. Серовская, перевод, 2013

© Издательство «СилаУма – Паблишер», 2013

* * *

Предисловие от партнера российского издания

Множество авторов сейчас пытаются научить вас создавать сайты, приносящие прибыль своим владельцам. Однако книги, написанные настоящими мастерами своего дела, встречаются редко. Перед вами одна из них.

Эта книга достойна самого внимательного изучения. Здесь есть все инструменты для того, чтобы вы смогли создавать по-настоящему современные, удобные и эффективно работающие сайты.

Не важно, профессионал ли вы в деле сайтостроения или решили сделать первые свои страницы, в этом бесценном труде вы найдете ответы на многие вопросы.

Каждый раздел написан ведущим специалистом в своей отрасли.

- Первый научит вас выстраивать грамотный процесс работы с заказчиками. Этот блок будет полезен не только «технарям», но и всем, кто по роду службы сталкивается с заказчиками напрямую.

- Второй в подробностях описывает процесс создания или переделки сайта от выбора платформы и языка программирования до определения хостинга, интеграции платежных систем и управления контентом.

- Из третьего вы узнаете, как виртуозно управляться с версткой, JavaScript и современными языками программирования. После прочтения HTML5 и CSS3 станут вашими надежными союзниками.

- Четвертый посвящен преобразованию сайта, смене стилей и кодов. Здесь уделено внимание и веб-типографике, и грамотной разметке страниц.

- Пятый погрузит вас в мир Java-возможностей. Отныне перемещения, трансформации и анимация с JavaScript перестанут быть для вас загадкой.

- В последних разделах внимание уделяется повышению юзабилити, дизайнерским фишкам и приданию сайту индивидуальности. Не поверите, но тут вам в помощь привлекается даже психология. Вы узнаете, как вызвать нужные эмоции у пользователей и запрограммировать их поведение.

И все это обильно снабжено примерами, иллюстрациями и html-кодами, которые вы сможете сразу использовать в своей работе.

Эта книга способна полностью перевернуть ваш подход к работе над созданием сайтов, сделать его более продуктивным и денежным.

Напоследок хочется сказать о том, что вы – везунчик. Вы имеете отношение к одной из самых интересных и динамично развивающихся сфер – разработке и веб-дизайну. Но чтобы получать от своей работы не только удовольствие, но и материальное вознаграждение вкупе с вечной благодарностью от заказчиков, вы должны знать, как этого добиться. И мы уверены, что в этом вам поможет эта книга.

Удачи!

*Команда iAGE,
Технологии вашего Интернет-будущего
(iage.net)*

Вступительное слово научного редактора

Вы подумываете переделать свой сайт? Или вы специалист, который занимается их созданием? В любом случае всем вам будет полезно взять верный курс на разработку современного сайта.

Каждый год издаётся большое количество книг, скорее даже справочников, для технических специалистов. Есть книги и по маркетингу, юзабилити, проектированию целевых страниц (landingpages), оптимизации рекламы. Но вот куда копнуть глубже и где почитать про подход к разработке современного сайта? С чего начать, как выстроить процесс работы, какими технологиями воспользоваться и что не забыть по дороге?

Книга укажет направление для работы. Прочитав книгу, вы сможете разобраться, как добавить сайту «человеческое лицо», узнаете фишки юзабилити, поймёте как сделать отзывчивый (responsive) дизайн, а также убедитесь, что HTML5 и CSS3 уже давно в бою. Из книги вы узнаете и о перспективном подходе к вёрстке сайтов – Flexbox. Я уверен – за ним будущее.

Книга посеет и зерно здорового скептицизма – а по правильному ли подходу вы (или ваши подрядчики) разрабатываете сайт? Строите ли работу от контента, делаете ли прототипы, или до сих пор мыслите статичными макетами?

Мы в «ЛидМашине» уже частично использовали отдельные вещи, о которых рассказывается в книге, в своей работе и проектах наших клиентов. Теперь мы поглядим на весь процесс сверху и, вероятно, скорректируем процесс. Взгляните и вы.

Александр Сарычев,

маркетёр-аналитик

лаборатории

интернет-маркетинга

«ЛидМашина» (LeadMachine.ru)

Введение

Автор: Элиот Джей Стокс

Когда вы работаете в стремительно развивающейся индустрии, то каждый миг этого развития очень волнующий. Замечу, что сказать: «Самое захватывающее сейчас – это работа в веб-дизайне и разработке» – будет несколько неверно. Работа и жизнь на острие новых интернет-технологий всегда были захватывающими и всегда таковыми будут.

И все же в последние месяцы я ловлю себя на том, что говорю эти слова, потому что сейчас действительно захватывающий момент. Во времена широкого использования табличной верстки и инноваций новых медиа, мы были словно беспокойные дети, которые ищут поддержки у своих родителей из мира аналоговых медиа; в начале этого столетия мы превратились в неуклюжих подростков, которые экспериментируют с различными сервисами и техниками, как в лучшую сторону (веб-стандарты), так и в худшую (плагины). Сейчас мы выросли примерно до двадцатилетних и преобразовали все, чему научились, в новый, захватывающий опыт, который, в конце концов, окончательно охватит изменяемую, мягкую, открытую и изменчивую натуру Интернета.

Мало просто иметь продвинутые сервисы, равно как и мало уметь их использовать. Настоящая зрелость – это глубокое понимание того, что может и должно быть достигнуто.

Нет ничего революционного во множественной поддержке браузеров для экспериментальных CSS-характеристик. Не новость, что есть возможность использовать тысячи различных гарнитур. Обычное дело – вставка медиазапросов для изменения стилей в соответствии с шириной браузера. Но применение этих инструментов придает вашей работе изящество.

Эта концепция зрелой индустрии, более того – нашей собственной компетентности, в ней раскрылась на конференциях по всему миру, на страницах печатных журналов, дневников и книг и, конечно же, в блогах. Создалось ощущение взросления Интернета, что не так давно побудило меня переделать сайт Smashing Magazine. С чем мы столкнулись при переделке, вы узнаете на страницах этой книги.

Тема переделывания сайта иногда принимает здесь форму делового обсуждения, ведь это тоже часть процесса, подчеркивает один из авторов, Пол Боуг. Также обсуждается модернизация веб-сайта при помощи таких сервисов, как HTML5, CSS3 and JavaScript, о чем написали Бен Шварц, Ли Веру, Дэвид Стори и Кристиан Хейлманн. Представлена тема и в рассказе про адаптивные мобильные техники (авторы Стивен Хей и Энди Кларк). И, конечно, она раскрывается в широком обсуждении, представленном в разделах, авторы которых Рэйчел Эндрю, Дмитрий Фадеев, Марк Эдвардс, Арэл Балкан и Аарон Уолтер.

Желание ускорить ход событий, которые запустили механизм переделывания Smashing Magazine, также повлияло на создание и контент книги. Эта страсть, которая связывает нас больше как единомышленников, а не конкурентов, – страсть, которую, я уверен, вы прочувствуете, когда будете читать эти страницы, – вдохновит каждого из нас сделать Интернет лучше с помощью... задания градиентов через CSS, использования веб-шрифтов, использования медиазапросов.

Бизнес-подход в переделывании сайтов

Автор: Пол Боуг

Рецензент: Коллис Тайед

Переделывание сайтов – это та область, в которой самые смелые разработчики и дизайнеры могут позволить разгуляться своей фантазии. Пустой документ Photoshop, библиотека фрагментов кода CSS3 – для них нет ничего более захватывающего, чем погружение в мир новых возможностей. Однако, как это ни печально, я здесь для того, чтобы разрушить их воздушные замки.

В последующих разделах представлены клевые дизайнерские техники, еще более крутые возможности для кодинга, а множество прекрасных наглядных примеров дизайна сайтов помогут разработчикам не заблудиться в лабиринте новых инструментов.

Я же затрону тему бизнес-подхода в реконструкции сайтов.

Почему я остановился на такой скучной теме, спросите вы? Ответ прост. Суть бизнес-подхода в дизайне состоит не только в том, чтобы создать удачный сайт, но и время от времени полностью переделывать его, несмотря на кажущуюся безупречность проекта. Я, как и все вы, принимал участие в разработке множества сайтов, и меня буквально швыряло по волнам изменяющихся технических заданий, нечетко сформулированных инструкций и разногласий с заказчиками. К счастью, годы опыта научили меня тому, что если ты достаточно подготовлен, можно избежать этих ловушек.

Цель этого раздела – вооружить вас знаниями и познакомить с техниками разработки для того, чтобы ваша последующая работа над реконструкцией сайта не превратилась в ночной кошмар. Рассмотрим тему по следующим пунктам:

- Подводные камни при переделывании сайта.
- Тщательное исследование проекта.
- Работа с клиентами.
- Тестирование дизайна.
- Жизнь сайта после реконструкции.

Итак, до того как вы воодушевитесь этим списком, давайте выясним, когда же пора кардинально переделывать сайт.

Пришло ли время переделывать сайт?

Как только начальник или клиент просят нас переделать сайт, мы готовы ринуться в бой. По-другому и быть не может! Это же то, что мы обожаем делать! К сожалению, глобальные перемены не всегда являются лучшим решением, и мы как эксперты по разработке обязаны объяснить почему.

С момента своего рождения Интернет переживал ряд периодических изменений. Также и каждые несколько лет кто-то из высшего руководства приходит в ужас при виде своего сайта и требует его сменить. Прежний сайт выбрасывается, а его место занимает новый.

Какое-то время этот сайт сияет на просторах Интернета. Однако контент не обновляется, технологии движутся вперед, а вкусы и пристрастия меняются. Некогда новый блестящий сайт постепенно блекнет до тех пор, пока компания не начинает стыдиться его.

Через пару лет кто-то из руководителей вновь видит необходимость сделать что-то с сайтом, и процесс начинается сначала.

Когда полная переработка сайта является расточительством

В следующий раз, когда клиент попросит вас полностью переделать сайт, будьте готовы вступить с ним в спор и объяснить, почему эта просьба может оказаться ошибкой.

Причины могут быть следующими:

- В процессе глобальной реконструкции затрагиваются все составные части сайта, даже те, которые все еще работают безупречно. И сайт компонуется заново.
- Большую часть своей жизни сайт работает неэффективно из-за того, что контент не обновляется и постепенно устаревает дизайн. И хотя раз в несколько лет сайт радует вас своей новизной, в остальное время он рассматривается, скорее, как помеха и поэтому работает не в полную силу.
- Пользователи редко положительно реагируют на большие изменения. Вам достаточно лишь посмотреть, какой шумный протест вызывает у них обновление Facebook, чтобы осознать эту проблему¹.
- Периодическая глобальная реконструкция сайта невыгодно сказывается на расходе денежных средств, так как это требует солидных инвестиций каждые несколько лет.
- Тестирование эффективности комплексной переделки сайта – процесс сложный, так как изменяется очень многое.
- Затяжное отсутствие обновлений дает пользователям мало оснований для возвращения на ваш сайт.

Во многих случаях я советую своим клиентам отказаться от глобальной реконструкции в пользу постепенной перестройки сайта.

Хорошие дизайнеры создают новое, великие дизайнеры переделывают старое

В 2005 году Кэмерон Молл впервые популяризовал идею о предпочтении постепенной перестройки сайта, а не его полной переработке. Концепция существенно развилась с того момента, но ей все еще легко дать определение: перестройка или перегруппировка сайта –

¹ [Facebook Changes Confuse Users, as a Major Overhaul Looms // The Washington Post, \[smashed.by/fbc\]\(https://www.smashed.by/fbc\)](https://www.washingtonpost.com/archive/local/2014/05/22/facebook-changes-confuse-users-as-a-major-overhaul-looms/2014/05/22/)

это серия поэтапных изменений, производимых со временем для того, чтобы решать отдельные бизнес-задачи.

«Перегруппировка сайта – это серия поэтапных изменений, проводимых со временем для того, чтобы выполнять характерные бизнес-технические задачи».

Кэмерон Молл

Другими словами, сторонники постепенного развития сайта отвергают идею его серьезной реконструкции каждые несколько лет, ради того чтобы он оставался актуальным. Вместо этого они предлагают программу непрерывной пошаговой разработки, которая увеличит эффективность сайта при выполнении бизнес-задач. Это поможет избежать подводных камней, возникающих при дорогостоящей модернизации каждые несколько лет.

Это мнение противоречит мысли, что сайт на самом деле может быть когда-либо готов. Суть подхода в том, что дизайн необходимо развивать на основании постоянного тестирования. Как только больше узнаешь о поведении и предпочтениях пользователей, сразу же эти новые знания нужно использовать в дизайне сайта. Это позволит ему заработать еще более эффективно. Следует сказать, что метод перестройки подходит не каждому проекту.

Когда сделать заново предпочтительней, чем улучшать постепенно

Несмотря на то что я фанат поэтапного усовершенствования сайтов (т. е. перегруппировки), чаще всего в моей работе, которой я занимаюсь в своем агентстве Headscape, все-таки приходится полностью переделывать их. Мы стараемся, где только возможно, использовать текущее состояние сайта как отправную точку, но часто его создание происходит с нуля, без опоры на предыдущий опыт. Все это потому, что при создании сайтов в редких случаях разработчики думали об их долгосрочном развитии. Жизни сайта после его создания придавали мало значения. Позже в этом разделе мы обсудим данную проблему.

К тому же перегруппировка сама по себе может послужить причиной для крупной реконструкции. Даже тот сайт, в план развития которого была заложена возможность поэтапных обновлений, время от времени нуждается в полном переделывании. На то есть две причины. Во-первых, во многих постоянно изменяющихся сайтах будут накапливаться нестыковки, которые постепенно разрушат интерфейс.

«В конце концов, стратегия поэтапных изменений так или иначе, разрушает связанность, требуя новой архитектуры для пользовательского интерфейса»

В своей статье, написанной в поддержку поэтапных изменений², Якоб Нильсон говорит:

Другими словами, любой сайт со временем нужно будет переделывать полностью.

Вторая проблема, связанная с поэтапным изменением, это его влияние на базовый код. Как-то в нашем агентстве мы более шести лет работали с одним из клиентов. Это время было потрачено на непрерывный процесс улучшения и поэтапного изменения сайта. Добавлялись новые функции, в то время как другие выбрасывались. Дизайн менялся на основании обратной связи с заказчиком. Все эти изменения в конце концов превратились в кошмар. Одна часть сайта была написана на классической ASP³, другая часть на .NET⁴. CSS-файлы⁵ пухли

² Nielsen, Jacob. Fresh vs. Familiar: How Aggressively to Redesign, smashed.by/nielsen

³ Active Server Pages – технология создания веб-приложений корпорации «Майкрософт». – Примеч. ред.

⁴ .NET Framework – программная платформа. Основой платформы является исполняющая среда CLR, способная выпол-

от строк кода, в которых больше не было необходимости. Мы планировали и документировали так хорошо, как только могли, но в итоге стало понятно, что весь базовый код придется переписать.

Не удивительно, что клиент отказывался платить значительную сумму, не видя никаких визуальных отличий. Тогда мы совместили эту работу с изменением дизайна сайта и таким образом убили двух зайцев одним выстрелом.

Несмотря на то что процесс поэтапных изменений предпочтительнее, периодическое полное переделывание сайта все еще имеет смысл.

Суть в том, чтобы распознать признаки, которые дадут нам знать, когда требуется полная переделка.

Сигналы к масштабным изменениям

Мы уже определили два сигнала, которые указывают на то, что сайт лучше переделать заново, чем перегруппировать: если в интерфейс пользователя вкрадывается несовместимость и если код становится неуправляемым.

Вопросы кода обычно связаны с производительностью. Если сайт страдает от серьезных проблем с производительностью, а поэтапных изменений недостаточно, чтобы исправить их, тогда полная реконструкция могла бы стать решением. Создание целого сайта с нуля дает вам возможность оптимизировать производительность, удалив унаследованный программный код.

Полная переделка сайта может быть важной еще просто потому, что существующий дизайн исчерпал свои возможности. И хотя, как правило, еще есть возможность совершать поэтапные изменения, но некоторые аспекты все же ужасно сложно обновлять, не навредив остальному дизайну.

Например, изменение базовой модульной структуры сайта повлияет на все, начиная от навигации и заканчивая размером шрифта. Поэтому если нужно радикально изменить модульную сетку из-за нового несовместимого контента или изменений в разрешении экрана, то создание нового сайта может стать необходимым.

Также есть бизнес-причины, по которым считается, что переделать заново сайт будет лучше, чем поэтапно усовершенствовать его. Значительное переделывание несет в себе возможности продвижения, которые вы не получите от поэтапных изменений. И еще, большая переделка сайта дает радикальную встряску, которая иногда так необходима для того, чтобы старый сайт зажил новой жизнью.

И, наконец, есть еще одна простая причина, по которой я продвигаю идею о полном переделывании сайта. Это когда организация основательно поменяла свое позиционирование. В этом случае недостаточно будет слегка переделать логотип. Наоборот, если компания серьезно меняет свое позиционирование, то это повлияет на все, начиная от целевой аудитории и заканчивая контентом и внешним видом сайта.

Я часто мог наблюдать, что взять и лишь вклеить новый логотип в сайт бывает недостаточно. Бренд – это нечто большее, чем просто логотип. Итак, неважно, выбираете вы глобальную переделку или постепенную перегруппировку сайта, но делать это вы должны, определив причины.

нять как обычные программы, так и серверные веб-приложения. – Примеч. ред.

⁵ Cascading Style Sheets – каскадные таблицы стилей – формальный язык описания внешнего вида документа, написанного с использованием языка разметки (например, HTML). – Примеч. ред.

Причины для изменений

Как отмечает в своей статье⁶ Кэмерон Молл, переделывать сайт просто из-за того, что он внешне «постарел», недостаточно. Изменения должны приводиться в действие бизнес-задачами.

Типичные причины для изменений следующие:

- Изменение тенденций рынка.
- Изменение бизнес-модели.
- Падение конверсии.
- Рост требований о поддержке сайта со стороны клиентов.
- Смена позиционирования бренда.

Замечу, что с вашим бизнесом связаны не только эти задачи (что важнее, чем просто эстетика), но и выгода, которую вы получите от незамедлительных действий. Другими словами, если показатель конверсии вашего сайта падает или ваши бизнес-предложения изменились, то вы не захотите ждать два года до следующей существенной реконструкции сайта, чтобы воспользоваться им.

Вот поэтому часто предпочтительней делать поэтапные изменения. Все меняется быстро как в онлайн, так и в офлайн. Если вы хотите, чтобы ваш сайт работал с максимальной эффективностью, то не можете ждать, когда полностью переделаете его.

В зависимости от актуальности проблемы изменения сайта могут быть незначительными, такими как доработка текстов, или существенными, такими как полное переделывание интерфейса пользователя. Как бы то ни было, думаете ли вы об изменениях или уже пришло время изменить ваш сайт, все решения должны подкрепляться бизнес-задачами и быть частью действующего и поэтапного процесса развития.

К сожалению, изменение сайта сопряжено с определенными опасностями.

⁶ Moll, Cameron. Good Designers Redesign, Great Designers Realign, smashed.by/realign

Как избежать ловушек в проекте

Когда мы начинаем работу над новым проектом, перед нами открывается бесконечное множество возможностей. Мы возбуждены, полны энтузиазма и идей. Почему же тогда когда все подходит к концу, мы просто хотим запустить проект и никогда больше о нем не вспоминать?

Без разницы, полностью ли мы переделываем сайт или занимаемся умеренной перегруппировкой, для излишне неосторожных дизайнеров не будет недостатка в ловушках. Хуже всего, что нам свойственно повторять одни и те же ошибки. Перед тем как окунуться в новый проект, какое-то время обязательно обдумайте те общие вопросы, которые могут возникнуть, и то, как вы будете их решать.

Хотя любой проект индивидуален, ниже я представил самые большие проблемы, с которыми я столкнулся за 16 лет моей работы веб-дизайнером:

- Расширение рамок проекта.
- Принесение в жертву стиля и трендов.
- Создание сайта без учета того, будет ли возврат инвестиций.
- Негативная обратная связь.

