

УДК: 004.9

АНАЛИЗ СОВРЕМЕННЫХ ТЕХНОЛОГИЙ, ПРИМЕНЯЕМЫХ ПРИ РАЗРАБОТКЕ САЙТОВ

Хараули, А. Я., Негребецкая, В. И.

ФГБОУ ВО «Курский государственный университет», колледж коммерции, технологий и сервиса, Россия, Курск, e-mail: alexander.kharauli@gmail.com, violetta-negrebel@mail.ru

Рассматриваются современные технологии, применяемые при разработке сайтов; этапы, инструменты и подходы к разработке, применяемые при web-разработке. Выделены основные проблемы выбора инструментов, и способов разработки. Проанализированы и обобщены современные тенденции развития web-разработки, описаны наиболее перспективные инструменты и сделаны выводы о реалиях web-индустрии.

Ключевые слова: web-разработка, инструменты web-разработки, подходы web-разработки, коммерциализация.

ANALYSIS OF MODERN TECHNOLOGIES USED IN WEBSITE DEVELOPMENT

Harauli, A. Ya., Negrebetskaya, V.I.

Kursk state University, College of Commerce, technology and service, Kursk, Russia, e-mail: alexander.kharauli@gmail.com, violetta-negrebel@mail.ru

Modern technologies used in the development of websites; stages, tools and approaches to development used in web development are considered. The main problems of choosing tools and methods of development are highlighted. The article analyzes and summarizes current trends in the development of web development, describes the most promising tools and draws conclusions about the realities of the web industry.

Keywords: development, development tools, development approaches, commercialization

Цифровая трансформация с каждым годом охватывает все больше различных сфер жизни и бизнеса. Если бизнес хочет быть конкурентоспособным обычных информационных сайтов уже недостаточно, необходимы мобильные и веб-приложения, которые не просто дают пользователям информацию, но и позволяют выполнять какие-то функции: получать или заказывать товары и услуги, предоставлять инструменты. Так, например, современным банкам уже недостаточно иметь сайт с информацией, необходимо иметь онлайн инструменты для своих клиентов, личный кабинет, где пользователь может управлять счетами, инвестициями, кредитами. Даже малому бизнесу, для повышения конверсии,

необходимы удобные инструменты, например запись на прием к врачу или парикмахерскую, бронирование столика в ресторане или детской игровой комнаты для проведения дня рождения.

Самим владельцам нужно своевременно получать информацию в удобном виде по состоянию своей компании, например, сбор статистических данных и аналитики по разным отделам производств, или производительность отделов. Часто каждый из отделов, по-своему собирает эти данные, и даже могут использовать разные инструменты и владельцу нужно тратить много личного времени, чтобы разобраться во всем этом, косвенно или прямо это может влиять на эффективность компании и в конечном счете на прибыль. Тут тоже поможет цифровая трансформация и разработка веб- или мобильного приложения.

Технологии не стоят на месте и постоянно развиваются, и то, что использовали несколько лет назад уже может быть неактуально сегодня, или то, что не могли сделать несколько лет назад уже стало реальностью. Существует больше количество современных инструментов, которые помогают создавать веб- и мобильные приложения быстрее и качественней. Выбор технологии программирования и разработки — отправная точка в реализации любого web-проекта. От этого зависит функциональность, удобство будущего ресурса, то, насколько эффективным будет продвижение сайта, а также целый ряд других параметров, от быстродействия до корректности отображения на разных устройствах. Так как технические требования к каждому ресурсу являются уникальными, «идеальной», универсальной технологии программирования не существует. В каждом случае решение подбирается индивидуально. Основываясь на современных веяниях в web-индустрии, можно выделить какие из технологий и инструментов будут востребованы в ближайшее время и почему на них стоит обратить внимание при создании современного веб-приложения.

Весь состав выполняемых при создании веб-сайтов работ можно разделить на три группы: web-mastering, web-design и web-programming.

Web-mastering – это технология создания веб-сайтов различного вида без использования программирования, чаще всего с помощью WYSIWYG-систем, позволяющих набирать на экране необходимый кадр, а при сохранении его – автоматически составляющих программу на языке HTML [2].

