

Содержание:

Введение

Влияние глобальной компьютерной сети Internet на современный мир не имеет исторических аналогов. Его сегодняшний день – это начало эпохи электронного проникновения во все сферы человеческой жизни, это нечто большее, чем просто маркетинговая кампания, это основа новой философии и новой деловой стратегии.

Вполне логично предположить, что и с точки зрения рекламы продукции или услуги Интернет – наиболее значимый ресурс. Большинство современных людей пользуются Интернетом, как наиболее доступным источником информации.

Web-технология полностью перевернула представления о работе с информацией, да и с компьютером вообще. Оказалось, что традиционные параметры развития вычислительной техники – производительность, пропускная способность, емкость запоминающих устройств – не учитывали главного "узкого места" системы – интерфейса с человеком. Устаревший механизм взаимодействия человека с информационной системой сдерживал внедрение новых технологий и уменьшал выгоду от их применения. И только когда интерфейс между человеком и компьютером был упрощен до естественности восприятия обычным человеком, последовал беспрецедентный взрыв интереса к возможностям вычислительной техники.

Создание Web-сайтов является одной из важнейших технологий разработки ресурсов Internet. Хороший сайт, вбирая в себя всю полезную информацию, является лучшей визитной карточкой и коммерческой фирмы и образовательного учреждения, работая на них в любое время суток.

Проведя анализ нескольких сайтов обеспечивающих информационную поддержку предприятий малого бизнеса на предмет соответствия их поставленным целям, психофизиологическим требованиям и информативности, а так же корректности выбора средств разработки можно выделить следующие характерные недостатки:

- не соответствие сайта психофизиологическим требованиям, которое приводит к повышенной утомляемости или отвлечению внимания;

- увлечение анимацией, графическими изображениями и видео фрагментами, которые не несут смысловой нагрузки, автоматически влечет за собой уменьшение количества потенциальных посетителей web-сайта, так как повышает требования к пропускной способности Интернет-соединения и объем трафика;
- избыточность или недостаток представленной информации;
- некорректный выбор средств разработки, который влечет за собой необоснованное повышение стоимости готового продукта.

Таким образом, определение функциональных возможностей средств и языков для разработки сайта, а так же психофизиологических требований предъявляемых к текстовой и графической информации представленной на экране монитора представляется актуальной задачей.

К тому же сейчас, когда число онлайн-пользователей оценивается более чем в 500 миллионов человек (причем это число растет со скоростью приблизительно 7% в месяц), приходится искать пути использования новых технологий, которые могли бы оказать содействие в привлечении и удержании внимания пользователей, путешествующих по безбрежным просторам Web.

1. Особенности разработки web-приложений

Web-приложения имеют архитектурные, технологические и пользовательские особенности:

- web-приложения должны развертываться очень быстро, и требования к ним, вероятно, меняются в ходе этапа разработки;
- все более частой практикой в web-разработке становится замена документов на реальные прототипы приложения;
- очень раннее подключение конечных пользователей к тестированию и развитию web-приложения;
- в то время как при разработке обычных приложений создается только одна версия приложения, которая удовлетворяет всем заданным требованиям, для web-приложений все более частой (и желательной) становится публикация в web-сети приложения, которое еще не удовлетворяет всем заданным требованиям;

- получение ранних отзывов пользователей становится все более важным для разработки web-приложения, а цикл развития действительно начинает рассматриваться в качестве реальной возможности для улучшения web-приложения, а не как дополнительная трата времени на доведение приложения до рабочего состояния;
- работа по проектированию web-приложений делится на проектирование данных и проектирование навигации.

Разработка web-приложений включает специфические процессы:

- web-приложения являются специальным видом обычных программных приложений, и поэтому web-инженерия может рассматриваться в качестве специальной разновидности программной инженерии;
- разработка приложений для web-сети предполагает использование нескольких хорошо определенных правил и соглашений, которые предоставляют стабильно работающую, устойчивую и масштабируемую среду разработки и выполнения;
- существуют специальные процессы разработки, которые учитывают специфические возможности web-приложений.

Модель жизненного цикла современных web-приложений показана на рис.1.



Рис.1. Модель жизненного цикла современных web-приложений

Основное различие данной модели от модели разработки обычного ПО заключается в понимании "развертывания приложения" не как вида деятельности (работы), а как перехода между работами.

В web-сети развертывание приложения для его пользователей действительно не является трудоемкой работой: в связи с централизованной архитектурой обычных

web-приложений, отсутствием специального кода приложения на стороне пользователей, использованием браузера в качестве среды выполнения работа по развертыванию web-приложения в значительной степени облегчается и ускоряется.