Давайте рассмотрим каждую проблему по отдельности.

Борьба с расширением рамок проекта

Ничто так не бросает в дрожь любого дизайнера, чем фраза: «У меня есть идея!» – услышанная из уст заказчика.

Рамки каждого проекта почти неизбежно будут расширяться. Позицию заказчиков можно понять. Они не являются веб-экспертами, в отличие от нас, и поэтому не думают обо всем заранее. Только работая бок о бок с нами, они начинают реализовывать возможности.

Как же тогда поступить с изменением технического задания? Есть один выход – упереться рогами и сказать: «Нет». Но это может привести к конфронтации и разрушить ваши отношения с клиентом.

Методы, которые мы нашли для достижения успеха, – это составление списка пожеланий и идей, а также работа по этапам. Когда у заказчика или у нас возникнет идея, она будет вноситься в список пожеланий. Эти идеи не должны подвергаться цензуре или оценке, они просто должны добавляться.

К концу очередного этапа перечень пожеланий анализируется. Пункты тщательно выверяются, тем, которые остаются, отдается предпочтение, и они вносятся в следующую стадию разработки проекта.

Как только клиент ясно осознает этот процесс, у него создается определенное видение и формируются ожидания того, как будут осуществляться изменения. Клиенту также нравится быть все время «в процессе», что хорошо сказывается на повторных сделках.

Заказчик обязательно будет настаивать на том, чтобы некоторые его идеи включались в проект. Советую вам избегать споров по этому поводу на промежуточном этапе разработки. Сумейте объяснить заказчику, что реализация идей на данной стадии может навредить проекту. Предложите ему заморозить их до запуска следующего этапа, и обсудить потом.

Заморозка спорных вопросов до второй стадии имеет три преимущества:

- Заказчик гораздо реже навязывает идеи, как только видит свой восхитительный новый сайт.

- Создание нового сайта не будет тормозиться деталями, что вероятнее всего позволит запустить его вовремя.

- С новым живым сайтом ваши позиции в споре о том, что именно должно быть в его рамках, усиливаются.

Но не только обсуждение рамок проекта вселяет ужас в наши сердца.

Принесение в жертву стиля и тренда

Еще одно восклицание, от которого веб-дизайнеры бледнеют, это что-то вроде: «Мой сын подсел на Facebook. На нашем сайте должен быть Facebook». Не вопрос, Facebook – это не проблема. Но заказчик запросто может сказать: «У наших конкурентов крутой сайт. Мы хотим такой же» или «Нам нужен веб-два-нольный сайт».

Фишка в том, что заказчики часто кидаются с головой в то, что модно, престижно и успешно, и ждут, что мы будем делать это вместе с ними. Я не хочу сказать, что мы круче и лучше. Как веб-дизайнеры, мы любим новейшие тренды, будь то адаптивный дизайн, градиенты или красивые тени.

Но проблема в том, что модные тенденции будут меняться по запросам заказчика. То, что так нравится ему на момент начала работы над проектом, может стать ненавистным к моменту утверждения дизайна. Даже если клиент остается стойким в своих взглядах, через какое-то время сайт станет выглядеть несовременным, а это усложнит процесс внесения изменений и, возможно, приведет к тому, что клиент увидит вашу работу не в лучшем свете.

Самая мощная защита в борьбе с модой – это задать простой вопрос: «Зачем?» Ответить, что это «круто» или «в тренде», будет недостаточно. Нам нужно спросить, почему определенная идея хороша, и привести реальные деловые доводы. Мы должны увязать их с нашим выбором и выбором заказчика.

Если заказчик становится жертвой моды, не пытайтесь переубедить его. Наоборот, спокойно спросите его, почему он считает какую-либо идею хорошей. Зачастую мягкого подталкивания будет достаточно для того, чтобы заказчик понял, что он в сетях соблазна новизны.

Если это не даст желаемых результатов, действуйте дальше и задайте ему вопрос, откуда он ожидает возврата инвестиций.

Разработка сайта без учета возврата инвестиций

Большинство требований в заявках, которые я получал много лет, выглядели, скорее, как перечень пожеланий, чем исчерпывающий бриф. Я считал своим долгом отшлифовать каждое предложение так, чтобы оно было выгодно для бизнеса заказчика. И делал я это охотнее, чем в точности исполнял все, о чем он просил. Заказчики (как и все мы) соблазняются функциями и возможностями. Они не принимают во внимание стоимость внедрения в сравнении с возвратом инвестиций, потому что они некомпетентны в этом вопросе. Помогать тут им или нет, зависит от вас.

Помню, когда наше дизайнерское агентство только начинало раскручиваться, мы создавали сайты с многоязычной поддержкой. А так как в те годы многоязычная поддержка еще не являлась стандартной функцией, то и стоила она дорого. Я не задавался вопросом, для чего она нужна заказчику и уж тем более как будет переводиться их контент. В результате функция никогда не использовалась. В основном я внедрял их пожелания, а не удовлетворял нужды их бизнеса. Соответственно, я тратил кучу денег заказчика.

Чем больше функций вводится, тем больше сложностей и выше цена. Ваша задача – помочь заказчику придерживаться простых вещей.

Вы, возможно, уже заметили, что здесь красной нитью проходит одна тема: всегда придерживайтесь простых решений. Ваша задача как веб-разработчика состоит в том, чтобы воздерживаться от излишеств как своих, так и заказчика. Вместо того чтобы глобально переделывать сайт, остановитесь на искусной перегруппировке. Вместо того чтобы добавлять больше функций, определите набор простых функциональных возможностей и придерживайтесь его. Вместо слепого поклонения новейшим трендам сосредоточьтесь на простом, вечном и классическом.

Это гарантирует заказчику высокий показатель возврата средств, которые он вложит в сайт, работая с вами. А также сведет до минимума опасность последней ловушки при переделывании любых проектов: негативной обратной связи.

Что делать с негативными отзывами

Я уже говорил о том, что реакция людей на изменения редко когда бывает положительной. Некоторые из нас просто не любят изменений, а другие разочаровываются в них, потому что все изменилось не так, как они ожидали. В любом случае все изменения на сайте, вызовут реакцию и у пользователей, и у заинтересованных в бизнесе лиц.

Мы установили, что минимальные изменения уменьшат шансы на критику. Но это не всегда возможно, а даже если и получится, вы все равно получите негативную обратную связь. Не следует доверять первым откликам, будь они положительные или отрицательные.

Я вспоминаю, как мы обсуждали этот вопрос с Даниелем Буркой, когда он был ведущим дизайнером в Digg. Он рассказывал мне, как сложно ему было не реагировать на критику сразу же, после того как выходил в свет новый элемент его проекта. Он предпочитал тут же решать кажущуюся проблему. Но потом он научился тому, что если подождать пару недель, пользователи привыкают к изменениям и принимают их. В конце концов, это стало его стандартным методом. Он не делал дальнейших изменений до тех пор, пока элемент проекта не проживет хотя бы двух недель.

Когда заказчик или пользователь видят новую разработку, они дают поспешную оценку, которая редко отражает их конечное восприятие.

Кого-то новый дизайн может сразить наповал, и только со временем обнаружится, что сайтом совершенно неудобно пользоваться. Так же кто-то может с первого взгляда возненавидеть изменения, а потом проникнуться к ним любовью.

Я овладел тремя простыми тактиками эффективного управления негативной обратной связью. Во-первых, в графике разработки проекта нужно оставлять промежуток между первым показом дизайна заказчику и внесением изменений в него. Это дает клиенту возможность привыкнуть к дизайну, перед тем как дать какой-либо отзыв.

Во-вторых, активно побуждайте клиента использовать это время для того, чтобы «переварить» дизайн. Объясните ему доходчиво, что первая реакция – не всегда самая правильная. Пусть заказчик вернется к дизайну через какое-то время, прежде чем он выскажет свое мнение.

И, наконец, заранее предупредите заказчика о том, что может произойти с новым дизайном, когда он выйдет в свет. Скажите о том, что пользователи могут выразить негативную реакцию. Критика обычно пугает заказчика. Ясно, что это может вызвать дрожь в его коленках и ухудшить положение вещей.

Возможно, заказчики испытают такое же напряжение, когда пропустят вашу разработку через себя. Возможно, что они тут же будут реагировать на каждый отрицательный комментарий, за исключением тех заинтересованных в проекте лиц, у которых не хватает времени на то, чтобы «впитывать» дизайн.

Предотвратите эту проблему, заранее обсудив ее с заказчиком. Вы сможете вернуться к этому моменту потом, если возникнет проблема. Заказчик убедится, что это обычное явление и вы к нему готовы.

Подготовка – это ключ к успеху. Это – путь к лучшему сайту для заказчика и наиболее стоящему проекту для вас. Но чтобы быть подготовленным на все сто, вы должны выполнить домашнее задание.

К исследованиям будь готов!

Когда сроки и бюджет ограничены, есть огромный соблазн сразу же погрузиться в процесс разработки дизайна. Однако делать это весьма неразумно. Во-первых, нам нужно четко понимать, что мы делаем и зачем мы это делаем. А для этого нужно провести некоторые исследования. Прежде чем объяснить, что я подразумеваю под этим, я хочу вкратце обрисовать, почему исследование так важно для процесса.

Зачем вам делать домашнее задание?

Есть две причины, по которым нужно проводить исследования до начала разработки любого проекта. Во-первых, это даст вам необходимые знания о проекте. Во-вторых (и, вероятно, что наиболее важно), вы будете во всеоружии, когда настанет время утвердить вашу работу.

Если мы имеем твердое представление о таких вещах, как бизнес-задачи, конкуренция, статистика и слабые стороны существующего проекта, тогда и объяснять наши дизайнерские решения, имеющие отношения к заказчику, будет значительно проще.

Доказывать, что пустое пространство вокруг контактной формы эстетически привлекательно, это не то, во что заказчик может вникнуть. Но если вы скажете ему, что свободное пространство поможет в решении бизнес-задачи, связанной с заполнением этой формы, — они вас поймут.

Итак, в чем же суть исследования?

Какое исследование я должен провести?

Количество проводимых вами исследований должно быть пропорционально ценности проекта. Размер неважен, вы должны провести хоть какие-то изыскания. Меня часто удивляет отсутствие базовой информации в обычной заявке. В требованиях часто не хватает таких фундаментальных вопросов, как:

- Для чего нам сайт?
- Чего мы ждем от сайта?
- Как мы сможем измерить его эффективность?
- Что будет делать пользователь на нашем сайте?

Как веб-разработчики, мы обязаны вытянуть эту информацию из клиента, до того как начнем работать над проектом. Бизнес-задачи должны встать во главе всего списка.

Заставить заказчика четко сформулировать свои бизнес-задачи может оказаться проблемой. Я часто задавал вопрос заказчикам, для чего им сайт и какие цели они преследуют. Внятного ответа я не получал.

Я уже не думаю о том, почему так было. Вместо этого я сажусь вместе с клиентом на старте проекта и методом мозгового штурма мы определяем с ним бизнес-задачи. Затем мы расставляем приоритеты по списку и определяем для них измеримые критерии успеха.

Например, такая расплывчатая бизнес-задача, как «увеличение продаж», должна превратиться во что-то более конкретное и напрямую связанное с призывом к действию. Так, «увеличение продаж» должно стать «увеличением числа горячих лидов через контактную форму сайта».

Расстановка бизнес-задач по приоритетам важна, потому что иногда они будут сталкиваться. Например, решение одной бизнес-задачи может принести клиенту больше лидов, в

то время как решение другой задачи с более высоким приоритетом может демонстрировать пользователю продукцию. Поэтому, если кто-то предлагает заставить пользователя представить свой e-mail раньше, чем показать товар лицом, в противовес вы можете сказать, что демонстрация продукции важнее.

Постановка конкретных и умеренных бизнес-задач приведет к минимальному количеству исследований, проводимых в каждом проекте. Но если говорить о большинстве проектов, то тут мы должны копнуть немного глубже. Один из способов – это проведение опроса заинтересованных лиц.

Ценность опроса заинтересованных лиц

Опрос заинтересованных лиц – это полуструктурированная дискуссия с теми, кто извлекает пользу с сайта. Это может быть тот, кто непосредственно работает на сайте (например, контент-редактор), или отдельные лица (например, руководители отделов), которые надеются с помощью сайта реализовать свои бизнес-задачи.

Опросы заинтересованных лиц дают четыре преимущества:

- **Они вводят веб-дизайнера в курс бизнес-требований**

Когда вы сталкиваетесь со сложной бизнес-моделью в новой области, опросы заинтересованных лиц ценны тем, что помогают вам понять требования заказчика. Из разговора с этими людьми вы узнаете о структуре организации и ее отрасли и определите, каким образом сайт будет соответствовать бизнес-требованиям заказчика.

- **Они дают более полный ракурс**

Многие веб-проекты крупных организаций влияют на многочисленные стороны бизнеса. Чтобы полностью осмыслить потенциальное влияние проекта, вы должны обсудить его со всеми участниками. Многие проекты заказываются одним отделом, у которого будет свое определенное видение целей. Беседуя с другими заинтересованными лицами, вы убедитесь, что проект, скорее, помогает, чем мешает другим людям внутри компании.

- **Они политически выгодны**

К сожалению, внутренняя политика является реальностью большинства крупных компаний. Это говорит о том, что нет недостатка в людях, которые хотят быть услышанными. Опросы заинтересованных лиц создают ту среду, где каждый может выразить свое мнение.

Я заметил, что если люди, работающие в организации, чувствуют, что к их мнению прислушиваются, они гораздо охотнее будут целиком и полностью поддерживать разработку. Опросы заинтересованных лиц также помогают утвердить дизайн, так как вы можете сослаться на их комментарии как на оценку определенных элементов дизайна.

- **Они обеспечивают доступ к лицам, которые в действительности принимают решения**

При работе над большим проектом вы будете часто иметь дело с младшими сотрудниками, тогда как те, кто в действительности принимает решения, остаются за кулисами. И в этом может заключаться проблема. Опросы заинтересованных лиц позволят вам поговорить с этими людьми и понять, чего они хотят.

Умело построенные опросы гарантируют проекту ясные и определенные цели, из которых каждый в компании извлечет свою пользу и в то же самое время утвердит общую выгоду.

Опросы заинтересованных лиц сосредоточены на разработке и будущем сайта. Но многое нужно узнавать и из того, что уже есть на сайте.

Исследуем то, что имеем в настоящий момент

Мы начинаем работу над многими проектами с анализа онлайн-присутствия клиента. Этот анализ часто выливается в отдельный проект еще до того, как мы рассмотрим вопрос о переделывании сайта. Это дает заказчику возможность вместе с нами поработать над маленькими задачами, прежде чем выделить ресурсы на большой проект. И ему будет легче принять решение о долгосрочном сотрудничестве с нами без предварительной оплаты.

Допуская, что заказчик откровенен на фазе исследования, у нас есть пять вариантов (в большинстве случаев выберем только один или два):

- **Стратегический анализ**

Стратегический анализ помогает увидеть, как выглядит сайт сейчас. Он показывает сильные и слабые стороны сайта и дает советы по улучшению. Это хорошая возможность дать заказчику извлечь выгоду из вашего опыта и показать ему, что вы профессионал, а не просто человек,двигающий пиксели.

- **Эвристический анализ**

Так же, как и стратегический, эвристический анализ выявляет силу и слабость существующего сайта. Разница в том, что эвристический анализ делают с помощью оценки нескольких критериев (от 1 до 3). Эвристический анализ дает оценку сайту, но не в плане стратегии улучшения.

- **Конкурентный анализ**

Конкурентный анализ проводится по критериям, схожим с эвристическим, но относят их к конкуренции. Это дает ценное представление и помогает заказчику учиться на ошибках конкурентов.

- **Аналитический отчет**

Аналитический отчет освещает проблемы с сайтом и является отправной точкой для оценки изменений. Он может многое прояснить в поведении пользователя. Например, вы можете узнать, у какого из разделов сайта конверсия лучше. Этот анализ также покажет точки выхода и этим вскроет проблемные страницы.

- **Персонажи**

Персонажи – это мощный механизм ориентации на пользователей. Набор персонажей расскажет нам о пользователях сайта и о том, чего они пытаются достичь. Они дают не только демографическую информацию, но и понимание того, как пользователи пользуются сайтом, какие разделы посещают. И хотя это процесс трудоемкий, данный вид исследования очень ценен.

Мы обсудили выгоды, которые мы получаем, если понимаем смысл дизайнерских решений и можем их обосновать. Мы коснулись того, как это помогает клиенту лучше вас узнать. Последний вопрос, который важно обсудить, это слаженные взаимоотношения с клиентом при работе над проектом.

Принцип совместной работы при переделывании сайта

Многие взаимоотношения, которые я наблюдаю между дизайнерами и заказчиками, рушатся. Дизайнер подчиняется заказчику и заканчивает работу изолированно.

Традиционные взаимоотношения «клиент-исполнитель» опасны по ряду причин:

- Клиент не извлекает выгоды из богатого опыта дизайнера.
- Дизайнер разочаровывается, потому что его воспринимают только как человека, двигающего пиксели.
- Клиент чувствует себя исключенным из процесса работы над проектом.
- Недостаток общения между двумя сторонами приводит к непониманию. Из-за того, что дизайнер работает изолированно, велик шанс создать нечто такое, что заказчик не примет.
- Иногда заказчик вынужден принимать важные решения в области, в которой он просто некомпетентен.

Мой принцип таков – совместная работа с заказчиком на равных, вовлечение его в каждую деталь процесса, а также более активная собственная позиция в принятии решений.

Вместо работы над дизайном в одиночку, я работаю рядом с клиентом, показываю ему эскизы, прототипы, карты настроений и варианты макетов дизайна.

Когда утверждается окончательный дизайн, заказчик с большей охотой одобряет его по трем причинам.

Во-первых, дизайн для него не явится сюрпризом. Клиент будет видеть работу, которая проводилась над ним, и поэтому финальное творение – это ее естественный результат. Для того чтобы данный принцип сработал, вам необходимо привлекать клиента к разработке дизайна.

Во-вторых, клиент будет ощущать себя владельцем проекта. На каждом этапе процесса он будет получать обратную связь от вас, а поэтому станет считать проект больше своим, чем вашим.

И, наконец, заказчик будет в состоянии понять, как родился дизайн, потому что он внес свою лепту в его разработку. Плюс в том, что заказчик, который понимает и чувствует свою причастность к дизайну, сыграет хорошую роль, когда будет убеждать всех в его достоинствах. Для работы по такому принципу вам необходимо вовлекать заказчика в процесс разработки.

Как правильно работать над проектом

Если вы работаете с дизайном чисто интуитивно, вам, конечно, будет трудно взаимодействовать с заказчиками. Ключевым моментом в совместной работе клиентом является привлечение его к решениям по разработке. Я не предлагаю посадить заказчика рядом с вами, когда вы работаете. Но он обязательно должен участвовать в разнообразных этапах разработки.

Я вовлекаю заказчика в несколько этапов разработки дизайна:

• На этапе обсуждения идей

До начала своей работы над дизайном, я сажусь с заказчиком, чтобы обмозговать с ним идеи. Это момент хорош для того, чтобы взглянуть на сайт, посмотреть, что воодушевляет

заказчика, и обсудить цвет, оформление и образы. Также это момент подходит для обсуждения персонажей.

- Основной вопрос, который я задаю: «Если бы сайт был знаменитостью, кто бы это мог быть?» Это помогает обеим сторонам визуализировать качества, которые вы сможете внести в дизайн.

- **После набросков карт настроений**

После первоначального мозгового штурма я иду и создаю карты настроений, где отражаю различные направления дизайна. Затем я обсуждаю их с заказчиком и продолжаю отшлифовывать. Я не слишком сильно усердствую. Нужно, чтобы их было легко создавать и чтобы я мог быстро воспроизвести их заново. Это мой шанс опробовать различные эстетические подходы одновременно.