Примером такого редактора может служить программа от Adobe – Adobe Muse (рисунок 1).

Так же существуют и другие аналоги, такие как Microsoft Expression Web, Mobirise, Webflow, Macaw, Bootstrap Studio, Rapidweaver и прочие, но не стоит забывать и о онлайн конструкторах сайтов, Nethouse, Ukit, Tilda, Wix и множество других.

Основная задача web-mastering'a состоит в проектировании сайта, разработке такой его конструкции, которая способствует достижению основных экономических целей. Web-mastering включает в себя определение целей, которые должны быть достигнуты с помощью сайта, определение внешних условий, концептуальное проектирование веб-сайта, врезку его в информационную систему предприятия, заполнение его информацией, размещение в Интернете, исследование эффективности сайта и его эксплуатацию (т. е. ежедневный контроль, корректировку, обновление, модернизацию).

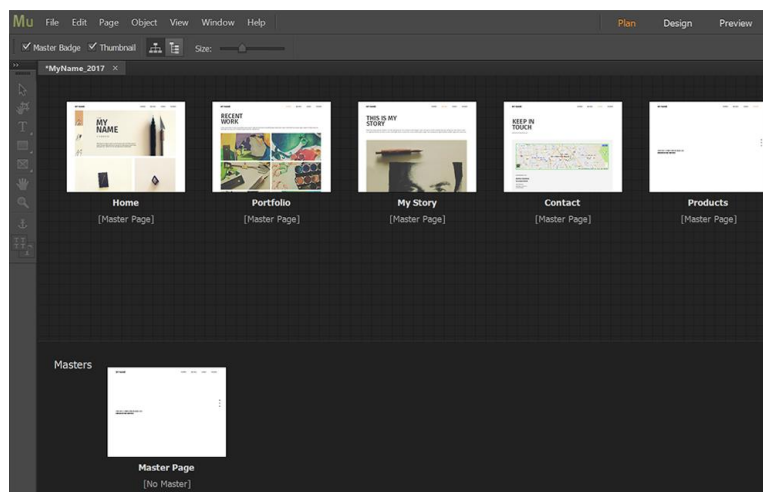


Рисунок 1 – Пример WYSIWYG редактора Adobe Muse

В основе технологии создания веб-сайта лежат причинно-следственные связи, начинающиеся с цели создания сайта. Когда цель известна, определяется целевая группа посетителей сайта, т. е. категория людей, являющихся потенциальными клиентами фирмы. Затем определяется, что именно может заинтересовать на сайте эту группу людей, чем их можно привлечь – на основе этой информации разрабатывается структура сайта. Затем разработанная структура реализуется в виде веб-страниц, связанных с помощью гиперссылок в определенную систему с учетом того, как удобнее перемещаться по сайту его посетителям – потенциальным клиентам фирмы (будут ли они всеядными и захотят ли ходить по страницам случайным образом или будут перемещаться по определенным маршрутам). Формируется необходимая информация, производится заполнение созданных веб-страниц информацией (набор, редактирование, сканирование графических материалов, вставка графики в текст и т. д.).

В создании сайта, как и в любом другом проекте, есть несколько этапов, а в частности их два, Front-end и Back-end разработка. Понятия backend и frontend и неотделимы друг от друга, потому что вместе представляют собой целый рабочий сайт, рассказать об одном без другого невозможно. Но эти этапы отвечают за разные задачи, которые представлены на рисунке 2.



Рисунок 2 – Задачи Front-end и Back-end разработки.

Если объяснять совсем просто, то фронтенд – это все, что отображается в любом браузере и относится к пользовательскому интерфейсу и публичной части сайта, а бекэнд – это программно-административная, т.е. служебная часть. Связь между этими компонентами крайне необходима, так как бэкенд поддерживает обеспечение приложений, запускающих фронтэнд, а также предоставляет доступ к базам данных, а значит и позволяет клиентам пользоваться сайтом. Рассмотрим их более подробно.