Данная модель предлагает явное соединение работы по "поддержке и развитию" с "анализом требований". Именно этот переход характеризует модель, он создает второй цикл модели, который включает "анализ требований". Данный цикл называется "циклом развития". Первый цикл связывает "проектирование", "реализацию" "тестирование и развитие". Данный цикл называется "цикл построения и проверки". Такие два цикла соответствуют двум этапам, которые характерны для современных web-приложений: оффлайн-разработка и онлайн-разработка. Оффлайн-разработка заключается в создании web-приложения в среде разработки (например, Visual Studio) без использования реального web-сервера. Разработка в режиме онлайн выполняется после переноса web-приложения под управление реального web-сервера: создание виртуального каталога, файлов настройки; управление web-сервером.

2. Основные участники разработки web-приложений

В разработке web-приложений принимают участие различные специалисты, такие как аналитик приложения, архитектор приложения, администратор приложения, программисты, web-дизайнеры, авторы контента, менеджер контента, web-мастер.

Аналитик приложения на этапе определения требований собирает информацию о причинах (мотивах), вызвавших разработку данного приложения, и преобразует их в спецификацию (детальное описание) требований к приложениям. При этом он оценивает долговременные стратегические бизнес-цели и ограничения, а также преобразует их в кратковременные, конкретные требования к приложению.

На этапе проектирования приложения архитектор данных анализирует требования, которые связаны с контентом и данными прикладной области. На основе этого он разрабатывает концептуальную модель данных, которая организует данные в структуру, и представление, которое может быть доступно и может использоваться в приложении.

Архитектор приложения анализирует требования к функциям и сервисам, которые должны быть реализованы приложением. На основе этого он разрабатывает концептуальное решение логики работы приложения (описанной с помощью моделей, рисунков и языков описания спецификаций), которое разрабатывается на основе модели данных.

На основе созданных спецификаций программист (или разработчик) реализует решения, которые схематично описаны архитекторами данных, и приложения выполняют тестирование и отладку реализованных решений. В большинстве случаев программисты также управляют развертыванием приложения.

Администратор приложения является основным участником развертывания и развития приложения. Он отвечает за поддержку приложения, периодическое выполнение резервного копирования, управление сообществом пользователей, сбор отзывов пользователей.

Непосредственно в разработке web-приложений, помимо программистов и web-дизайнеров, также принимают участие следующие специалисты: wcb-мастер; автор контента; менеджер контента.

Программисты являются ответственными за программную реализацию логики работы приложения и работы с контентом (содержанием баз данных, наборами документов, изображениями, мультимедиа и т. п.). Они должны уметь описывать сложную логику работы приложений и сложные запросы для получения требуемых данных. Программисты также выполняют тестирование и отладку реализованных решений.

Web-дизайнеры отвечают за разработку внешнего вида (дизайна) страниц web-приложения, который является очень важным как для удобства использования, так и для привлечения пользователей. На основе нефункциональных требований, связанных с корпоративными графическими особенностями заказчика и известными стандартами коммуникации, определяется графический внешний вид приложения; структурируется контент и графика в схемы (структуры) страницы; выбираются подходящие свойства стилей (например, шрифты, цвета и размеры изображений). Web-дизайнеры обычно знают языки форматирования (HTML, CSS) web-страниц и умеют работать с программными системами проектирования и оформления страниц. Например, такими как Macromedia Dreamweaver или Microsoft Expression Studio. Кроме этого, они также могут иметь некоторые знания таких языков, как XML и XSLT. Но они не являются программистами и не имеют опыта в

описании логики работы приложения и составлении программ.

Для дизайнера не требуется знания программирования. Они должны уметь работать с программными инструментами оформления и разработки графических интерфейсов прототипов страниц.

Проблема разработки web-приложений во многом связана с тем, что программисты и дизайнеры совместно работают с одними и теми же web-страницами. Технологии разработки должны обеспечить максимальную независимость работы программистов и дизайнеров друг от друга.

Оба набора таких знаний и умений являются очень специализированными; они требуют большого опыта и критически важны при разработке web-приложений. Однако как нельзя требовать от дизайнера страниц создания хранимых SQL-процедур, также нельзя требовать и от программистов проектирования и оформления web-страниц. Обычно приложения, в которых программисты выполняли проектирование пользовательского интерфейса, имеют проблемы с удобством его использования. Аналогично приложения, которые были задуманы и построены web-дизайнерами, имеют проблемы с масштабированием и их трудно поддерживать.