- **Когда делаю прототипы**

Я всегда планирую встречу с клиентом (и даже с другими участниками) для обсуждения каркаса (wireframe). Мы садимся и делаем наброски ключевых страниц. Не нужно делать прототипы с высокой точностью – достаточно того, чтобы заказчик почувствовал, что он внес вклад в развитие данного направления. А я отшлифую их потом, после встречи.

- **Шлифовка макетов**

Когда придет время представить заказчику дизайн (или HTML-прототип), это не будет для него сюрпризом, потому что все основано на картах настроений и каркасах. Но я все же оставляю клиенту возможность обсудить напоследок любой вопрос, прежде чем перейду к финальному этапу.

Вы будете приятно удивлены, как мало итераций вам придется сделать в процессе выполнения проекта, если вы примените этот метод. Клиент часто счастливо одобряет дизайн всего лишь с несколькими изменениями, потому что он был вовлечен в процесс работы над ним с самого начала. Однако будьте осторожны: все может пойти наперекосяк, если вы покажете дизайн тем, кто не принимал участия в процессе разработки. И вот тут презентация дизайна просто необходима.

Презентация проекта

Все титанические усилия, которые вы прилагали в совместной работе с клиентом, могут пойти насмарку, если последнее слово осталось не за ним. Тесная работа с клиентами вероятнее всего приведет к тому, что внутренне они будут защищать дизайн, но тот, кто увидит его впервые, не сможет дать обоснованную оценку, потому что у него нет полной информации о разработке.

Единственный выход из данной ситуации, который я нашел, это презентация дизайна остальным лицам, которые также принимают решения.

Я уверяю вас, что таким образом они видят всю скрупулезность процесса моей работы, а также слышат мои идеи, вложенные в дизайн.

В идеале эта встреча может быть проведена в форме живого общения или конференции по телефону. Но это не всегда возможно. Другое решение, которое я нашел, это представить дизайн на рассмотрение с пояснениями. Оформленный как видеофайл с голосом за кадром, он дает хороший эффект. Это гарантирует, что все будут иметь возможность услышать объяснения процесса разработки.

Таким образом, зрители не дадут оценку дизайну по скудному первому впечатлению. Судя по своему опыту, скажу, что заказчикам эта идея тоже нравится. По их многочисленным отзывам, такой способ презентации дизайна впечатляет гораздо больше, чем статическое

изображение. Видео также хорошо действует сейчас, когда сайты становятся все более и более динамичными.

Вместо статического изображения вы можете показать элементы JavaScript и даже, при необходимости, адаптивного дизайна. Лучше всего то, что сайту не надо быть кросс-браузерным, поскольку он работает в одном браузере. Конечно, дела могут пойти из ряда вон плохо, и неважно, насколько хорошо вы презентуете дизайн, если заинтересованные лица начнут давать обратную связь.

Обратная связь через тестирование

Мы любим считать себя разумными существами. На самом деле, мы не такие. На нас все имеет влияние, начиная с детского опыта до невыпитой чашки кофе сегодня утром.

Ни на что люди не реагируют так эмоционально, как на дизайн. Все знают, что им не нравится, и у всех нас разные вкусы. Один из моих клиентов отклонял даже цвет дизайна, потому что он напоминал ему о платье, которое носила его пожилая родственница!

С таким субъективным подходом к дизайну прийти к консенсусу в плане восприятия сайта трудно. Хотя существуют основные принципы дизайна, также присутствуют факторы, по которым одна и та же вещь кому-то кажется великолепной, а другому – ужасной. Если не обуздать процесс, то этап утверждения превратится в лотерею. К счастью, кое-что можно предпринять. Секрет кроется в психологии.

Так как мы люди, то мы любим считать себя последовательными во взглядах. Вот почему заказчик, который вовлечен в процесс создания дизайна, с гораздо меньшей охотой будет отклонять его. Тогда бы это было противоречием самому себе.

Другая причина, по которой данное свойство играет нам на руку, это то, что людям нравится считать себя логичными. Если вы представляете логичный, объективный аргумент в пользу дизайна, то слушатель гораздо быстрее примет его, ведь он должен реагировать логически. Желание быть последовательным и поддерживать свой имидж может даже преодолеть его личную неприязнь к дизайну.

Вспоминаю, как несколько лет назад я делал дизайн в отрасли высшего образования. Сайт был прямо нацелен на то, что тогда называлось «поколением MySpace»⁷. Лично я ненавидел этот дизайн, и мой заказчик тоже.

Однако тестирование пользователей, на которых мы ориентировались, в конечном итоге показало, что живенький дизайн и кричащие краски задают им нужный тон. Перед лицом этих фактов о нашем собственном мнении нам пришлось временно забыть.

Вот почему тестирование дизайна так важно. Оно отводит решения от субъективного мнения и приближает их к объективному исследованию. Оно также концентрируется на пользователе. Заказчик может ненавидеть дизайн, но если тот получает резонанс от пользователя и выполняет бизнес-задачи, то это должно подействовать на него.

А как вы будете уговаривать клиента принять метод тестирования при переделывании его сайта?

Продажа тестирования заказчику

Проблема тестирования в том, что на него тратятся время и деньги. Многие клиенты с неохотой платят за это. Им больше нравится опираться на своих «экспертов» и свои собственные предпочтения. Поскольку основная причина, по которой клиент не хочет проводить тестирование, кроется в деньгах, то вам придется сделать на них акцент при обосновании тестирования. Когда я убеждаю заказчика провести тестирование, я говорю ему о двух финансовых выгодах.

Во-первых, тестирование может значительно повысить эффективность сайта. Это гарантирует, что дизайн четко передаст ваши сообщения и побудит пользователя откликнуться на призыв к действию. В конце концов, это скорее заставит сайт генерировать значительный возврат на инвестиции.

⁷ Некогда популярная социальная сеть. – Примеч. ред.

Во-вторых, тестирование пользователей поможет сэкономить значительную часть денег. Без него вы можете пойти по неэффективному пути. Будут даже такие моменты, когда вы станете выбрасывать некоторый функционал, потому что тестирование покажет, что пользователю он не нужен. При своевременном и частом тестировании вы улавливаете проблемы с интерфейсом пользователя и набором функций до того, как потратить слишком много драгоценного времени.

Короче говоря, без тестирования пользователей процесс переделывания сайта будет проходить вслепую. Нет никакой гарантии, что результат будет эффективным. А в процессе непременно будут допускаться ошибки.

И раз уж заказчик принимает участие в тестировании, то вам нужно выбрать наиболее подходящий его вид.

Какой вид тестирования проводить

Есть несколько возможных вариантов, и выбрать правильный очень важно. Я остановлюсь на трех, которые я применяю чаще всего.

В среде веб-дизайнеров опрос как метод вытягивания информации из пользователей не используется широко. Но при определенных обстоятельствах он может стать эффективным. Я работал с клиентами из научных кругов, которые ставили под сомнение ценность тестирования при небольшом количестве участников.

На их взгляд, объем выборки был слишком мал, чтобы иметь статистическую важность. Опрос же оправдал себя как решение, потому что в него было вовлечено значительно большее количество людей.

Для достижения наибольшего эффекта нужно воспользоваться вопросами с ограниченным набором вариантов ответа, это даст возможность провести сравнение.

В опрос хорошо включить следующие пункты:

- Выбор между двумя подходами к дизайну, с едва уловимой разницей.
- Получение мнения по поводу того, правильно ли дизайн отражает ключевые качества, характеризующие бренд (например, свободный и небрежный либо профессиональный дизайн, консервативный или либеральный?).
- Вопрос пользователям, какими опциями из ограниченного набора они будут пользоваться.
- Проведение сортировки закрытых карт (где пользователь должен систематизировать страницы по категориям).

В инструментальных средствах (в сервисных программах, в серверах), поддерживающих этот вид онлайн-тестирования пользователей, нехватки нет. Я лично использую два – Verify4 и WebSort .5. Польза от опроса в том, что ответы подсказывают определенный тип дизайна, а клиент приобретает уверенность при принятии решения.

Как получить отзывы о дизайне

Второй вид тестирования пользователей – тестирование дизайна. Здесь проверяется, насколько эстетичен и удобен в использовании интерфейс. Обычно я провожу тестирование дизайна на начальной стадии разработки с относительно большой группой пользователей (20 и больше). Опрашивать их можно индивидуально или в малых группах. Я провожу несколько тестов, но наиболее распространенные – это тест на эмоциональную реакцию и флеш-тест.

В тесте на эмоциональную реакцию вы показываете пользователю дизайн и (как было сказано выше) просите его сделать выбор между ключевыми парами. Дизайн тесный или просторный, классический или современный? Смысл в том, чтобы оценить, соответствует ли дизайн тому образу, который вы представляете. Ключевые слова, которые выберет пользователь, должны содержать желаемые результаты. Если пользователи выбирают эти слова, я знаю, что я на правильном пути в разработке эстетики.

Иногда, правда, такие эмоциональные тесты могут преподносить сюрпризы. Мой коллега показывал один свой дизайн пожилой даме. Увидев его, она вдруг разрыдалась. Просто оказалось, что элемент дизайна – собачка – напомнила ей о ее больном питомце. Это говорит о том, что когда вы проводите тестирование дизайна, тенденции лучше всего искать по его результатам, чем по индивидуальным комментариям.

Флеш-тест основывается на контенте и визуальной иерархии. Вы показываете дизайн пользователям на несколько секунд и потом убираете его. Затем вы просите их воспроизвести элементы страницы.

Воспроизведенные элементы и их последовательность хорошо указывают на то, правильно ли расставлены акценты в дизайне. Например, если пользователю не удастся назвать ваш главный призыв к действию, значит, скорее всего, что-то нужно изменить. Оба эти теста могут проводиться в форме опроса и, таким образом, вовлекать намного больше пользователей. Но беседа с пользователями вживую имеет преимущества. Это помогает не только лучше понять их, но и дает вам шанс следить за их вопросами и копнуть немного глубже.

Юзабилити-тестирование

Наверное, самый хорошо известный вид тестирования – это проверка удобства пользования (юзабилити-тестирование). Как видно из названия, оно проверяет удобство и простоту использования вашего дизайна больше, чем его эстетический вид. Подобную тему широко раскрыли Стив Круг, Якоб Нильсон и Джаред Спул, поэтому я не хочу здесь слишком сильно углубляться в нее.

Я же хочу подчеркнуть, что успех юзабилити-тестирования зависит от многократных подходов в течение всего жизненного цикла проекта. В течение нескольких лет я откладывал тестирование пользователей на конец проекта, когда было уже слишком поздно вносить изменения. Тестирование на завершающей стадии также не дает возможности провести последующие, чтобы подтвердить эффективность усовершенствований.

Для эффективного тестирования его следует проводить на каждом этапе процесса переделки сайта. Мы должны тестировать абсолютно все, от эскизов, карт настроений и до заверченного дизайна. Проведение однократного теста недостаточно.

Когда и кого тестировать

Причина, по которой тестирование так часто сводится к формальной сессии в конце проекта, кроется в очевидных затратах на него. Люди думают, что найти участников и провести сессию – это процесс дорогой и трудоемкий. Позвольте мне убедить вас в том, что это не так.

Юзабилити-тестирование редко требует точного соответствия по демографическому признаку между выборкой респондентов и предполагаемой целевой аудиторией. Большинство преград, с которыми мы сталкиваемся и которые мешают удобству пользования, обычны для всех.

Для того чтобы набрать участников опроса нужно не больше и не меньше, чем ссылка на существующий сайт. Исключением является тестирование дизайна, для которого тре-

буется приложить немного больше усилий, чем просто набор участников. Их количество должно быть сравнительно высоким, и подбор нужных людей действительно имеет значение. Но сами по себе тесты могут проводиться где угодно и без специального оборудования.

Только однажды я присутствовал на сессии по тестированию пользователей, которая проводилась в специальном центре тестирования с полупрозрачными зеркалами и подключенным видео. Но я думаю, что на самом деле это было менее эффективным, чем партизанское тестирование. У вас будет больше возможности регулярно устраивать тестирование, если вы будете проводить его облегченную версию.

В конце концов, тестирование помогает сделать сайты лучше и легче получить одобрение клиента. Это говорит о том, что клиенты хотят сказать свое слово, а поэтому мы должны уметь получать обратную связь от них.

Как быть с отзывами заказчика

Хотя тестирование реальных пользователей уводит заказчиков от собственного мнения, иногда им тоже нужно помочь правильно выразить свои отзывы о дизайне.

Мы ухудшим положение вещей, если спросим: «Как вы думаете?» или «Не могли бы вы дать мне свой отзыв?» Подобными вопросами мы стимулируем их выражать свое личное мнение. Вместо этого я задаю заказчикам ряд специальных вопросов, когда собираю их отзывы по определенному дизайну. Вот некоторые неплохие вопросы:

- Вы согласны с тем, что изменение дизайна отражает ценность бренда компании?
- Соответствует ли переделывание сайта тем бизнес-задачам, которые мы обговаривали?
- Соответствует ли переделывание сайта персонажам, которые мы обсуждали?
- Побудят ли пользователей утвержденные призывы к действию после переделки сайта?
- Отражает ли дизайн принятые карты настроений и прототипы?
- Все ли подходящие отзывы, полученные из тестирования, мы учли?

Весь ваш труд окупится авансом. Когда заказчик будет иметь возможность обращаться к утвержденным картам настроений, прототипам, результатам тестирования пользователей и т. п., он больше сосредоточится на бизнес-задачах и потребностях пользователей, чем на собственном мнении.

Одержимость личным мнением не единственная проблема. Часто обратная связь от клиента представляет собой список изменений в дизайне, которые вы должны внести по их желанию. К сожалению, эти изменения не всегда в лучшую сторону.

Я активно поддерживаю своих клиентов в том, чтобы они вносили свои предложения по улучшению дизайна. Ведь я же верю, что они могут дать ценную информацию, особенно если вы вовлекаете их в процесс дизайна и обучаете их.

Положение вещей становится угрожающим, когда клиент предлагает решение проблемы, которое он не может четко изложить. Например, клиент просит вас изменить цвет с голубого на розовый, не называя причины. А причина в том, что аудиторию составляют девочки в возрасте 9–12 лет.

Не зная сути проблемы, вы не предложите лучшее решение и не оцените, насколько к месту идея заказчика. Поэтому важно, чтобы он сначала мог высказать проблемы, а потом предложить решения.

Если у заказчика входит в привычку предлагать решения скорее, чем выражать проблему, то обычное напоминание часто возвращает их на место.

Не удастся – задайте вопрос «зачем?». Как я уже сказал, вопрос «зачем?» поможет клиенту разобраться в корне проблемы.

Суть данного раздела сконцентрирована на процессе работы с заказчиком при создании сайтов, ориентированных на бизнес-задачи. Мы увидели, как нужно проводить исследования и тестирование проекта, а также рассмотрели методы работы с заказчиками. Прежде чем подвести черту, давайте посмотрим, насколько долго может продлиться жизнь сайта, который вы создаете. И в особенности как обеспечить будущее сайту после его переделывания.

Вечная переделка

Я должен признаться, что как веб-дизайнер я не всегда был дальновидным в своем методе создания сайтов для заказчиков, что шло во вред мне и им. Уверен, что я не один такой.

Мы совершенно не виноваты в этой недальновидности. По большому счету она – порождение культуры, о которой я говорил ранее. Когда клиент каждые несколько лет утверждает новый сайт (зачастую, всякий раз с разными дизайнерами), мало толку планировать его долгое существование.

Я уже объяснял недостатки подобного подхода, как для клиента (с точки зрения эффективности сайта), так и для дизайнера (с точки зрения дальнейшего сотрудничества). Поэтому мы должны строить непрерывные рабочие отношения с заказчиками, а не ограничиваться однократными встречами. Это срабатывает независимо от того, дорабатываем ли мы сайты или переделываем их с нуля.

Как создать программу непрерывной работы

Как мы уже обсудили, есть веский довод для того, чтобы непрерывно инвестировать в сайт. Однако представлять что-то и осуществлять это на практике – две разные вещи. Важно настроить механизмы, которые обеспечат непрерывные инвестиции.

Я предложил составлять перечни пожеланий и поэтапное развитие сайта как способы подтолкнуть клиента к планированию. Но это не единственные варианты. Раз в три месяца мы устраиваем с заказчиками встречи, где обсуждаем, как можно усовершенствовать их сайты. Также мы проводим ежегодный анализ сайта и предлагаем варианты его усовершенствования на будущий год.

Какой бы механизм вы ни использовали, будь то инфобюллетени по электронной почте или ежегодные анализы, цель одна: показать заказчику, как сайт может перейти на следующий уровень. И помните, что вы должны не просто представить ему новые функциональные возможности и техники. Вы должны объяснить бизнес-выгоды, которые можно извлечь из этого. Только тогда они увидят возврат инвестиций, которые сделают, воплощая ваши идеи.

Однако кроме текущей программы инвестирования есть и другие способы обеспечить сайту его дальнейшее будущее. Существуют еще технические решения.

Использование технологий для обеспечения будущего сайту

Нам следует помнить, зачем и что мы делаем как веб-разработчики. Если мы об этом забудем, то начнем практиковать неправильные вещи. Речь идет о веб-стандартах.

В основе веб-стандартов лежат простые принципы: мы должны разделять контент, дизайн и логику работы сайта. Одним из многих преимуществ этого подхода является то, что это облегчает изменение сайта в будущем.

Веб-стандарты помогают обеспечивать будущее сайтов. Но все равно я вижу тревожное количество сайтов, созданных с CSS, HTML и JavaScript, в которых это разделение не так явно: строки JavaScript либо встроены непосредственно в код HTML, либо рассчитаны на присутствие определенных элементов HTML, а тот, в свою очередь, «нашпигован» классами и идентификаторами элементов, чья единственная цель – задание внешнего вида страницы. Не поймите меня неправильно: я не ярый борец за чистоту кода. Я знаю степень неминуемо-

сти таких накладок в коде. Когда вы пишете код, спросите себя, как ваши решения повлияют на обновления через год-другой.

Следующий вопрос, который возникает, когда мы говорим о будущем сайта, – это поддержка браузеров. Как веб-дизайнеры, мы много говорим о поддержке старых браузеров, но мало о поддержке будущих. Мы должны обеспечивать доступ к сайтам через старые браузеры, но нам также нужно думать и о будущей жизни сайта.

Поддержка старого браузера, чья рыночная доля снизится за счет появления новых, имеет мало смысла в финансовом плане. Я – ярый сторонник создания сайтов с использованием HTML5 и CSS3, так как мы знаем, что эти технологии будут все больше и больше поддерживаться. Мы создаем сайты для будущего, не заикливаясь на браузерах, чья рыночная доля идет на убыль. Понятно, что можно установить баланс, но кто-то может возразить, что поддержка старых браузеров – не лучшее вложение и без того ограниченных ресурсов. Говоря о поддержке, я не могу не посвятить часть раздела мобильному будущему сайта.

Планирование мобильного будущего сайта

Как веб-разработчики, мы входим в азарт от мобильности. Однако мобильность все еще не стоит на повестке дня у многих заказчиков. Для многих из них использование мобильности еще сравнительно невысокое, и поэтому они неохотно вкладываются в него.

Несмотря на это, мало кто выражает свой скептицизм по поводу того, что мобильность может сыграть существенную роль в будущем сайта. Это означает, что даже если поддержка мобильности сайта не является одной из бизнес-задач компании в ближайшем будущем, клиентам нужно уже сейчас планировать ее.

Шаги, которые следует предпринять, будут зависеть от того, какую мобильную стратегию предпочтет клиент. У него есть несколько вариантов, и нам надо рассказать клиенту о них.

Для начала он должен решить, что он хочет: приложение или сайт. Приложения ориентированы на задачи и узко специализированы, тогда как сайт направлен на контент и его возможности намного шире. Если заказчик выбирает приложение, то вопрос в том, какое оно будет: основанное на веб-технологиях или «родное», т. е. написанное, например, под конкретную мобильную платформу. Это сложное решение, которое мы не можем подробно рассмотреть в этом разделе⁸.

Так или иначе, здесь представлены некоторые моменты, которые вам необходимо обсудить с заказчиком:

- Какие «родные» характеристики должны быть доступны?
- Вы можете позволить себе разработку под множество платформ?
- Вы захотите разделить доходы с владельцем магазина приложений?
- Ваше приложение будет доступно через магазин приложений?
- Есть ли какие-либо выгоды от распространения приложения через такие магазины?

Когда клиент останавливается на сайте, то и выбор становится проще. Если сайт будет переделываться глобально, то сейчас самое время сделать его отзывчивым (т. е. заставить его реагировать на доступное экранное разрешение). На маленьких экранах (таких как смартфоны) макет упрощается для удобства использования, тогда как на больших экранах у вас будет обычный макет под настольные компьютеры.

Если сайт не планируется серьезно переделывать в скором времени, тогда предпочтительней выбирать адаптивный подход. Адаптивный дизайн легче осуществлять на существу-

⁸ Вы можете узнать больше о мобильной платформе в редизайне в разделе Арэла Балкана в этой книге.

ющем сайте, это позволяет изменить формат ключевых экранных размеров (таких как экран планшетного компьютера).

Однако адаптивный дизайн, так же как и отзывчивый, подойдет не для каждого макета, и хотя адаптивный сайт может выглядеть великолепно на сегодняшних устройствах, есть вероятность, что он окажется неоптимизирован для будущих.

В настоящее время Интернет переживает еще одно глобальное развитие, где применяются новые технологии, новые устройства и методы. И нам, веб-разработчикам, выпала миссия помочь нашим заказчикам подготовиться к встрече с этим прекрасным новым миром.

Что дальше?

В этом разделе мы осветили много тем, но не слишком углублялись в них. Основная идея была в том, чтобы научить вас думать на более высоком уровне, чем только о технологии и дизайне, концентрироваться на сложных проблемах, с которыми сталкиваются клиенты, и обеспечивать эффективность их сайтов насколько это возможно. Все, что вам нужно, это взять на вооружение несколько принципов. Вот некоторые рекомендации:

- **Будьте кем-то большим, чем просто исполнителем проекта**

Неустанно работайте над тем, чтобы изменить ваши взаимоотношения с клиентами. Прекратите быть просто человеком,двигающим пиксели, работайте в команде с клиентами. Умейте возразить им, особенно когда они требуют глобальной переделки сайта. Вместо этого предложите им преобразование и вовлекайте их в этот процесс.

- **Подготовьтесь перед стартом**

Сдерживайте свои порывы приступить к переделыванию с ходу; сначала лучше выполните домашнее задание. Убедитесь, что вы готовы к тому, что проект может разрастись. Убедитесь с помощью тестирования заинтересованных лиц и аналитики действующего сайта, что вы понимаете бизнес-задачи.

- **Тестируйте все**

Не полагайтесь только на свой опыт разработчика. Проводите тестирования для того, чтобы процесс разработки был менее субъективным, и для обоснования изменений. Тестирование также поможет вам подсчитать потенциальный возврат вложений, который клиент ждет от вашей работы.

- **Планируйте будущее**

Разработайте с вашим заказчиком действующую программу развития, которая обеспечит будущее сайту. Подтолкните его к тому, что нужно принимать во внимание мобильный Интернет и будущие браузеры, а не ориентироваться на старые технологии, чья доля идет на убыль.

Вот и все. Надеюсь, что я не слишком деморализовал вас всеми этими разговорами о возврате инвестиций (ROI) и бизнес-механизмах. На самом деле это может быть захватывающей штукой. Мы разработчики, а не художники. Основное отличие в том, что мы создаем вещи, которые решают проблемы. Иногда это проблемы пользователей, но иногда это проблемы, которые испытывают сами клиенты.

Об авторе



* * *

Пол Боуг (р. 1972) вырос в Уашворде (Великобритания), учился в Университете Портсмута и начал свою веб-карьеру в IBM в 1994 году, когда веб-дизайн включал в себя блокнот, серый фон и не имел возможности верстки. Пол – соучредитель «Хэдскейп» (Headscapе – компания, работающая в области веб-дизайна и адаптивного дизайна). Здесь он работает над веб-стратегиями, пишет об успешных сайтах и выступает на конференциях по всему миру. Время от времени он выпускает подкасты. Пол живет в небольшом городке в самом сердце английской сельской местности и активно участвует в жизни местной церкви. Он любит проводить время, играя в Skyrim (компьютерная игра). Пол считает себя любителем животных, потому что его отец – фотограф дикой природы, поэтому у них дома все время были все: от совы до оленей. Его любимые цвета – все оттенки серого. Самый большой урок, который он изучил за свою карьеру, – быть страстными и полным энтузиазма, никогда не терять любовь к Интернету и играть с инновациями. Послание Пола к читателям гласит, что успешность проекта обеспечивается сотрудничеством с клиентом. Возможно, вам непросто работать с клиентом, но без его знаний о бизнесе и сфере деятельности сайт будет провальным. Более того, клиент должен любить ваш дизайн. Если ему не нравится дизайн, он не будет вкладываться в него и использовать в полную силу. Вы обязаны работать совместно.

О рецензенте



* * *

Коллис Тайед – соучредитель и генеральный директор [Evato](#). Он начинал работать в качестве веб-дизайнера, делая макеты в Photoshop и нарезая их для разработки большинства ранних сайтов [Envato](#). Сейчас Коллис больше времени проводит, планируя, разрабатывая стратегии и работая с электронной почтой, но веб-дизайн всегда будет в его сердце!

Выбор платформы: технические решения для переделки вашего сайта

Автор: Рэйчел Эндрю

Рецензенты: Райан Карсон, Харли Финкельштейн

Этот раздел написан для тех, кто участвует в планировании работ по переработке сайта. В предыдущем разделе Пол Боуг рассматривал бизнес-сторону полной переделки и поэтапной доработки вашего сайта. В этом разделе мы обсудим некоторые технические проблемы, с которыми вы можете столкнуться в своей работе. Для переделывания интерфейса сайта не всегда требуется полностью переписывать код. Мы дадим вам несколько советов, как утрясти этот вопрос.

Прежде чем окунуться с головой в глобальное переделывание сайта, мы, по совету Пола, должны узнать все о существующем сайте и серверах, которые им управляют. При переработке системы очень легко потерять ценную информацию. Поэтому даже если вы полностью переделываете сайт или выбираете новую платформу, узнайте все, что только можно, о существующем сайте, чтобы не повторять ошибок ваших предшественников.

Прежде чем что-то предпринять, вы должны вникнуть во все технические требования. Возможно, переработка сайта добавит новые функции, которые нужно будет поддерживать. Например, для сайта электронной коммерции вам нужно знать, как принимаются платежи по Интернету и какие сервисы нужны для этого. Хостинг, который вы используете, должен удовлетворять техническим требованиям нового сайта или доработке существующей платформы. Поэтому не забудем подумать о том, как выбрать хороший хостинг.

Начало работы над проектом – прекрасный момент, чтобы оценить существующий сайт и убедиться, что как специалист вы в хорошей форме. В конце этого раздела вы найдете идеи о средствах разработки, контроле версии и об изменении версии существующего сайта. Как и в предыдущем разделе, мы обсудим здесь многое, только не слишком подробно. Формат книги не позволяет. Наша главная цель – заставить вас задуматься о различных аспектах сайта и выбрать те основы, которые вы будете применять в своей дальнейшей работе

Для кого этот раздел?

Неважно, с кем и для кого вы переделываете сайт: привлекаете ли подрядчиков для создания собственного сайта или работаете в команде разработчиков в качестве нетехнического специалиста, этот раздел – для вас. Изучив ее, вы поймете процесс переделывания, и это принесет вам пользу.

Возможно, вы тот, кто занимается разработкой сайта постоянно либо для себя (в этом случае заказчик вы сами), либо для стороннего заказчика или работодателя. В этом случае все в этом разделе будет для вас интересным. Для разработчика переделывание сайта – это фантастическая возможность. Сайт или приложение, которые у вас есть сейчас, имеют не только сильные и слабые стороны, но и базу пользователей, и посещаемость. Все это вы сможете изучить.

Проигнорировав на свой страх и риск эти данные, в конечном итоге вы просто воссоздадите те же самые проблемы, которые существуют и сегодня, либо не включите необходимые пользователям важные функции. Когда вы занимаетесь кардинальной переделкой сайта, у вас есть возможность узнать о его болевых точках и проблемах от пользователей и владельца. Система управления контентом (CMS⁹) очень сложная, поэтому сайт не обновляется? Клиенты интернет-магазина постоянно звонят и просят помочь им сделать заказ? Дизайн ограничивает возможность добавления текста и фотографий? Если вам удастся решить эти проблемы – создать CMS, которой люди будут пользоваться с удовольствием, или наполовину сократить количество звонков от обескураженных покупателей, – вы получите бескрайнюю благодарность вашего клиента или начальника. Согласитесь, приятно знать, что ваша работа улучшает чью-то жизнь или помогает развивать бизнес.

Если разработкой сайта занимается подрядчик, этот раздел все равно будет интересен для вас. Мы не станем слишком глубоко копаться в технических темах. Общее понимание вопросов, которые обсуждаются в этом разделе, сделает ваше общение с разработчиками легким и непринужденным.

Теперь допустим, что ваша роль – нетехническая. К примеру, вы работаете исключительно над графическим дизайном, либо занимаетесь управлением проекта, либо копирайтингом. Опять же понимание того, что делают разработчики, только поможет общению.

⁹ Content management system – система, позволяющая изменять контент на сайте, создавать новые страницы с помощью так называемой «админки» сайта.

Узнать все о действующей платформе

Как уже было сказано, все, что вы знаете о сайте, который собираетесь переделывать, очень важно. Поэтому нужно не только проанализировать сайт самому, но и поговорить с людьми, которые им пользуются: с его владельцами, посетителями и теми, кто его поддерживает.

Поинтересуйтесь, что им нравится в существующем сайте. Это может быть все, что угодно. Например, клиенту нравится, как выглядит сайт и как в нем представлен бренд, или, возможно, он находится на хороших позициях в поисковой выдаче. Пользователи могут сказать, что сайт удобный и можно обойтись без всяких прибабасов, которые так любят заказчики.

Поговорите с теми, кто обновляет контент и узнайте, что они думают о функциях существующего сайта. Если вы решите сменить платформу, берите на карандаш все, чем люди пользуются в своей работе. У многих она строится полностью вокруг системы. Бывает, что, отняв у людей возможность вести отчетность по заказам или делать записи в блог, вы явно усложните им жизнь. Если вы не заменяете эти функции, то вам лучше предоставить систему более высокого качества!

В каждой системе есть вещи, которые сводят пользователей с ума. Но даже если вы обслуживаете сайт, не считайте себя знатоком в них. Много раз я сталкивался с тем, когда недостаточно подкованные технически пользователи думали, что неполадки в системе возникли по их вине и поэтому молчали. И всякий раз, когда не удавалось сохранить данные, они думали: «Опять я это сделал!» – и повторяли все заново. Вместо того чтобы поднять проблему, пользователи списывали ее на то, что не умеют пользоваться компьютером.

Но даже если пользователь и виноват, а система допускает постоянную потерю данных, ваша задача – принять превентивные меры. И все же вопиющие ошибки часто не регистрируются по несколько месяцев, а пользователи предпочитают молчать и месяцами работать с ними. Если вы собираетесь сохранять уже существующую систему, отыщите и исправьте эти болевые точки. Поверьте, те, кто пользуется системой каждый день, смогут тогда вздохнуть спокойно.

Есть системы, которыми люди пользуются в своей ежедневной работе постоянно. Так вот неплохо было бы отследить этот процесс. Что они делают изо дня в день? Очень часто человек даже не в курсе, что новая система может избавить его от изнурительной работы. Я видел пользователей, которым приходилось по несколько раз вводить в систему одни и те же данные. К примеру, они вводили одну и ту же информацию в разные места или копировали данные из одного документа в другой, тогда как очень простым решением стала бы автогенерация отчетов или добавление в CMS скрипта, который копирует данные из одного места в другое. Если вы будете просто смотреть на сайт с управляемым контентом или систему электронной коммерции, то ничего не увидите. Садитесь с рядом с администратором и наблюдайте, как пользователи работают в системе.

Если вы переделываете сайт, то придется решать не только видимые проблемы. Вам нужен сайт с такими функциями, которых у него нет сейчас? Хотите продавать онлайн? Решили наконец-то, что сайту нужна система управления контентом? Текущей CMS сложно пользоваться, или она не поддерживает контент, который вы хотите создавать?

Если вы будете знать, каких функций не хватает сайту, то решите, нужно ли вам сменить систему или дорабатывать ее дальше. Как только вы поймете, как пользователи и администраторы работают с сайтом или приложением, можете начинать собирать техническую информацию для переработки

Сбор технических требований

В предыдущем разделе мы обсуждали требования к сайту. Здесь мы поговорим о том, как выполнить их технически. Остерегайтесь заявлений типа: «Мы хотим, чтобы все было так, как в текущей системе» – если только не вы ее создали! Если клиент этого хочет, убедитесь, что вы получили подробную информацию обо всем том, что должно быть в новой системе.

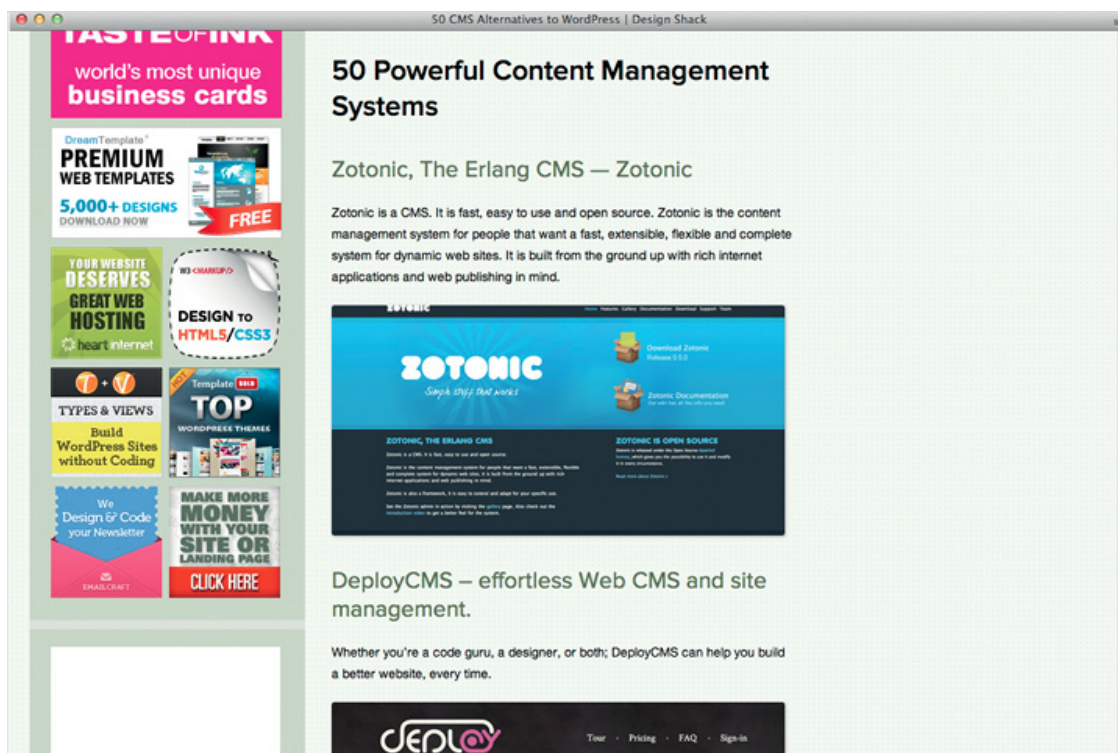


Рисунок 2.1. Любой, кто выбирает CMS, столкнется с массой опций. Чтобы решить, что работает для вашего проекта, нужно понимать технические требования

Не сделаете этого – готовьтесь к тому, что перед самым запуском вас спросят: «А как эта система поддерживает задачу, которую пол-офиса делает каждый день?» (хоть это и не так на самом деле). И таким образом заставят вас вносить значительные дополнения в проект. Знаю это по своему опыту!

Управление контентом

Почти всем сайтам требуется та или иная форма управления контентом. Это может быть обновление страниц, добавление продукта в магазин или редактирование текста в приложении. В какой степени клиенту необходимо управлять контентом и кто будет его редактировать? В этом разделе мы будем использовать термин CMS в очень широком смысле, для описания любого инструмента редактирования контента – от простого текстового редактора до полнофункциональной корпоративной CMS бизнес-уровня.

Вот некоторые мысли по поводу выбора CMS:

- Всем ли редакторам (контент-менеджерам) нужны одинаковые права доступа?

- Будут ли управлять контентом специальные сотрудники или это станет частью работы других людей?
- Нужно ли поддерживать многоязычность?
- Какой тип контента будет редактироваться?
- Какие нужны средства редактирования?

Всем ли редакторам (контент-менеджерам) нужны одинаковые права доступа?

Итак, для работы на сайте нужен один редактор (например, владелец малого бизнеса) или несколько? Если второй вариант, то все ли редакторы будут иметь равные права? Или один будет иметь доступ к каким-то разделам системы, а другие нет?

Вот несколько сценариев:

- Сайтом большой компании управляет один человек, который утверждает весь контент, а в подготовке контента принимают участие несколько сотрудников. Клиент хочет, чтобы эти редакторы могли создавать контент и отправлять его на рассмотрение. После одобрения, он публикуется управляющим редактором.
- Компания хочет, чтобы отдел кадров мог публиковать и снимать вакансии на сайте и вел раздел, который касается вопросов найма сотрудников. У этих пользователей не должно быть прав для внесения изменений в другие разделы.
- Владелец сайта электронной коммерции не хочет, чтобы его редакторы имели доступ к собранным в системе отчетам по продажам или информации о покупателях. В то же время он хочет, чтобы его бухгалтер имел доступ к данным о продажах, и ни к каким другим больше.
- Владелец сайта хочет, чтобы все желающие могли писать в блог, но не трогали контент других разделов.

Будут ли управлять контентом специальные сотрудники, или это станет частью работы других людей?

Очень важно понять, что могут люди, которые создают и редактируют контент. Речь здесь не только о технических умениях, к примеру хорошо ли они знакомы с CMS, но и о том, насколько они чувствуют дизайн и контролируют, как все выглядит. Также проверьте их копирайтерские способности. Если вначале контент пишет опытный копирайтер, а после поддержкой занимаются владелец или сотрудник, не занимающийся копирайтингом, то CMS должна помочь им понять, что и как писать. Об этом – в статье «Ваша CMS как куратор дизайна и контента»¹⁰.

Нужно ли поддерживать многоязычность?

Даже если вы запускаете одноязычный сайт, но в ближайшем будущем нужно будет поддерживать несколько языков, подготовьтесь к этому заранее. Добавить на сайт поддержку нескольких языков гораздо сложнее, чем заложить эту функцию сразу. Конечно, это требование ограничивает выбор вариантов готового ПО.

Если сайт нужно переводить, выясните, как будет проходить этот процесс, чтобы поддерживать его в системе. Переводчики будут просто переводить весь контент в Word-доку-

¹⁰ [Andrew, Rachel. Your CMS as Curator of Your Design and Content / smashed.by/cms-curator.](https://smashed.by/cms-curator)

ментах или чем-то подобном, а потом отправлять редактору, чтобы он вносил их в систему? Не будет ли переводчику проще читать и переводить контент прямо в CMS?

Какой тип контента будет редактироваться?

У большинства сайтов, которым требуется CMS, относительно статические страницы и четкая стандартная древовидная структура. Наиболее логично создавать такой контент постранично, чтобы администраторы легко могли найти страницы для редакции.

В некоторых сайтах основная функция – блог и несколько поддерживаемых страниц. Тогда вы с заказчиком можете остановиться на использовании такой CMS, как WordPress. В ее ядре – функциональный и качественный блог, в ней также можно добавлять отдельные страницы.