Хотя некоторые программисты полагают, что они обладают умением, достаточным для выполнения всех задач, связанных с формированием динамических страниц, опыт показывает, что разработка сложных web-приложений является результатом совместных усилий дизайнеров и программистов. Для достижения максимальной производительности специалисты с разными умениями должны иметь возможность работать независимо над компонентами, которые больше всего подходят к их опыту.

Автор контента создаст новый контент (например, новые статьи, документацию, фотографии, сообщения блогов и т. п.), который добавляется и публикуется web-приложением.

Менеджер контента отвечает за сбор контента, оценку контента, обеспечение качества и конечную публикацию.

Web-мастер отвечает за поддержку и частично развитие web-приложений. Обычно каждое web-приложение доступно в web-сети, предлагает где-то (например, на странице контактов или в нижней части страниц) возможность связаться с web-мастером, чтобы, например, сообщить о неработающих ссылках и других

проблемах работы web-приложения.

Кроме этого, весь процесс разработки web-приложений включает реальных пользователей приложения, в особенности на этапе оценки удобства использования приложения и его развития во времени.

Понятно, что не все перечисленные специалисты участвуют в разработке любых web-приложений. Некоторые функции могут выполняться одновременно одними и теми же специалистами.

3. Современные методологии разработки web-приложений

Наиболее известными современными методологиями проектирования и разработки web-приложений являются:

- WebML – метод разработки и язык Web Modeling Language;
- WSDM – один из первых методов разработки web-приложений Web Site Design Method.

Методология WebML

Методология WebML является подходом к разработке web-приложений на основе модели. Основной вклад WebML состоит в разработке набора понятий, обозначений и методик для создания web-приложений, активно использующих данные, которые могут применяться командами разработчиков для поддержки всех видов работ жизненного цикла приложений – от анализа до развертывания и развития. Методология WebML объединяет традиционные приемы, хорошо известные разработчикам, такие как сценарии использования на языке UML и концептуальное проектирование данных с помощью модели Entity-Relationship, с новыми понятиями и методами для проектирования гипертекстов, которые являются важными для web-приложений.



Рис.2. Этапы модели разработки WebML

Процесс WebML является итеративным и инкрементальным. В данном процессе разные этапы повторяются и уточняются до тех пор, пока не будут получены результаты, полностью соответствующие требованиям к приложению. В связи с этим жизненный цикл продукта проходит в несколько итераций, на каждом из которых создаются прототипы или частичные версии приложения. На каждой итерации текущая версия приложения проверяется и оценивается, а затем расширяется или модифицируется для того, чтобы удовлетворять как уже собранным требованиям, так и вновь появившимся.

Анализ требований WebML состоит в выполнении следующих работ:

- Выявление групп пользователей, для которых разрабатывается приложение. Каждая группа описывает пользователей, имеющих один и тот же профиль или выполняющих одинаковые виды деятельности, с одинаковыми правами доступа к одним и тем же классам информации.
- Спецификация функциональных требований, которые связаны с функциями, предоставляемыми пользователям. Для каждой группы пользователей выявляются и описываются релевантные виды работ, которые должны выполняться приложением; каждый вид работ является согласованным набором элементарных задач.
- Выявление базовых информационных объектов, т. е. основных информационных активов, к которым может быть предоставлен доступ пользователям и которыми он может манипулировать.
- Декомпозиция web-приложения на представления сайта, т. е. разные web-страницы, спроектированные таким образом, чтобы удовлетворять хорошо описанному набору функциональных требований и требований пользователей. Для

каждой группы пользователей создается по крайней мере одно представление сайта, поддерживающее функции, выявленные для данной группы.

Проектирование приложения в WebML выполняется с помощью специальных концептуальных схем WebML, которые описывают структуру прикладной области и компоненты навигации на высоком уровне абстрактности, независимо от особенностей реализации. Проектирование web-приложения включает два вида работ:

- Проектирование данных: соответствует преобразованию базовых информационных объектов, выявленных в ходе анализа требований, в полную и согласованную схему данных.
- Проектирование гипертекста: создаются схемы представлений на основе ранее описанной схемы данных. Отличительной особенностью подхода WebML является упор на концептуальное моделирование спецификаций гипертекста.

Реализация web-приложений, разработанных с помощью WebML, выполняется с помощью специальных инструментов (например, WebRatio CASE tool), которые в значительной степени помогают проектировщикам создавать базы данных и web-приложения. Прежде всего они предлагают визуальную среду для рисования концептуальных схем данных и гипертекстов. Затем такие визуальные описания сохраняются в виде XML-документов, которые используются в качестве входных данных для WebML-генераторов кода.