Чем больше сайт, тем серьезнее требования к контенту. Например, сейчас мы переделываем сайт для фестиваля искусств. Многие годы фестиваль остается привлекательным за счет подборки ценных видео– и аудиоматериалов, а также тысяч высококачественных фотографий, которые ежегодно снимают профессиональные фотографы

В сайтах с постраничной структурой предполагается, что контент-редакторы будут каждый раз отыскивать в архивах важный материал и перезагружать его заново на сайт. В данном случае это не лучший способ. Вместо этого мы создали отдельный медиасервер, со средствами поиска и маркировкой материалов. Все это для того, чтобы их было легко находить и вставлять в страницу. CMS готовит фотографии нужных размеров, включая те, что нужны для адаптивного дизайна. Изучив архивы материалов и, оценив их объем, мы поняли возникающие в связи с этим проблемы. Это позволило нам предложить такое решение.

Есть сайты, которые больше похожи на веб-приложения, неважно, система ли это электронной коммерции, или традиционные приложения. Эти системы ориентируются не на постраничную структуру (хотя они и могут включать в себя некоторые страницы), а на различные виды контента, требующие обновления.

Слишком часто в таких системах небольшие тексты (например, текст на кнопке) жестко кодируются в самом приложении. И поэтому только разработчик может изменять текст в призыве к действию. В идеале должно быть так, чтобы рядовые пользователи могли его редактировать.

A/B-тестирование

На некоторых сайтах, в особенности на тех, что продают товары, вам нужно тестировать разные версии страниц, чтобы узнать, какой контент, макет страницы или взаимное расположение блоков приводит к большему числу конверсий.

В зависимости от типа сайта конверсии¹¹ могут быть такие: покупка товара через сайт, подписка на получение пробной версии продукта, на e-mail-рассылку или заполнение формы. Если потребуется такой вид тестирования, как он будет осуществляться с помощью CMS? Если часть посетителей нужно отправить на одну версию страницы, а остальных – на другую, здесь не обойтись без стратегии. Об этом виде тестирования вы можете прочесть на сайте Smashing Magazine в исчерпывающей статье «Исчерпывающее руководство по A/B тестированию»¹² либо на [сайте СилаУма](#) (примеч. ред.)

¹¹ Конверсия – один из важнейших параметров эффективности сайта. Обычно конверсией называется совершение целевого действия. Показатель конверсии считается как отношение числа пользователей, совершивших целевое действие к общему числу посетителей сайта или целевой страницы. – Примеч. науч. ред.

¹² [Chopra, Paras. The Ultimate Guide to A/B Testing / *smashed.by/abtesting*](#).

Электронная коммерция

Добавить функцию электронной коммерции на сайт может оказаться простым делом. Возможно, достаточно будет всего лишь поставить несколько кнопок «Купить сейчас» платежной системы PayPal¹³. Но могут быть и трудности, например, если вам придется настраивать свой магазин на поддомене готовой платформы онлайн-магазина или разрабатывать свой собственный. Конечно, сегодня процесс продажи товара напрямую через сайт не надо усложнять. Но все же есть над чем поломать голову, когда вам предстоит сделать правильный выбор в зависимости от вида продукции. В этом разделе мы пробежимся по некоторым из вариантов. Несомненно, вам будет это на руку, если вы в первый раз внедряете функцию электронной коммерции – ведь это все равно что ступить на необитаемый остров.

Что продаем?

Может быть, в вашем онлайн-магазине продаются товары, которые отправляются покупателю по почте или с курьером? Или, возможно, товары, которые поставляются электронно, такие как электронные книги, музыка или ПО. Также транзакциями на сайте могут быть пожертвования или подписка на сервис. Если товары скачиваемые, то подумайте над тем, как они будут доставляться клиенту после оплаты.

Как покупать?

Один ли товар будет продаваться (например, электронная книга), или посетители смогут просматривать товары и добавлять разнообразные позиции в свою корзину? Можно ли будет выбрать товар по характеристикам (например, футболку по размеру и цвету)? Нужно ли делать категоризацию товаров, чтобы упростить их поиск? Товар будет ограничиваться одной категорией или его можно будет найти в нескольких? Пригодятся ли ярлыки и ссылки среди дополнительных или связанных товаров (скажем, позволяющие владельцу продвигать аксессуары к продукции)?

Будут ли на сайте специальные предложения – «При покупке одной вещи, вторая – в подарок!», «Скидка 20 %», «Две вещи по цене одной!», «Купите X, получите Y за полцены!»? Реализовать подобные предложения в созданной по заказу системе может быть достаточно непросто. А если вы используете коробочную CMS, то нужно знать, сможет ли она поддерживать их.

Дьявол (как и бюджет) в деталях. Из всех сайтов, над которым я работал, наиболее склонны к расширению рамок функционала онлайн-магазины. При планировании подумайте обо всех заданных выше вопросах. Конечно, есть много чего заманчивого и желание иметь как можно больше всевозможных функций велико. Но если вы создаете CMS самостоятельно и не хотите использовать коробочные решения, то лучше старайтесь разрабатывать более простые вещи. Ведь это значительно экономит вам время.

¹³ В России PayPal не распространена, и лучше использовать другие платежные системы. – Примеч. ред.

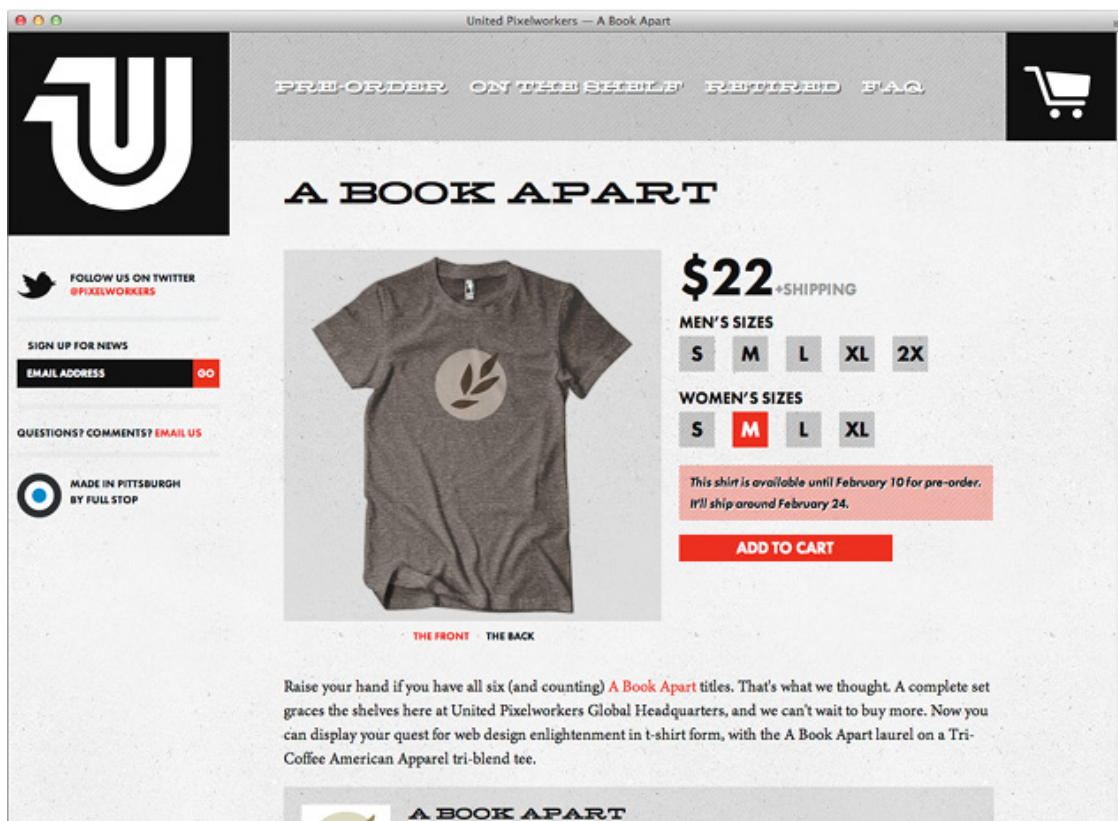


Рисунок 2.2. Вещь, которая, кажется простой, на самом деле может иметь ряд опций (например, футболка может быть женской или мужской и разных размеров)

Учетная запись и отслеживание заказа

У пользователей может быть опыт использования учетной записи (аккаунта) и отслеживания заказов. Возникают вопросы: обязательно ли пользователям создавать учетную запись? Смогут ли они отслеживать свои заказы и видеть их путь от «В обработке» до «Отгружено»? Аккаунт должен иметь такие основные функции управления, как восстановление забытого пароля или обновление контактных данных.

Как принимать оплату?

Вероятно, вам понадобится принимать платежи от покупателей по кредитным и дебетовым картам. Есть несколько вариантов, разных по сложности и цене. Обычно платежи обрабатываются с помощью платежной системы, аккаунта продавца и платежного шлюза и программы-посредника SaaS (программное обеспечение как услуга). Принимать платежи онлайн через платежную систему – легко. Плюсы здесь в том, что создать аккаунт несложно, а интеграция может быть различной: от простого варианта – встраивания кода кнопки на страницу, до полной интеграции с вашей системой.

На платежном рынке есть и другие игроки, но большинство из них действуют в немногих странах. Причина в том, что большая часть законов о процедуре оплаты в каждой стране имеет свою специфику. Если вы будете принимать регулярные платежи, то вам могут пригодиться Chargify и Recurly. И хотя сейчас они доступны только в США¹⁴, Stripe выглядит многообещающе как метод принятия платежей онлайн.

¹⁴ В России лучше воспользоваться услугами процессинговых компаний, действующих по законодательству РФ. – Примеч. ред.

Чтобы напрямую принимать платежи по карте, избегая посредников, вам нужно завести аккаунт онлайн-продавца. Это позволит принимать платежи по кредитке и переводить деньги на ваш банковский счет. Если у вас есть действующий банковский счет для офлайн-новых продаж, вероятно, вы не сможете воспользоваться им для онлайн-новых транзакций. Транзакции через Интернет не совсем безопасны, поэтому прежде чем начать выполнять их, свяжитесь со своим банком. Банк посоветует защитить свой платеж, в большинстве случаев через провайдера платежных сервисов PSP (payment service provider), иногда называемым платежным шлюзом.

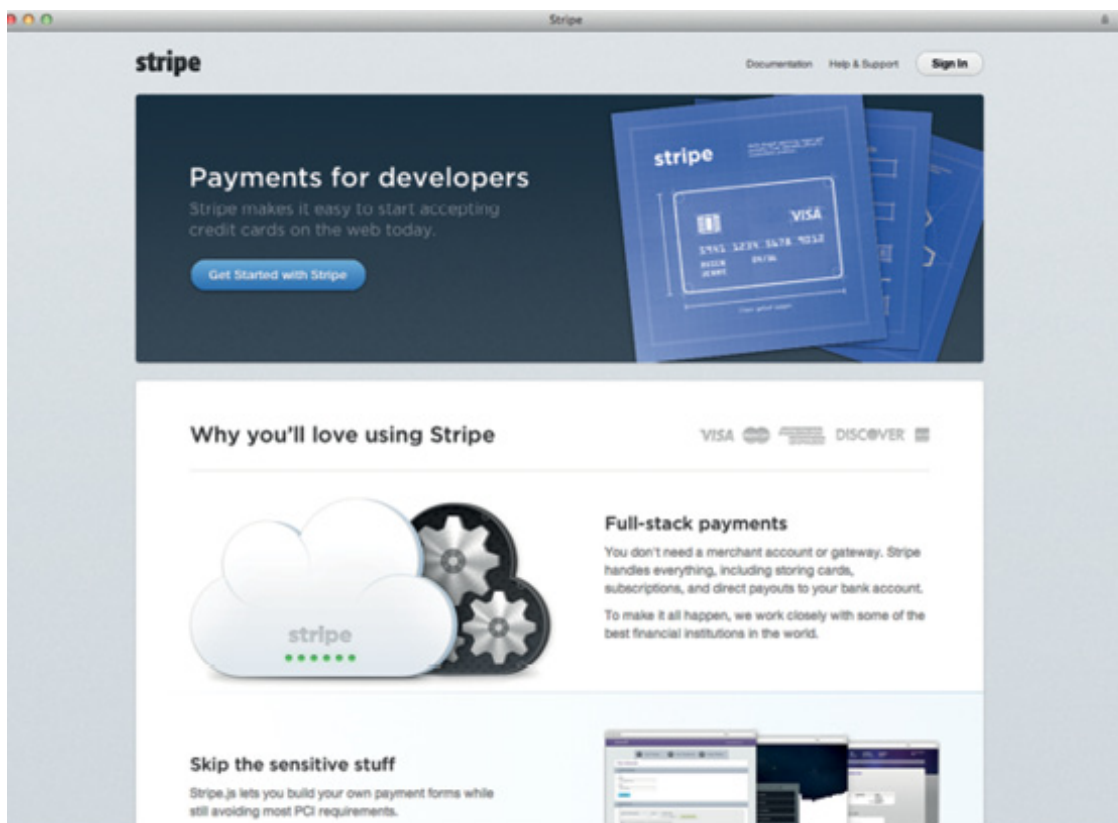


Рисунок 2.3. [Stripe.com](https://stripe.com) – это новый игрок на рынке, предлагающий простой способ приема платежей

Но что вы точно не должны делать, так это сохранять данные кредитной карточки, чтобы позже ввести их в терминал офлайн. Это противоречит условиям коммерческого договора. Поэтому, пока банк не даст вам свое письменное согласие и вы не выполните требования PCI DSS¹⁵ (о них мы расскажем позже), просто не делайте этого.

Платежный шлюз

Платежи через шлюз позволят вам принимать оплату от клиентов по карте, проверять ее номер и состояние счета, а после этого безопасно переводить платеж в ваш банк. Взаимодействовать со шлюзом можно двумя способами:

- **Через страницу платежей**

¹⁵ Payment Card Industry Data Security Standard – стандарт безопасности данных индустрии платежных карт. – Примеч. ред.

Пользователь переходит с вашего сайта на защищенную страницу провайдера платежного сервиса и вводит данные карты.

• **Через интеграцию по API¹⁶**

Пользователь вводит данные своей карточки на вашем сайте (на странице с сертификатом защиты, который поддерживает протокол SSL), и эти данные затем отправляются в шлюз. Ваш сайт в этом случае служит посредником; пользователь не в курсе, какие транзакции проводит банк, а видит их только на вашем сайте.

Преимущества интеграции платежной страницы в том, что ваш сайт вообще не имеет отношения к данным карты, поэтому вы не несете ответственность за безопасность клиента. Но есть один большой минус. Это то, что вы уже не сможете контролировать процесс оплаты, потому что на финальном этапе требуется собрать все данные и отправить их на платежный сервер. Кроме того, редко встречается возможность настроить страницу оплаты в своем фирменном стиле или даже просто вставить в нее логотип.

Необходимость уводить клиента с сайта очень беспокоит многих владельцев магазинов. Они боятся, что пользователи будут отменять транзакцию до того, как попадут на платежную страницу WorldPay или другого сервера. Но если отправлять клиентов на сайт известного банка для введения данных карты, то они будут больше доверять вам.

Скажу о себе. Когда на каком-то незнакомом сайте (например, мелкого розничного торговца) меня просят ввести данные моей карты, я начинаю нервничать и думать, а что с ними будут делать? Высветится номер моей карты в открытом виде на экране? Сохранятся ли ее данные в базе на сервере сайта?

Даже если страница имеет сертификат защиты и подтверждения о безопасности передаваемых данных, я все равно не имею понятия, что будет с моими данными, когда я нажму «Подтвердить». Но если в финале я окажусь на знакомой странице PSP, то буду спокоен за безопасность своих данных и за то, что они не попадут на сомнительный сайт. Я больше доверяю WorldPay, чем заурядному магазину виджетов.

Еще один плюс платежной страницы в том, что, если правила оплаты по карте изменяются, они будут обрабатываться на стороне провайдера PSP. Например, один из наших клиентов требовал 3D-Secure (на такой основаны технологии Verified by Visa и MasterCard SecureCode), чтобы принимать платежи Maestro. 3D защита требует от пользователей подтверждать платеж на странице своего банка, прежде чем он будет разрешен.

Если бы мы использовали API, нам нужно было бы изменить код, чтобы интегрировать 3D защиту. Но так как мы пользовались платежной страницей, мы просто уведомляли PSP, которая включала эту функцию для нашего аккаунта.

Все эти моменты подтолкнули владельцев многих сайтов к использованию платежных страниц. Они поняли, что нести ответственность за данные кредиток – только лишняя головная боль. Платежные страницы интегрируются со многими коробочными системами. После того как платеж сделан, пользователь перенаправляется обратно на ваш сайт, а специальный код позволяет идентифицировать пользователя и транзакцию и обрабатывать необходимые данные о совершенном заказе (например, поставить в базе данных отметку «Оплачено» или обеспечить клиенту доступ к цифровой продукции).

Преимущество полной интеграции API в том, что вы можете контролировать процесс оплаты от начала до конца, в том числе оформлять страницы оплаты в своем стиле. Но вы

¹⁶ Набор функций, предоставляемых сервисом для использования во внешних программных продуктах. Используется программистами для написания приложений для взаимодействия с исходным сервисом. Примером такого API могут служить, к примеру, Яндекс. Карты, которые можно встроить на свой сайт. – Примеч. ред.

также несете ответственность за безопасность данных пользовательских карт, и правила требуют, чтобы вы подтвердили, что используете передовые технологии.

PCI DSS

Стандарт безопасности данных индустрии платежных карт (The Payment Card Industry Data Security Standard (PCI DSS)) представляет собой совокупность 12 детализированных требований, которые распространяются на все компании, принимающие оплату по картам. Это относится не только к транзакциям онлайн. Обычный офлайн-магазин, который принимает платежи онлайн, должен тоже подчиняться требованиям PCI DSS для обоих способов оплаты: офлайн и онлайн.

Если вы принимаете онлайн-платежи через платежную страницу, но не получаете, обрабатываете или храните данные по картам, то вы можете заполнить сокращенный PCI DSS опросник (SAQ A), чтобы подтвердить, что ваш PSP совместим с PCI DSS. Если как средство интеграции вы используете API, то он полностью должен соответствовать PCI DSS (даже если вы не сохраняете данные карты) и включать ежеквартальные проверки безопасности, подтверждающих постоянное соблюдение требований.

В данном разделе мы не рассматриваем детали совместимости с PCI DSS, но если вы занимаетесь разработкой сайта, на котором платежи принимаются не через платежную страницу, тогда ознакомьтесь с ними или воспользуйтесь услугами тех, кто разбирается в данной теме.

Хранение данных карты

Я настоятельно советую вам не хранить данные карты на сервере клиента, даже в зашифрованном виде. Хранение данных требует совместимости с PCI DSS и поддержки сервера, а также Сети, в которых эти данные будут находиться в безопасности.

Если вам будет нужен доступ к ним, чтобы списать абонентскую плату, например вы можете поискать платежный шлюз, который предлагает сервис хранения данных. Если же вы собираетесь сохранять данные карты только для использования оплаты в один клик (как это делает Amazon), будьте осторожны! Вы в самом деле хотите взять на себя ответственность за данные вашего клиента? Хотите иметь дело с дополнительными и текущими расходами по поддержанию совместимости?

Валюта и местные налоги

Вполне возможно, что вам придется отчитываться по местным налогам, или НДС в Европе. Конечно, достаточно сложно разобраться в том, какие налоги нужны, но вы должны быть уверены, что ваша система обработает их правильно. Например, моя компания владеет скачиваемым продуктом, мини-CMS, которая называется Perch. Наша компания зарегистрирована в Великобритании, поэтому мы должны взимать налоги с британских покупателей. Еще нам нужно взимать НДС с покупателей из Евросоюза, пока у них есть действующий регистрационный номер плательщика НДС. Если покупатели из страны, которая не входит в состав Европейского Союза, мы не должны брать с них НДС. Итак, система должна уметь подтверждать номер налогоплательщика, а также правильно рассчитывать цены с или без НДС.

Большинство магазинов принимают оплату в одной валюте. Для того чтобы принимать платежи в разных валютах, чтобы посетители могли выбирать нужную им, смотреть соответствующие цены и вносить оплату, вам понадобится ввести нужную валюту в свой аккаунт продавца.

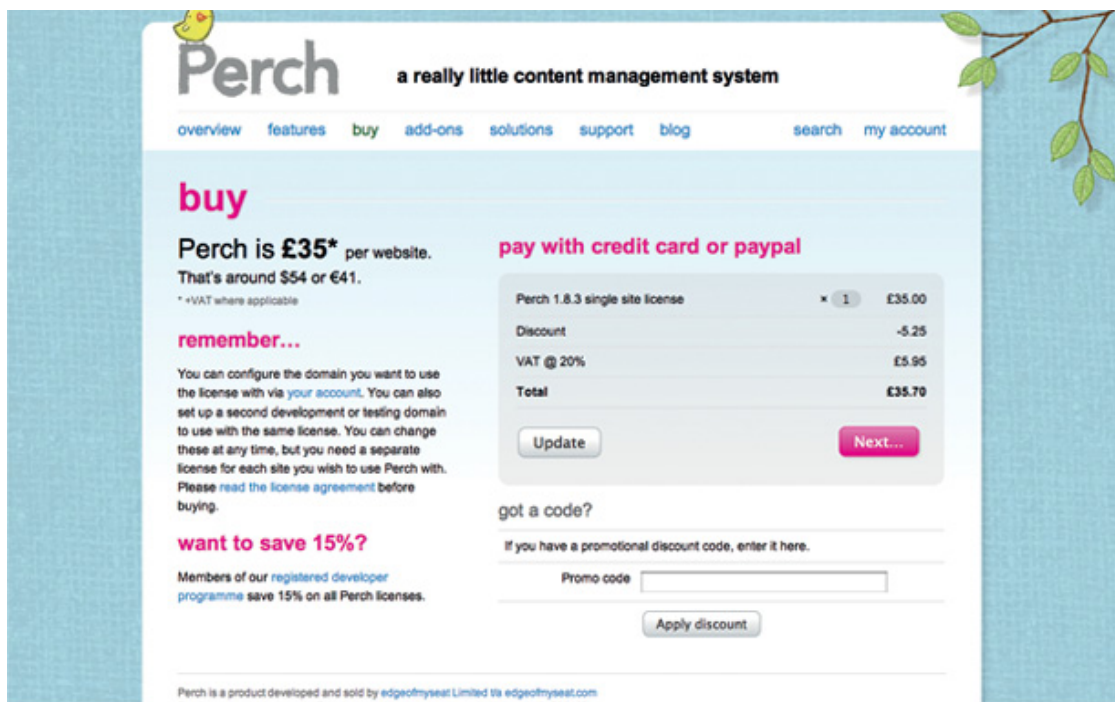


Рисунок 2.4. Платежная страница сайта. Включает НДС, скидку и приблизительную цену в долларах США. Несмотря на то что продается всего лишь один продукт, нужно принимать во внимание несколько величин

Другой вариант – это показывать цены в разных валютах, а оплату принимать только в местной. Вы можете обновлять курсы валют либо вручную, либо автоматизировать процесс с API. Если покупатели платят только местной валютой, то они должны быть в курсе, что стоимость показана чисто для информации и что реальная цена может слегка отличаться (из-за неустойчивых валютных курсов).

Обязательно посоветуйтесь с бухгалтером, когда будете иметь дело с деньгами, особенно, если вы принимаете платежи в иностранной или разной валюте. Если с самого начала знать, как обращаться с платежами и валютными курсами, то в будущем никаких проблем не возникнет.

О доставке

Если вы продаете физический товар, который нужно отправлять каким-то образом, вы должны включить расходы на транспортировку и, возможно, организовать отслеживание заказов. Так как вы продаете онлайн, то можно привлечь покупателей из других стран. Вам нужно решить, как рассчитать отправку в разные пункты и как вы будете доставлять товар: только в пределах вашей страны или в другие тоже.

Обычно онлайн-продавцы предлагают бесплатную доставку при заказе от определенной суммы. Также они практикуют доставку через разные службы доставки, такие как почта и курьерские службы (в зависимости от того, когда клиент хочет получить свой товар). Не забудьте учесть эти моменты, когда будете разрабатывать свой сайт.

Цифровые продукты

Покупатели ожидают, что они скачают цифровые продукты (такие как электронные книги, музыка и ПО) сразу же после их покупки. Доставка может быть оформлена в форме

ссылки или страницы для скачивания в своем профиле (вместе с лицензионным кодом если нужно).

Система должна обезопасить продукты до их приобретения и обеспечить в аккаунте клиента область, где они будут доступны (или как минимум отправлять ссылку на e-mail). Также можно генерировать код продукта. Опять же системы-посредники могут обеспечивать весь этот функционал в рамках оплачиваемых услуг.

Отчет и другие функции

Ваш клиент хочет обрабатывать заказы сразу же, как только они поступят, и вероятно, отмечать отправленные позиции, как только они будут обработаны. Возможность выгружать данные по заказам в CSV-файл, будет полезна как при организации e-mail-рассылки, так и при выгрузке платежных данных в офлайновую систему учёта продаж.

Вот ряд других вопросов, которые надо задать себе или заказчику:

- Нужно ли вам контролировать остатки через сайт?
- Как вы будете поступать с заказами, которые выполнены частично?
- Должен ли сайт генерировать счета и товарные накладные или это будет происходить в офлайне?

Интеграция с другими системами

Многие сайты не существуют автономно, а интегрируются с другими системами и сервисами. Интеграция реквизитов доступа (т. е. логина и пароля) от сети-посредника довольно стандартная процедура, и, возможно, вам будет нужно связываться с системой-посредником для контроля остатков и учета, особенно это касается магазинов электронной коммерции.

Несколько лет назад мы разрабатывали сайт для университета, который позволял студентам и персоналу обновлять свои данные и запрашивать определенные анкеты из офиса университета. По этой причине нужно было обеспечивать логинам защиту. Для обычного сайта мы бы использовали свои собственные идентификационные классы с CMS. Но здесь нам пришлось работать с уже существующими логинами студентов. Эта информация сохранялась в LDAP (облегченный протокол службы каталогов), поэтому наш сайт должен был идентифицировать пользователей с помощью университетского сервера LDAP.

Так мы создали новый интерфейс для нашей стандартной CMS-системы, который идентифицировался через сервер LDAP. Когда работаешь с такой системой-посредником, написать код ничего не стоит.

Много времени ушло именно на то, чтобы получить доступ к серверу LDAP и понять, как подтверждается вход в систему. Мы использовали свою собственную CMS и создали идентификационный интерфейс, что упростило процесс написания кода.

В следующий раз мы разрабатывали обычную систему электронной коммерции для магазина одежды, торгующего онлайн. В нем продавались разнообразные дизайнерские модели в ограниченных количествах. Часто рубашки определенного размера и дизайна были представлены лишь в одном экземпляре. Из-за того, что рубашки продавались одновременно и в офлайн-магазинах, и онлайн, нужно было вести обновляемый учетный реестр. Если вещь продавалась онлайн, ее сразу же надо было убирать с полок обычных магазинов, а если она была куплена в обычном магазине, то информация на сайте должна была сейчас же обновляться.

Вообще, установка этой функции вызывает явные проблемы (за вещь, которую только что продали онлайн, кто-то может уже расплачиваться на кассе обычного магазина). Луч-

шее, что мы могли сделать, так это обеспечить одновременную двустороннюю связь между сайтом и системой электронно-кассовых продаж.

EPOS¹⁷ была разработана другой компанией, и нам пришлось работать с ее разработчиками, чтобы скомпоновать две системы. Мы ждали по три недели, пока эти разработчики соберут нужные ресурсы и выполнят наш запрос.

С этим мы ничего не могли поделать, и очевидно, что это время необходимо было включать в расписание проекта.

Ясность в нашем общении и представление четкой и полной информации, тестирование интерфейса, помогли свести к минимуму риск задержки. Но если вы знаете, что без посредников не обойтись, уточните, как они работают и сколько времени им нужно, чтобы обработать тот или иной запрос. Вы сможете тогда планировать свои действия с оглядкой на них.

Очень важно с самого начала узнать все о системах-посредниках, которые вам нужны. Не забывайте об этих важных задачах. Это упростит вам выбор технологии и коробочного ПО. Также это необходимо учитывать и при планировании сроков проекта как в своей компании, так и у партнеров.

¹⁷ Electronic point of sale – оборудование для калькуляции и учета транзакций. – Примеч. ред.

Ограничения

Каждый проект имеет свои ограничения: во времени, в бюджете, в нехватке навыков и в несметном количестве других факторов. Прежде чем принять технические решения, вам нужно учесть их. Иначе эти ограничения встанут реальной преградой на вашем пути.

Бюджет

Когда будете планировать бюджет, учитывайте также и время. Наем разработчика или компании для создания сайта со стороны вносит ограничения в бюджет и временной график вашей работы над проектом.

Даже если посредник предлагает фиксированную сумму, она может колебаться в зависимости от того, сколько времени ему потребуется для работы. Разработка проекта своими силами также должна учитывать бюджет и время, будь то крайний срок его запуска или просто расчет времени на разработку.

Кроме чистых затрат на развитие, вам нужно будет учесть и затраты на посредников. Ведь мы все больше полагаемся на их помощь в обеспечении нашего сайта ресурсами. Хотите купить фотографии из фотобанка или шрифты? Затраты на хостинг могут увеличиться, если действующий провайдер не удовлетворяет техническим требованиям вашего нового сайта.

Если вы будете применять различные способы оплаты, то интернет-магазин понесет дополнительные затраты.

Программное обеспечение посредника может требовать оплату за лицензию, но это сократит время на разработку сайта. Вам, возможно, придется каждый месяц оплачивать услуги хост-системы. Так что включите в бюджет всевозможные затраты в начале проекта.

Выбор языка программирования

Возможно, ваша команда разработчиков или другое агентство по созданию сайтов имеют некий опыт в языке разработки. Он-то и определит базовый язык сайта, пока вам ужасно не захочется чего-то новенького. У хороших разработчиков всегда под рукой не один заранее заготовленный кусок кода и способы решения характерных задач на этом языке.

Но для того чтобы изменить язык на более «крутой», им потребуется серьезная переподготовка и много времени, потому что придется создавать новые библиотеки для базовых вещей. Это оправдает себя в том случае, если существующий язык мешает продвижению сайта (или из-за того, что мало кто из разработчиков знает его, или из-за того, что его развитие застопорилось). Предположим, ваша система создана на классическом ASP, и команда, которая поддерживает ее годами, знает этот язык. Однако он уже вытеснен ASP.NET (и не развивается настолько активно), поэтому строить на нем новую систему не имеет смысла. Кроме всего прочего, найти разработчиков, которые знают Classic ASP не так-то просто. Да и язык устарел настолько, что он не подходит для модернизации сайта.

Если вам придется столкнуться с подобной ситуацией, вашей команде нужно будет освоить новый язык для переделки сайта. Если агентства со стороны предлагают вам преобразование в Classic ASP, спросите их напрямую, почему они проталкивают эту технологию. Объясните им, наконец, что вы хотите сайт, который будет жить долго, а устаревшие приемы вряд ли продлят его дни.

Если вы сами создаете сайт, и вас тянет изучить определенный язык, тогда придется утрясти вопросы с отсрочкой сдачи проекта насколько это возможно. Все в ваших руках.

Постигать что-то новое всегда интересно, но не кидайтесь от одного к другому, только потому, что это модно. Лучше знать один язык, но досконально, чем несколько, но поверхностно.

Если проект готовится для заказчика, скорее всего, он станет ограничивать вас во времени и бюджете. Ограничения могут иметь под собой как техническую подоплеку (например, требование создать и разместить сайт под определенную серверную архитектуру), так и политическую (например, требование использовать ПО с открытым кодом).

Технические ограничения

Если компания держит свои сайты на собственных серверах, скорее всего, у нее есть своя серверная архитектура. В этом случае нужно точно знать, что вы будете устанавливать.

Например, сервер работает на Windows, а компания могла бы установить PHP; это спасло бы вас от головной боли при установке, потому что есть некоторые различия между PHP на Linux и PHP на Windows (в основном если сервер работает на IIS, а не Apache).

Если организация управляет собственными серверами, тогда выясните, будет ли у вас доступ к тестированию и развертыванию приложений. В некоторых проектах у меня его не было вообще, мне приходилось паковать код и данные и отправлять их IT специалистам организации для установки. Если это ваш случай, тогда скопируйте ПО хостинга для тестирования у себя. Этим вы сэкономите себе время, потому что тестирование и изменение на рабочем сервере – процесс не из легких.

Интеграция с другой системой – своего рода ограничение. И определенные сторонние приложения могут быть исключены, если вы не можете написать плагины для их интеграции.

Политические ограничения

Вы можете столкнуться с ограничениями, которые продиктованы внутренней организационной политикой и предпочтениями клиента. К примеру, любое стороннее ПО должно быть с открытым кодом или использоваться должны только технологии Майкрософт.

Термин «открытый/исходный код» часто непонятен. Когда люди решают, что программное обеспечение должно быть открытым, обычно они не имеют в виду лицензию на него или бесплатное пользование. Все, что они хотят, это иметь возможность модифицировать (изменять) код, если нужно. Это позволит обезопасить себя в том случае, если разработчик стороннего продукта по каким-то причинам пропал или откажется от развития и поддержки продукта. Если от вас требуют использования ПО с открытым кодом, проясните для себя, что под этим подразумевается. Многие коммерческие продукты имеют незашифрованных и доступный для модифицирования код, но при этом не имеют лицензии открытого исходного кода.

Иногда вы можете преодолеть политические ограничения, если приведете аргументы, почему какое-то решение лучше, чем то, которого от вас требуют.

Но они должны быть достаточно вескими. Ведь если клиент непоколебимо верит в какую-то технологию, то скептицизм с его стороны вам обеспечен.

Писать новый код или улучшать старый?

Решить, начинать ли преобразование сайта с полным обновлением серверного кода всегда непросто. Даже если существующая система имеет много проблем, может показаться, что вы выбросите на ветер кучу денег и приложите массу усилий, если будете заменять ее. В этой части мы разберем причины, по которым мы либо должны поддерживать уже существующую платформу, либо создать абсолютно новую.

Разработчик всегда подвержен искушению торжественно шагнуть вперед и создать что-то новое. Кому, в конце концов, понравится ковыряться в чужом коде? У всех нас свои методы работы; свои стандарты написания кода. Есть стандарты и системы, которые мы хорошо знаем и которым доверяем абсолютно. Но если мы выметим все подчистую из существующей системы и начнем все с нуля, мы рискуем потерять много полезного, что было в ней. И еще полбеды, если сайт предназначен просто для продвинутого управления контентом. А если вы решили переписать код сложной системы электронной коммерции, в которую в течение долгого времени вносили кучу различных доработок функционала?

Спросите себя, действительно ли существующая система не отвечает вашим требованиям настолько, что вы готовы полностью заменить ее, или же вам просто непривычно с ней работать.

Также не пренебрегайте опытом тех, кто пользуется системой. Если люди добавляют контент, обновляют продукцию или выполняют другие задачи в программе каждый день, то их нужно будет переобучать. Перенастройка и усовершенствование того, что есть, может быть даже поэтапное, избавит вас от этой необходимости.

Нельзя забывать и об ограничениях в сроках и бюджете. Начинать проект заново встанет заказчику в копеечку, да и времени на развитие будет затрачено больше. Внесение изменений в существующий код (рефакторинг) позволит вам распределить затраты, выкатывая обновления постепенно, по мере их готовности.

Но есть случаи, когда создание с нуля имеет смысл. Как мы уже говорили, программисты-новички будут рады заменить систему сайта на ту, которую они знают отлично.

Если вы нанимаете собственного разработчика или команду либо начинаете работать с третьим лицом и хотите долго и плодотворно сотрудничать с ними, то тогда резонно перейти на платформу, в которой они асы.

Мы уже сказали вам, что язык программирования может устареть, или сайт был основан на более не поддерживаемом ПО, или функции, которые вы хотите добавить, перегружают существующую структуру. Не выбрасывайте деньги на ветер!

Еще одна причина, по которой надо бы избавиться от существующей платформы, – это то, что она может ограничивать вас в выборе дизайна или изменении ее структуры.

Например, сторонняя платформа систем электронной коммерции устанавливает шаблонные дизайн и процесс оформления покупки, и вы не можете внести изменения, которые повысят продажи. Трата времени на поиск проблем и решений, для того чтобы не допускать подобного, выбивает из рабочей колеи всех, кто этим занимается.

В конце концов, прежде чем принять решение установить новую платформу или перенастроить старую, досконально изучите существующую систему и посмотрите, как она уживается с настоящими и будущими требованиями.

Собственная разработка или коробочное ПО?

Предположим, вы захотите заменить сервер частично или полностью. Тогда подумайте, разработать ли вам персональное решение, или положиться на коробочное ПО, или, может быть, применить их комбинацию. В этом разделе мы оценим преимущества каждого варианта.

Зачем разрабатывать свое?

Если у вас специфичные и редкие требования, то продвижение собственной разработки может стать лучшим вариантом. Стороннее ПО должно привлекать широкий рынок, чтобы разработчики не тратили свое время зря. К тому же так заманчиво добавлять множество характеристик, чтобы платформой заинтересовалось больше народу. Вся проблема в том, что разработка системы может закончиться тем, что в ней будет сотни ненужных вам вещей.

И если движок не довести до ума, его функциональные свойства будут замедлять скорость вашего сайта, болтаться как бесполезные элементы и увеличат накладные расходы.

Если у вашего сайта большой трафик и вы с самого начала в курсе, что должны оптимизировать каждую строку серверного кода и внешнего интерфейса, то создание собственного решения позволит учесть весь список требований. Моя личная проблема в том, что я недостаточно использую язык SQL. Некоторые приложения делают сотни ненужных запросов базы данных, которые существенно повышают нагрузку на сайт.

Может случиться, что лицензия на стороннюю программу не подойдет к приложению, которое вы разрабатываете. Может быть, и так, что в системе есть все, что нужно, но какие-то 10 % функций, которые вы добавляете, требуют довольно глубокого проникновения в ядро программы, проще говоря, ее взлома. Вряд ли кто-то из разработчиков софта даст вам добро на это. Вот она и проблема! Сложность будет еще и в том, чтобы усовершенствовать программу.

Лучше делайте свои изменения через официальный API. Если не сможете, все равно не взламывайте программу. Потому что то, что у вас получится, вы едва ли сможете поддерживать.

Плюсы в разработке и развитии собственной системы в том, что вы точно сможете подогнать ее под свои требования. Стандартная программа может быть написана великолепно, и так же великолепно удовлетворять требования заказчика. Но при этом утечка конфиденциальной информации все равно возможна. А теперь ваша задача — оценить, что лучше: использование готовых программ или создание своей, точно ориентированной на ваши задачи?

Выбираем коробочную платформу?

Коробочное ПО имеет явные преимущества. Вам не надо начинать с чистого листа. Вы получаете то, что уже находится в боевой готовности. Если вы ищете легкий способ управления контентом или ведения онлайн-магазина, а ваши требования достаточно просты, то наверняка одно из многих существующих в программе решений удовлетворит вас.

Есть бесплатные сторонние программы, а есть лицензионные, за которые приходится платить. У некоторых бесплатный только основной продукт, а дополнения — платные. Они добавляются быстро, поэтому заранее подумайте о том, что вам нужно для проекта. В одной системе может быть все включено в лицензионную оплату. Она, несомненно, обойдется вам

дешевле, чем бесплатная альтернатива, для которой нужно будет приобрести несколько коммерческих дополнений.

Не забудьте узнать, на какую поддержку вы можете рассчитывать. У многих бесплатных программ официальная поддержка стоит денег или вовсе не существует. Еще не помешает зайти на форум, где пользователи отвечают друг другу на вопросы. Посмотрите, сколько людей участвует в нем и как быстро приходят ответы.