4. Тестирование и оценка

Систематическое тестирование разрабатываемого приложения, благодаря доступности концептуальной модели и возможности преобразования модели для генерирования кода.

Основное внимание перенесено с проверки конкретных web-приложений к оценке правильности работы генератора кода. Если можно гарантировать, что генератор кода создает правильную реализацию для всех правильно составленных и имеющих смысл концептуальных схем (т. е. комбинаций моделирующих конструкций), то тогда тестирование web-приложений сводится к более решаемой задаче проверки правильности описания концептуальных схем.

Тестирование включает инновационные методы оценки качества разработанных web-приложений. Используется среда для управляемого моделью автоматического оценивания качества. Она поддерживает:

- статический анализ концептуальных схем (выполняется во время компиляции); он основывается на обнаружении в концептуальной схеме шаблонов проектирования и на их автоматической оценке относительно атрибутов качества, записанных в виде правил;
- динамический сбор данных об использовании приложения для автоматического анализа и сравнения с навигацией, описанной концептуальной схемой (выполняется во время выполнения); при этом выполняется автоматическая проверка и анализ усовершенствованного web- журнала, который дополняет обычный журнал web-сервера, следующими данными:
 - информацией об элементах и переходах, использованных пользователями;
 - информацией об объектах БД, опубликованных в просматриваемых страницах.

Для выполнения поддержки и развития также используется концептуальная модель приложения:

- запросы на изменения могут быть фактически преобразованы в изменения на концептуальном уровне (в модели данных или в модели гипертекста);
- изменения на концептуальном уровне затем преобразуются в реализацию.

5. Общие рекомендации по разработке web-приложений

Основной целью архитектора программного обеспечения при проектировании web-приложений является максимальное упрощение их структуры путем разделения задач на функциональные области, обеспечивая при этом безопасность и высокую производительность. Данные рекомендации помогут выполнить все требования и создать условия для эффективной работы web-приложений в обычных для них сценариях [2]:

- Выполните логическое разделение функциональности приложения. Используйте многослойную структуру для логического разделения приложения на слой

представления, бизнес-слой и слой доступа к данным. Это поможет создать удобный в обслуживании код и позволит отслеживать и оптимизировать производительность каждого слоя в отдельности. Четкое логическое разделение также обеспечивает более широкие возможности масштабирования приложения.

- Используйте абстракцию для реализации слабого связывания между слоями. Этот подход можно реализовать с помощью интерфейсных типов или абстрактных базовых классов; можно определить совместно используемую абстракцию, которая должна быть реализована интерфейсными компонентами.
- Определитесь с тем, как будет реализовано взаимодействие компонентов друг с другом. Для этого необходимо понимать сценарии развертывания, которые должно поддерживать приложение. Выясните, должно ли поддерживаться взаимодействие через физические границы (или границы процесса) либо все компоненты будут выполняться в рамках одного процесса.
- Используйте кэширование для сокращения количества сетевых вызовов и обращений к базе данных. При проектировании web- приложения используйте такие техники, как кэширование и буферизация вывода для сокращения сетевых вызовов между браузером и web- сервером и web-сервером и нижестоящими серверами. Правильно спроектированная стратегия кэширования, вероятно, единственный наиболее важный с точки зрения производительности аспект дизайна. ASP.Net предоставляет следующие возможности кэширования: кэширование вывода страницы, частичное кэширование страниц и Cache API.
- Используйте протоколирование и инструментирование. Необходимо выполнять аудит и протоколирование действий в слоях и уровнях приложения. Журналы регистрации событий могут использоваться для выявления подозрительных действий, что часто обеспечивает раннее обнаружение атак на систему. Не забывайте, что могут возникнуть сложности с регистрацией проблем, возникающих в коде сценариев, выполняющихся в браузере.
- Нужно хорошо продумать все вопросы аутентификации пользователей на границах доверия. При проектировании приложения необходимо предусмотреть аутентификацию пользователей при пересечении границ доверия, например при доступе к удаленному бизнес-слою из слоя представления.
- Не следует передавать важные конфиденциальные данные по сети в виде открытого текста. Если требуется передавать по сети конфиденциальные данные, такие как пароль или куки аутентификации, используйте для этого шифрование и

подписи данных либо шифрование с использованием протокола Secure Sockets Layer (SSL).

- Проектирование web-приложения нужно выполнять для работы с менее привилегированной учетной записью. Процесс должен иметь ограниченный доступ к файловой системе и другим ресурсам системы. Это позволит максимально сократить возможные негативные последствия на случай, если злоумышленник попытается взять процесс под свой контроль.