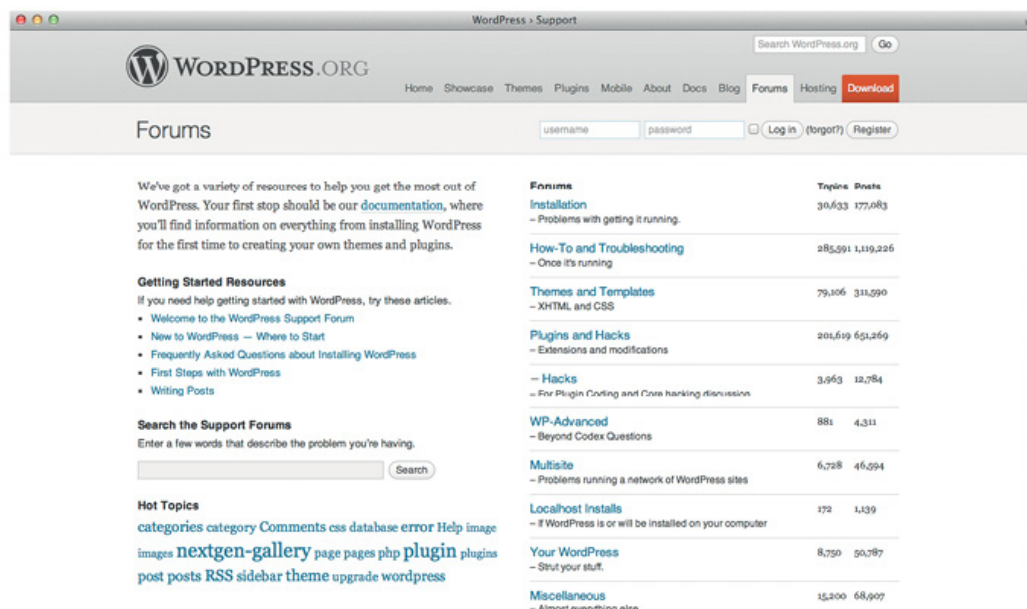


Рисунок 2.5. На WordPress оживленные форумы. Они для людей, которым нужно помочь в выборе правильного ПО

Если поддержка включена в стоимость, узнайте, в каком виде она предоставляется. Есть ли на нее ограничения или нет? Придется ли ждать? Существует ли общественная площадка поддержки и сразу ли персонал отвечает на вопросы? Twitter, кстати, тоже неплохой способ узнать о поддержке и общественном мнении о продукте. Просто спросите о нем и посмотрите, какие отзывы вы получите. Если поддержка хорошая, платный продукт может быть более стоящим, чем бесплатный вариант.

Облачные решения

Третий способ решения, который нам нужно обсудить, – это программное обеспечение как услуга (SaaS), известные также как облачное ПО. Есть системы управления контентом (CMS), которые, к примеру, используются в электронной коммерции. В них реализован различный функционал, начиная от простого «положить в корзину» и страницы оплаты и заканчивая полноценным интернет-магазином. Такое решение привлекательно тем, что вам не нужно ничего устанавливать, чтобы запустить и поддерживать сайт. Может лишь возникнуть необходимость настроить некоторые опции и выбрать шаблон.

Использование внешнего размещения – быстрый способ запустить в работу ваш сайт. Вообще, в проектах электронной коммерции, такое решение перекладывает проблемы безопасности и сложных функций на компанию, которая специализируется на этом. Но есть кое-что, над чем нужно подумать.

Возможно, заказчик должен платить за пользование сервером ежемесячно и, естественно, это нужно внести в бюджет. Проверьте, изменится ли цена в зависимости от вашего трафика и количества заказов.

Вы также не можете изменять работу такого ПО. Если разработчик вносит изменения, которые влияют на ваш бизнес, придется с этим смириться. Вы не сможете переписать какую-то часть так, как вы это делаете с программой, находящейся на вашем сервере. Это и понятно. К ее исходному коду вы имеете доступ.

Ваш сайт будет зависеть от деятельности разработчиков облачного ПО, которые крутятся в бизнесе и не перестают предлагать услуги заказчикам по доступным для них ценам. Программа, установленная на вашем сервере, даже если провайдер отойдет от дел, все равно остается у вас. Правда, вы потеряете поддержку, но по крайней мере сможете перейти на что-то свое абсолютно бесплатно. Если вы будете полагаться на внешний сервер, тщательно исследуйте его, чтобы убедиться, что он надежный, а компания – стабильная.

Бойтесь «евангелистов»!

Пользователи определенных продуктов порой проявляют свои странности. Вы это можете наблюдать на форуме, куда они заходят и наивно спрашивают: «А какая CMS лучше?»

По большому счету это бессмысленный вопрос. Он не прольет света на проблему, которую вы стараетесь решить. Читатель понятия не имеет, какой у вас сайт: крошечный с несколькими кусочками редактируемого текста или огромный трехязычный, вмещающий в себя тысячи документов.

И все равно кто-нибудь да расскажет вам о своей любимой CMS и в восторге станет доказывать, почему она лучшая из лучших. Этих людей мы называем «евангелистами»: они так любят софт, которым пользуются, что и представить себе не могут, что у кого-то может быть что-то другое. Я сам являюсь поставщиком CMS. Именно поэтому я иногда имею свою выгоду от «евангелистов» в среде наших пользователей. Но я и первый, кто в случае, если наше решение не вписывается в проект, скажет об этом прямо.

Бойтесь «евангелистов», когда вы изучаете стороннее ПО. Неважно, что это будет: целая платформа или простой скрипт. Задавайте специфические вопросы с учетом ваших требований. Тогда, скорее всего, в ответ вы получите более квалифицированное мнение.

Хостинг

Нас часто просят порекомендовать хороших хостинг-провайдеров и помочь с выбором хоста. Хостеров тысячи, но все они предлагают в конечном итоге серверы, которые могут мало отличаться друг от друга внешне, но по стоимости – в разы. Так что же выбрать? Может быть, просто хостера подешевле, у которого есть все, что вам нужно?

Дешевый хостинг – дорогой хостинг, или Скупой платит дважды

Многим хостинг-провайдерам удается сбывать свой продукт на рынке чисто из-за низкой стоимости. Можно достать хостинг за \$10 в год. Но то, что есть такой дешевый хостинг, еще не говорит о том, что надо пользоваться им. Он может в результате обойтись вам гораздо дороже, чем качественный хостинг с разумной, обоснованной ценой. Хороший хост стоит денег. А если заказчик толкает вас к дешевому провайдеру, то, надеюсь, что, прочитав этот раздел, вы вооружитесь неплохими контраргументами.

Веб-хост с оплатой \$10 в год, чтобы «сколотить» хоть какие-то деньги, должен разместить сотни сайтов на каждый физический сервер. Соответственно, это может привести к перегрузке, и какой-нибудь сайт с тяжелым трафиком застопорит остальные или даже выведет их в офлайн.

Google сейчас использует показатель скорости загрузки сайта для ранжирования выдачи¹⁸, таким образом, медленный сайт не только раздражает пользователей, но и ухудшает позицию выдачи.

Рассчитывать на техническую поддержку за \$10 в год было бы глупо. Ни один провайдер не сможет этого сделать за такие деньги. Если ваш сайт стал хуже работать и проблема коснулась пользователей, знайте, что вы можете быстро связаться со специалистом в приоритетном порядке. Услуги клиентам стоят денег. Так что низкая цена хостинга окупается за счет платы за поддержку.

Как выбрать хост

Ваш сайт имеет определенные требования. Например, чтобы запустить сайт на WordPress, вам нужен следующий набор:

- PHP версии 5.2.4 или выше,
- MySQL версии 5.0.15 или выше,
- модуль Apachemod_rewrite.

¹⁸ Google Webmaster Central Blog. Using Site Speed in Web Search Ranking / smashed.by/googlespeed.

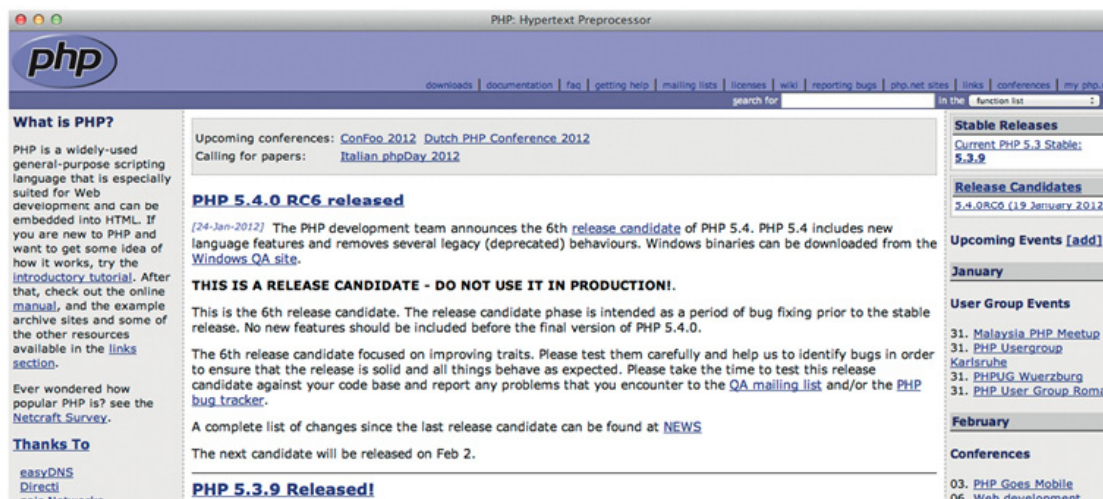


Рисунок 2.6. Проверьте последнюю стабильную версию PHP на php.net

Если вы используете серверный язык и базу данных, убедитесь, что ваш хост поддерживает их, а также и минимальную версию, которую требует ПО. Правило номер один: держитесь подальше от хостов со старыми версиями серверных языков. Вы можете ознакомиться с современной версией PHP на сайте php.net, проверив раздел «Стабильный релиз». На момент написания последняя версия – 5.3.9. Если хост поддерживает обновления, то его версия – 5.3, естественно, версии продукта не должны быть моложе, чем 5.2.4, как требуется для WordPress.

Хост может поддерживать серверы со старыми версиями PHP, если нужна поддержка пользователей, работающих на них. Но он должен быть в состоянии также их обновлять их по требованию или создавать новые аккаунты (учетные записи) для новых серверов.

Базы данных

Если вам нужна база данных для сайта (например, контент блога WordPress хранится в базе данных MySQL), посмотрите, предоставляет ли хост хотя бы одну. Если у вас свои скрипты, то внедрить каждый с собственной базой данных, возможно, будет легче. Поэтому проверьте, какое количество баз допустимо. Что касается скриптового языка сервера, выясните, совместима или нет версия базы данных того софта, который предлагает хост, с вашими скриптами.

Вам нужна поддержка e-mail-сервера?

Если вы хотите использовать один и тот же сервер как для e-mail, так и хостинга, узнайте, что он предлагает. Есть ли у него нужный вам web-mail? Сможете ли вы использовать e-mail с протоколами POP или IMAP? Легко ли использовать интерфейс для установки почтовых сообщений? Есть ли антиспам-фильтр?

Нужно ли вам размещать несколько доменов на одном сайте?

Некоторые хостинговые компании разрешают регистрацию только одного домена на вашем сайте. Если вы собираетесь завести несколько доменов, развейдите, возможно ли это вообще.

Принимаем решение

Существует тысячи хостинговых компаний. Выбирайте, но с умом! Ведь если вы подберете ненадежного хостера, последствия могут быть весьма серьезными для той компании, чей сайт является «сердцем» ее бизнеса. Так как же выбрать надежный хост?

Хорошим признаком служат рекомендации. Они очень действенны, особенно если человек, который их дает, пользуется хостингом не один год. Когда вы сотрудничаете с разработчиками сайтов, возможно, что они посоветуют надежного провайдера, с которым работали, и скажут, от кого надо бежать как черт от ладана!

Стоит поискать хостеров на Google. Если у людей были с кем-то из них проблемы, они сообщат об этом на форумах или в блогах. Если увидите много плохих отзывов о какой-то компании, держитесь от нее подальше. Еще один неплохой способ получения информации – это Twitter. В сущности, люди любят «пощебетать» о своих проблемах в реальном времени, даже если они не склонны расписывать их на форумах и в блогах.

Как я уже говорил, достать очень дешевый хостинг – не проблема. Прекрасно, когда сайт ваш личный или создается для друзей. Но если же он крайне важен для вас и для бизнеса заказчика, тогда помните, за что платите – то и получаете. Если хотите быстро справляться с проблемами, стоит заплатить немного больше.

Узнайте, как провайдер предоставляет поддержку. У некоторых хостов она осуществляется только через электронную почту, и это может быть для кого-то достаточным. Но ваш заказчик, возможно, захочет такого провайдера, которому он сможет позвонить, когда их почтовый сервер или сайт «упадет». Если вы знаете кого-нибудь, кто пользуется провайдером, который вы рассматриваете как вариант, узнайте, приходилось ли этим людям обращаться к нему за поддержкой и каков был опыт.

Как мы уже говорили, языки сервера нужно обновлять в целях безопасности и совместимости, как и настольные компьютеры. Если хостинг поддерживает очень старую версию языка, это говорит о том, что его серверы не современные. А это значит, что они уязвимы. Вы должны вникнуть в эти проблемы и попытаться установить скрипт с открытым кодом, который основан на современной версии языка программирования. Многие провайдеры дают гарантию на время работы, обычно в процентах: «Стопроцентная гарантия на время работы». Имеется в виду, что ваш сайт будет функционировать на 100 % в течение всего времени работы. Но не обольщайтесь и не рассчитывайте на многое!

Хостер обычно выплачивает компенсацию в случае, если сайт недоступен больше гарантированного времени, но вы сами должны контролировать такие моменты и предупредить хостера перед тем как работа сайта будет восстановлена. Нет ничего приятного, если ваш сайт вдруг в 2 часа ночи перестал работать. Также гарантия доступного времени работы распространяется на сам сервер, но не на каналы связи.

Когда я выбираю хостинг, я мало обращаю внимания на заверения о времени доступности серверов по словам самого хостера, я спрашиваю пользователей этого хостера об этом и о работе техподдержки. Будьте осторожны! Не нарвитесь на хостинг компании, которая занимается перепродажей серверов крупных поставщиков. Любой может купить сервер у крупного хостера и начать распродавать его дисковое пространство, при этом не имея малейшего представления о хостинге вообще. В этом случае, если ваш сайт упадет, вы столкнетесь с тем, что это посредник не в силах ничего с этим сделать. Он лишь может посоветовать вам обратиться напрямую к хостеру. С посредником, который болтался между вами и теми, кто действительно решит ваши проблемы, будет покончено.

	VM1000	VM2000	VM4000	VM8000	VM16000	VM32000
CPU, Xeon cores min. equiv.	1x 0.35GHz Xeon	1x 0.70GHz Xeon	2x 0.70GHz Xeon	2x 1.4GHz Xeon	4x 1.4GHz Xeon	4x 2.8GHz Xeon
Dedicated RAM	512 Mbytes	1024 Mbytes	2048 Mbytes	4096 Mbytes	8192 Mbytes	16384 Mbytes
RAID disk space	40 Gbytes	80 Gbytes	160 Gbytes	320 Gbytes	640 Gbytes	1280 Gbytes
Disk transactions/sec	10	20	40	80	160	320
Opt. 1: Unmetered connection	5Mbps 10:1	5Mbps 5:1	10Mbps 7:1	10Mbps 4:1	20Mbps 4:1	40Mbps 4:1
Opt. 2: Inclusive transfer	50GB (20Mbps)	100GB (20Mbps)	150GB (20Mbps)	250GB (20Mbps)	500GB (20Mbps)	1000GB (50Mbps)
IP addresses included	Up to 2	Up to 2	Up to 2	Up to 2	Up to 2	Up to 2
cPanel/WHM with Fantastico	\$12.50/mo	\$12.50/mo	\$12.50/mo	\$12.50/mo	\$12.50/mo	\$12.50/mo
Windows 2008 Server R2	n/a	\$7.50/mo	\$12.50/mo	\$20.00/mo	\$36.00/mo	\$65.00/mo
Uptime guarantee	99.995%	99.995%	99.995%	99.995%	99.995%	99.995%
Basic Port Patrol™ monitoring	Included	Included	Included	Included	Included	Included
Approx. lead time from payment	<20 mins	<20 mins	<20 mins	<20 mins	<20 mins	<20 mins
Cancellation notice period	7 days	7 days	7 days	7 days	7 days	7 days
Setup charge	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00
Monthly Price	\$18.50	\$25.00*	\$41.00*	\$79.00*	\$159.00*	\$329.00

* special offer, limited time only!

Easy multiple-site Web hosting with cPanel/WHM
If you are looking for a comprehensive, easy-to-use way to host one or many Web

What is a Miniserver VM Virtual Server?
Using Miniserver VM virtualisation allows us to divide a single powerful server

Рисунок 2.7. Хостинг компания Memset предлагает ряд личных (приватных) виртуальных серверных пакетов

Типы хостинга

Много лет виртуальный (разделяемый) хостинг считался единственно прибыльным способом объединять почти все большие сайты. Сайт на виртуальном хостинге разделяет сервер с сотнями других. Вам будет выделено свое ограниченное пространство, поэтому вы не сможете конфигурировать или устанавливать софт на сервер.

Как альтернативу виртуальному можно привести выделенный сервер – физический компьютер, который вы получите целиком. С ним вам будет намного проще, потому что вы сможете изменять его установки, а также его «родной» софт. Но это удовольствие не из дешевых, именно поэтому многим сайтам не нужны пространство и ресурсы выделенного сервера.

В последние годы появился новый вид хостинга. Называется он «виртуальный выделенный сервер». В плане управления операционной системой он соответствует выделенному серверу, но по факту вы делите один физический сервер с другими виртуальными. Плюсы такого виртуального хостинга в том, что вы можете работать с какой угодно версией операционной системы и ПО, которое вам по душе.

А это значит, что вы можете изменять сервер и устанавливать на нем сложные сайты.

Многие компании, предлагающие этот вид хостинга, по умолчанию прилагают панель управления, чтобы облегчить людям работу на сервере, в том числе и тем, кто никогда не администрировал такую систему раньше.

Две наиболее известные панели – это Plesk и cPanel (которые включают панель Web Host Manager). С помощью этих средств вы сможете установить новый сайт на сервере и актуализировать обновления, а также компоновать сервисы, доступные для индивидуальных сайтов. В сущности, имея виртуальный выделенный сервис, вы сами себе хозяева.

Виртуальные выделенные серверы также удобны для настройки показа сайта заказчиком, возможно на субдомене. Для дизайнеров, чье кредо – не передавать файлы до тех пор,

пока заказчик не оплатит их, такая настройка позволит предоставить на утверждение сайт полностью, прежде чем отправить его клиенту.

Виртуальный выделенный сервер дает компаниям максимальную гибкость. Но если у вас неспецифические требования, то для их выполнения, скорее всего, подойдет стандартный общий пакет хостинговых услуг. Он имеет соответствующие спецификации и надежную поддержку, а также постоянно обновляется.

Облачный хостинг отличается от виртуального и виртуально-выделенного. В нем ресурсы, необходимые для вашего сайта, могут быть разбросаны по другим серверам, и вы вольны в том, чтобы добавлять их или, если нужно, убирать. Этот вид хостинга подходит для приложений, которым иногда требуются мощные ресурсы, а иногда – низкие.

Хороший тому пример – продажа билетов. Наплыв пользователей на сайт наблюдается, когда в продажу выбрасывают билеты на какое-то популярное событие, но все остальное время трафик остается низким. С облачным хостингом у вас будет, когда нужно, доступ к дополнительным мощностям (конечно же, не бесплатно), а резервная мощность сохраняется круглый год.

Среда разработки

Данный раздел больше подойдет не тем разработчикам, которые работают в большой команде, а для дизайнеров, которые впервые взялись за полную переделку сайта. Мы поделимся своим опытом, а затем займемся непосредственно переделкой.

Неважно, переделываете ли вы работающий в данный момент сайт или просто дополняете его новыми элементами или интерфейсом, это не должно навредить пользователям. Даже если изменения ничтожны, никогда не обновляйте сайт, пока не протестируете изменения.

Один из методов, у которого много поклонников, это вести разработку, добавляя подпапки на существующем сайте. Не самая хорошая идея, однако! Когда вы будете готовы к запуску сайта и перенесете новые файлы выше, в корневой раздел сайта, относительные ссылки станут неверными. Все изображения могут «слететь» из-за этого. Придется вам тогда все восстанавливать. Согласитесь, не лучшее начало проекта!

Для разработки лучше настроить локальное оборудование, включая серверные скрипты и правильные пути к корню. Затем загрузите его на вспомогательный сервер, где вы будете проверять сайт на вебе, показывать его заказчику, наполнять контент и получать добро на дальнейшее существование.

Если вы дорабатываете «живущий» сайт или перекраиваете интерфейс пользователя, а код оставляете более или менее нетронутым, мой вам совет: экспортируйте базу данных и скачайте все файлы (об этом прочтете позже в статье «Перенос вашего сайта»). Это нужно для того чтобы настроить разрабатываемую и промежуточную версии сайта в точности, как на действующем сайте. Если база данных с рабочего сайта содержит конфиденциальную информацию о пользователях, лучше использовать псевдозаполнение данных. Не только для того чтобы защитить данные, но и во избежание их случайной отправки 10 000 пользователям из вашей локальной системы, когда будете что-то проверять!

Вспомогательный сервер

Локальный сервер устанавливается на ваш собственный компьютер или какой-то другой у вас в офисе, и вы используете его для разработки сайта. Когда будете готовы показать заказчику свою работу, вам придется переместить его куда-то, где к нему будет доступ с других сетей. «Куда-то» означает перемещение на вспомогательный сервер.

Он так же прост, как и поддомен, который указывает папку на вашем текущем сервере. Это позволит вам протестировать сайт на реальном сервере с заданной конфигурацией и всеми корректными путями.

Если вы создаете большое количество сайтов для заказчиков и собираетесь использовать те же настройки сервера (тот же язык, базы данных и проч.), вы можете взять дешевый виртуальный выделенный сервер и задать для каждого клиентского сайта свой субдомен для тестирования.

В этом случае желательно обезопасить директорию. Чтобы зайти на сайт, посетители должны будут ввести пароль. Это помешает случайным людям увидеть вашу «сырую» разработку и спасет от индексирования этого сайта Google.

При разработке сайтов со сложной структурой и разным контентом для заказчиков, мы обеспечиваем им доступ к его промежуточной версии, где они смогут добавить свой контент и тщательно изучить сайт. Потом мы перемещаем сайт, базу данных и прочие необходимые объекты на текущий сервер, что обеспечивает быструю и легкую миграцию новой версии сайта без ошибок.

Контроль версий

Мы не можем говорить о среде разработки, не упомянув о контроле версий. Даже если вы работаете в одиночку, вы должны его использовать. И это всегда полезно для тех, кто работает в команде. Ее члены смогут править файлы и не волноваться о том, что они «затирают» файлы друг друга. Вы всегда сможете вернуться к прежней версии, если что-то пойдет не так.

Если вы решили доработать сайт, а не переделать его полностью, ваш первый шаг – это импортирование всех файлов в одну из систем контроля версий. Так вы всегда сможете откатиться назад, если какие-то изменения чисто случайно затронут часть системы.

Даже когда вы работаете один, управление версиями защищает вас от самого себя. Например, от случайного удаления файла или быстрого изменения приложения. Я использую управление версиями, чтобы переносить работу с компьютера на компьютер, если это нужно. В конце каждого дня я фиксирую свою работу. Если моя дочь не пошла в школу и я работаю на дому, я могу взять текущие файлы и начать точно с того места, на котором я остановился позавчера вечером в офисе.

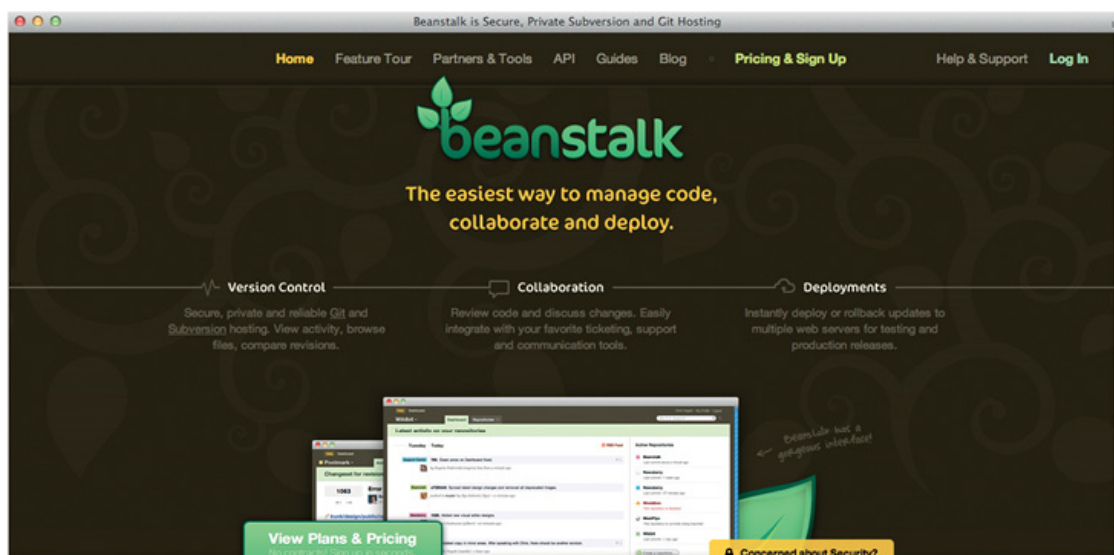


Рисунок 2.8. Beanstalk сервис для контроля версий

Существует несколько систем управления версиями. Вы, наверное, слышали о таких, как Subversion и Git. Обычно каждый выбирает, кому что нравится. Так как для выполнения работы вы можете менять сервер, то используя такие веб-сервисы, как GitHub или Beanstalk, вы можете быстро приступать к работе. Beanstalk имеет отличное руководство по управлению версиями на своем сайте¹⁹.

¹⁹ Beanstalk, «An Introduction to Version Control» / smashed.by/version-control.

Перенос вашего сайта

Вопрос, как переместить сайт с одного сервера на другой, мучает многих, особенно если сайт пользуется популярностью. Но если вы будете следовать рекомендациям ниже, то увидите, что сделать это возможно и без явного ущерба для посетителей. Этот процесс аналогичен переносу сайта с промежуточного сервера на основной.

Инструкции, данные ниже, предназначены для развертывания сайта на оборудовании обычного виртуального хостинга, с SFTP доступом. Если ваше оборудование сложнее, чем это, тогда у вас, вероятно, уже есть специалист, который поможет вам разобраться.

Установка нового оборудования хостинга

Для начала вы должны настроить хостинг. Проверьте, есть ли в нем все необходимые возможности, включая поддержку нужного вам серверного языка и базы данных.

Перемещение файлов на новый сервер

Для загрузки файлов сайта на новый сервер используйте SFTP. Новый сервер может предоставлять вам временный домен для тестирования сайта. Если нет, то я обычно временно создаю на сервере поддомен, пока я проверяю, все ли работает как надо.

Избегайте предварительного просмотра сайта с тильдой (~) или с вашим именем пользователя, иначе пути и ссылки побьются.

Перенос базы данных на новый сервер

Если вы работаете на приложениях не от Microsoft, то, скорее всего, вы используете MySQL. MySQL – это база данных, используемая для большинства приложений с открытыми «исходниками». Ее любят разработчики, которые работают с PHP, Python и Ruby on Rails.

Первое, что вы должны знать при работе с такой базой данных, как MySQL, это то, что в ней нет файла данных для скачивания и пересылки. База хранится внутри MySQL сервера. Чтобы получить данные, вам надо будет подключиться к нему. Многие серверы устанавливаются с PHPMyAdmin – веб-приложением, которое позволяет управлять базами данных через браузер.

Вы также сами можете загрузить и установить PHPMyAdmin. С ним вам будет удобно переносить базу данных MySQL между локальным, вспомогательным и рабочим серверами.

Настройка почтовых ящиков

Если ваша почта будет располагаться на новом домене, тогда установите почтовые учетные записи так, чтобы они были готовы к использованию, как только домен будет прикреплен к новому серверу.

Изменение сервера доменных имен, или переназначение записи домена

Если вы изменяете DNS домена на ваш новый сервер, то нужно изменить и серверы доменных имен. Если ваш DNS расположен где-то в другом месте, а вам нужно изменить

запись так, чтобы домен указывал на новый сервер, то измените лишь запись А. На это уйдет какое-то время. Пока новая запись между доменом и сервером не будет прописана по всему Интернету, продолжайте проверять почту со старого сервера день-другой, чтобы быть уверенным, что вы не потеряли ничего важного. Если прежний хост позволяет пользователям проверять почту в веб-интерфейсе без захода на домен, делайте так, пока почта не перестанет проходить через него.

Перенос сайта, работающего на базе данных

Для перемещения сайта, работающего на базе данных, нужно время: во-первых, чтобы запись DNS обновилась по всему Интернету, и во-вторых, для обновления кэширующих серверов. В течение этого промежутка времени, одни посетители могут направляться на старый хост, а другие – на новый. Если ваша база данных предназначена только для внутреннего управления контентом, и посетители не могут добавлять его, тогда просто попросите контент-менеджеров не изменять информацию на сайте до тех пор, пока они не будут уверены, что теперь они работают с сайтом на новом сервере.

В этом случае я проверяю сайт на копии базы данных и затем перемещаю базу непосредственно перед сменой доменных имен. Мне надо точно знать, что я перенес большую часть обновленного контента. Если посетители добавляют данные в базу (например, они делают заказы в онлайн-магазине), тогда вам ни в коем случае нельзя терять их при перемещении. Самый надежный способ – временно выключить сайт и разместить на нем страницу, с уведомлением, что идут технические работы.

Если так не получается, есть пара обходных маневров. Во-первых, сделайте перенос, как описано выше, а когда убедитесь, что DNS полностью переключился, сравните две базы данных и синхронизируйте записи от старых к новым. Если у сайта очень низкий трафик, а вы ожидаете только пару заказов или комментариев за это время, это сработает. Как вариант, вы можете сначала переместить базу данных. Пока будет настраиваться новый хостинг, чтобы внешний сайт мог соединиться с базой данных, вы можете переместить ее, а затем присоединить и старый, и новый сайты к базе данных в новых настройках сервера.

Если это ваш случай, то попросите вашего хостинг-провайдера (и разработчика, если он у вас есть) сделать все лучшим образом.

Куда дальше?

Мы пробежались по техническим вопросам, относительно переделки сайта или перехода на новую ступень развития. Не все в этом разделе пригодится именно вам, но все равно, обдумывайте свои технические решения, прежде чем принимать их. Даже если вы не один воплощаете их в реальность, избегайте спешки и не идите неверным, а значит и дорогим, путем!

Вспомните слова Пола Боуга из первого раздела: прежде чем «нырнуть с головой» в какое-то решение, сделайте домашнюю работу! Есть много способов достичь всего того, что вы хотите. А CMS или структура, которую вам пытаются продать, не всегда лучшее из того, что нужно вашему проекту. Всегда обращайтесь к своим исследованиям, бизнес-требованиям, имеющимся ресурсам и ограничениям. Тогда мудрое решение придет само собой! Все, что тщательно продумано, позволит сайту расти и развиваться.

Если вы полностью переделываете сайт, представьте, что вы это делаете в последний раз, и выбирайте такие решения, которые продвинут вперед ваш сайт и бизнес.

Об авторе



* * *

Рейчел Эндрю – веб-разработчик с навыками как в серверных языках, так и в верстке и разработке интерфейсов. Она написала множество книг, в том числе «Антология CSS: 101 основной совет, хитрость и уловка» (The CSS Anthology: 101 Essential Tips, Tricks & Hacks), четвертое издание которого было опубликовано в марте 2012 года. Рэйчел – основатель и управляющий директор edgeofmyseat.com (компания, занятая разработкой программного обеспечения), участвовала в создании «Пирч» (Perch), «действительно маленькой системы управления контентом». У Рэйчел есть личный сайт rachelandrew.co.uk, где она пишет о многих вещах, в том числе вебе и ведении бизнеса. Еще ее можно найти в Twitter под ником @rachelandrew.

О рецензентах



* * *

Харли Финкельштейн (р. 1983) – канадский гражданин, родившийся в Монреале, там же он получил ученую степень в области экономики. Также у него есть степень магистра в области права Университета Оттавы, города, где он живет сейчас. Он описывает Оттаву как холодный город с атмосферой провинции и радостью столицы. Помимо Интернета, он любит горные лыжи, бокс, готовить суши, диджеинг и красный цвет. Харли основал свою первую компанию в возрасте 17 и с тех пор работает в этой сфере. Сегодня он занимает должность руководителя платформы на Shopify, ведущей платформы электронной коммерции. На протяжении своей карьеры, Харли научился следовать за своими увлечениями и практиковать их каждый день. Его личный совет читателям: «Действуй энергично и находи изящные решения».



* * *

Райан Карсон – американец, живущий в Великобритании, отец семейства. Окончив Университет штата Колорадо со степенью в области компьютерных наук, он переехал в Великобританию в 2000 году. Он любит веб-технологии, кофе и фильмы. (Смотреть «Матрицу» в кино семь раз – это неправильно?) Он любит вовлекать и вдохновлять людей, и именно поэтому он увлеченно работает над мероприятиями для веб-сообщества. Он успешно построил и продал две компании и в настоящее время работает над третьей, Treehouse.

Погружение в HTML5

Автор: Бен Шварц

Рецензент: Рас Уикли

Если бы вы попросили меня рассказать вам о HTML5, я, пожалуй, начал бы с того, что сейчас ваша роль как веб-разработчика сильно изменилась по сравнению с тем, что было раньше.

Я буду обращаться к вам как к эксперту в области HTML (языка разметки), CSS (и всех ее вариаций в разных браузерах), JavaScript (и едва уловимой разницы между его воспроизведением в браузерах). Потом я перейду к теории дизайна, анимации, 3D, серверным технологиям и обработке звука.

Вы наверняка захотите узнать, зачем так много технологий собрано «под крышу» HTML, и, очевидно, удивитесь, почему вы решили взяться в первую очередь за веб.

HTML5 (как спецификация) «разбит» на мелкие кусочки, разбросанные по разным специализированным областям. Постарайтесь не раздражаться из-за этого.

Получив твердые базовые знания HTML, CSS и JavaScript вы сможете продолжить совершенствоваться самостоятельно и развить такие спецнавыки, которые другим и не снились. В сущности, когда вы овладеете не только узкоспециализированными, но и непрофильными навыками, вы станете просто бесценным и незаменимым для компании и коллег.

Для большей части этих новомодных фишек так называемый «передовой опыт» еще не накопили. Поэтому если вы хотите научиться чему-то действительно классному (и, возможно, прославиться в процессе), тогда самое время загрузить бета-версию браузера и начать эксперимент. Для этого, собственно, многие поставщики браузеров выпускают их бета-версии.

Ниже представлены, в частности, бета-версии так называемой «Большой пятерки» браузеров:

- У Google Chrome три версии релизов: «Beta» (для разработчиков) «Dev channel» (для разработчиков, которые хотят использовать возможности, созданные в течение последней недели) и «Canary» (так называемый ночной выпуск, полностью неопробованный). Вы можете скачать их по адресу smashed.by/chromedev.

- У браузера Safari одна версия: [Webkit \(webkit.org\)](http://Webkit.org).

- У Opera – версия «Next»: smashed.by/operadev.

- Firefox имеет «ночную» версию (smashed.by/ffndev) и пре-бета сборку

- Aurora (smashed.by/ffadev).

- Последний по списку, но не последний по значимости: версия Internet Explorer от Microsoft выходит в ручном режиме (и поэтому не является «ночной» сборкой): smashed.by/iedev.

Поддержка браузерами новых свойств выпускается в модульном порядке. А с производителями браузеров (такими как Google и Mozilla), выпускающими новую версию каждые 6–8 месяцев, номер версии теперь несет значительно меньше смысла, чем ранее.

Кто-то может провести параллель с тем, как разработчики вносят изменения на сайт. Да, у сайта есть версия, но она не важна для конечного пользователя. Поэтому, как разработчик, определитесь, с какими свойствами вам лучше написать свою историю.

Так как веб-технологии постоянно развиваются, мы должны иметь представление об их прошлом. К счастью, эту цель преследуют оба комитета по стандартам языка HTML. Так что расслабьтесь и дышите ровно, HTML5 не отдалит от вас пользователей и не обременит

работой. Какой бы DOCTYPE вы ни использовали, браузер пользователя отрендерит сайт в самом лучшем виде, насколько это возможно. Если вы используете новый элемент HTML5 вместе со старым DOCTYPE, все равно он будет отрендерен верно.

Здесь мы не будем говорить о WebGL, аудио и видео, устройстве API, Web sockets или SVG. На каждую тему уйдет по разделу, поэтому оставляю их вам на откуп. Вместо этого я расскажу вам все с самого начала. Мы разберем все важные базовые моменты, прежде чем двигаться дальше, к более сложным темам.

Мы откуда и куда?

HTML5 – это уйма всего. Мы уже далеко ушли от последней основной «версии» HTML. Сообщество по развитию гипертекстовых технологий (WHATWG) ссылается на нее как на «HTML: Живой стандарт» (от упоминания числа 5 отказались). А это значит, что у HTML нет версий. Как мы говорили, производители браузеров выбирают лучшие свойства для рендера, поэтому и отличается поддержка разных элементов разными браузерами.

WHATWG, W3C и «Компания»

Может быть, вы слышали о Всемирном консорциуме Сети (W3C). Мы уже упомянули о сообществе WHATWG. Оно было основано производителями браузеров Apple, Mozilla и Opera. Обеспокоенные тем, что W3C недостаточно занимается развитием HTML, они решили организовать свою группу.

Большая часть работы группы WHATWG пересекается с W3C, а в лицензии на спецификацию говорится, что «вам предоставляется лицензия на использование, воспроизведение и создание производных этого документа».

W3C в самом деле помогает в работе. Организация не разрабатывает стандарты, но охотно дает рекомендации. И хотя W3C финансируют большие компании – производители компьютеров и браузеров, она тем не менее специализируется на открытых стандартах, которые не ставят во главе угла какую-либо одну компанию.

Что же, как разработчик, вы можете быть уверены, что все новые разработки в HTML (особенно в области веб-приложений) создаются при значительной финансовой поддержке производителей браузеров (Webkit, Gecko и Opera) и со временем одобряются W3C.

Эти «странные» взаимоотношения привели к тому, что технология с лицензионной платой или являющаяся строгой собственностью у многих не пользуется популярностью. Браузеры соревнуются между собой сейчас столь же напряженно, как в самом начале.



Рисунок 3.1. Сайт Caniuse (smashed.by/ciu) демонстрирует, в каких случаях вы можете или должны использовать возможности HTML5

Знайте, какими функциями пользоваться

Современные разработчики должны понимать аудиторию, которую они обслуживают, уметь выбирать правильную технологию для работы и знать, каковы будут последствия, если какой-то функционал не поддерживается браузерами их аудитории.

Только маги и чародеи могут знать, насколько широко поддерживается какая-либо заданная функция. Если вы не относитесь к их числу, возьмите себе в помощь сайт When Can I Use²⁰. На нем находится список возможностей, которые поддерживаются в современных версиях всеми основными настольными и мобильными браузерами, а также перечень возможностей, которые будут представлены в будущих версиях. Он доступен для поиска и даже подключается к Google Analytics, чтобы показать вам, какие браузеры использует ваша аудитория. А теперь давайте погрузимся и посмотрим повнимательнее на основы HTML.

²⁰ smashed.by/wcai.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «ЛитРес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на ЛитРес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.