Заключение

С появлением web-технологии компьютер начинают использовать совершенно новые слои населения Земли. Можно выделить две наиболее характерные группы, находящиеся на разных социальных полюсах, которые были стремительно вовлечены в новую технологию, возможно, даже помимо их собственного желания. С одной стороны, это были представители элитарных групп общества - руководители крупных организаций, президенты банков, топ - менеджеры, влиятельные государственные чиновники. С другой стороны, это были представители широчайших слоев населения - домохозяйки, пенсионеры, дети.

Спектр социальных групп, подключающихся к сети Интернет и ищущих информацию в WWW, все время расширяется за счет пользователей, не относящихся к категории специалистов в области информационных технологий. Это врачи, строители, историки, юристы, финансисты, спортсмены, путешественники, священнослужители, артисты, писатели, художники. Список можно продолжать бесконечно. Любой, кто ощутил полезность и незаменимость Сети для своей профессиональной деятельности или увлечений, присоединяется к огромной армии потребителей информации во "Всемирной Паутине".

С развитием технологий гипертекстовой разметки в Интернете стало появляться всё больше сайтов, тематика которых была совершенно различной - от сайтов крупных компаний, повествующих об успехах компании и её провалах, до сайтов маленьких фирм, предлагающих посетить их офисы в пределах одного города.

Развитие Интернет-технологий послужило толчком к появлению новой ветки в Интернете - Интернет - форумов. Стали появляться сайты, и даже целые порталы, на которых люди со всех уголков планеты могут общаться, получать ответы на любые вопросы и, даже, заключать деловые сделки.

Создание сайта представляет собой маркетинговый шаг, направленный на создание информационного ресурса, который предоставит возможность для компании как удержать старых клиентов, так и привлечь новых.

Создание и разработка сайтов включает:

- утверждение первоначального технического задания на разработку сайта;
- определение структурной схемы сайта - расположение разделов, контента и навигации;
- web-дизайн - создание графических элементов макета сайта, стилей и элементов навигации;
- разработка программного кода, модулей, базы данных и других элементов сайта необходимых в проекте;
- тестирование и размещение сайта в сети интернет.

Существует множество средств для создания web-сайтов, но лишь некоторые из них способны предоставить разработчикам инструменты для решения подавляющего большинства стоящих перед ним задач. При разработке web-сайта из всех современных web-технологий, позволяющих создавать интерактивные web-страницы, необходимо выбрать наиболее подходящие для выполнения поставленных на первоначальном этапе задач.

Список использованных источников

1. Борисенко А.А. Web-дизайн. Просто как дважды два. – М.: Эксмо, 2008.- 320 с.
2. Бурлаков М. В. CorelDRAW 12. – СПб.; БХВ-Петербург, 2004. – 688 с.
3. Джамса Крис. Эффективный самоучитель по креативному Web-дизайну. HTML, XHTML, CSS, JavaScript, PHP, ASP, ActiveX. Текст, графика, звук и анимация. Пер с англ./Крис Джамса, Конрад Кинг, Энди Андерсон - М.: ООО "ДиаСофтЮП", 2005.- 672 с.
4. Дунаев В. Самоучитель JavaScript, 2-е изд. – СПб.: Питер, 2005. – 395
5. Зольников Д.С. PHP5. Как самостоятельно создать сайт любой сложности. – 2-е изд. стер. – М.: ИТ Пресс, 2007. – 272с.

6. Инькова Н. А., Зайцева Е. А., Кузьмина Н. В., Толстых С. Г. Создание Web-сайтов: Учебно-методическое пособие. Ч. 5. Тамбов: Изд-во Тамб. гос. техн. ун-та, 2005. – 56 с.
7. Мазуркевич А. PHP: настольная книга программиста /Александр Мазуркевич, Дмитрий Еловой. – Мн.: Новое знание, 2006. – 495 с.
8. Мак-Клелланд Дик. Photoshop 7. Библия пользователя. : Пер. с англ. – М.: Издательский дом "Вильяме", 2006. — 928 с.
9. Орлов Л. В. Web-сайт без секретов. / Л. В. Орлов. – 2-е изд. – М.: Бук-пресс, 2006. – 512 с.
10. Полонская Е.Л. Язык HTML. Самоучитель.: - М.: Издательский дом "Вильяме", 2005.— 320 с.
11. Создание Web-страниц и Web-сайтов. Самоучитель : [учеб. пособие] / под ред. В. Н. Печникова. – М.: Изд-во Триумф, 2006.— 464 с.
12. Якушев, Л. В. Начинаем работать в Интернет. Краткое руководство. – М.: Издательский дом "Вильяме", 2006. —128 с