Front-end разработка — это создание клиентской части сайта. Front-end разработчик занимается версткой шаблона сайта и созданием пользовательского интерфейса. Обычно front-end разработчик — это мастер на все руки. Он просто обязан обладать талантом дизайнера, быть искусным верстальщиком и хорошим программистом. Современный front-end developer должен легко владеть html5, css3, JavaScript (и как минимум JQuery).

У каждого front-end специалиста есть свои наработки, которые он хранит в виде framework. Многие разработчики в работе пользуются популярными framework-ами, такими как: Twitter, Bootstrap, Foundation 3, Compass. Frontend разработка подразумевает создание механизма взаимодействия пользователей со сценариями, встроенными в HTML-код сайта. Все, что посетитель сайта видит, щелкает или использует для ввода или получения информации, - итог frontend разработки. Разработчик пользовательского интерфейса создает клиентское программное обеспечение, которое воплощает в жизнь дизайн сайта. Скрипты загружаются браузером, обрабатываются, а затем запускаются отдельно от сервера. Таким

образом frontend разработка - взаимодействие технологий и видения web-разработчика в равных частях.

Frontend разработчики - это мост между дизайнером и внутренним программистом, что означает, что они должны быть как творчески, так и технически подкованными специалистами. Окончательный дизайн становится живым сайтом за счет frontend. Frontend код в действии - код, который разбивает проект на компоненты, а затем предоставляет информацию и их функциональные возможности, которые создали backend разработчики.

В целом, они создают веб-приложение, которое подходит как можно больше к возможностям рабочего стола. Их внимание сосредоточено на пользовательском интерфейсе, и технология, которую они внедряют, будет зависеть от того, насколько хорошо он выполняет свою работу согласно данным по скорости, эффективности и плавной функциональности.

Как правило, услуги frontend веб-разработчика включают [6]:

- пользовательский интерфейс;
- приведение концепции дизайнера в жизнь с помощью HTML, CSS и JavaScript;
- производство, модификация и обслуживание веб-сайтов и пользовательских интерфейсов веб-приложений;
- создание инструментов, которые улучшают взаимодействие пользователей с вашим сайтом в любом браузере;
- внедрение адаптивного дизайна для мобильных версий сайтов;
- использование своего frontend опыта, взаимодействие с API и многое другое;
- поддержание управления документооборотом программного обеспечения с помощью инструмента управления проектами, такого как GitHub и средств запуска задач, таких как Grunt and Gulp Консалтинг по передовым методам SEO;
- тестирование сайта во время разработки для удобства использования и исправления любых ошибок.

Back-end разработка — это та часть сайта, которая невидима посетителю, но которая непосредственно отвечает за работу сайта.

Backend программист — это разработчик web-программирования, занимающийся написанием содержательной части программы, логики ее действий. Он программирует и создает серверную часть веб-сайта, обеспечивает вывод необходимого контента из базы данных в нужных участках веб-сайта, автоматизирует процесс сбора информации о пользователях, защищает сайт от взлома и всевозможных DoS и DDoS атак. В зависимости от поставленной задачи, использует также C# и ASP.NET или добавляет к этому Javascript, а иногда и другие языки (например, C++ и Python).

Визуальное решение сайта зависит от выбора дизайна макета и шаблонов, верстки и определенных скриптов, поддерживающих те или иные мультимедийные элементы. Серверное обслуживание же является результатом управления контентом и грамотного написания кодов и приложений. С какой-то точки зрения можно даже заключить, что бэкэнд гораздо важнее фронтэнда, ведь существование второго в большей степени зависит от существования первого, чем наоборот. Это весьма правильный вывод, т.к. бэкэнд может быть автономным сервисом, а фронтэнд – нет.

Чтобы создать хорошую техническую поддержку сайта необходимо владеть хотя бы одним из многих языков программирования. Для бэкэнда подходит множество языков. Стоит обратить внимание на эти самые распространенные: PHP; Python; Go; Ruby; Java.

Для более сложных проектов необходимо иметь в распоряжении такие инструменты как php, node.js, Go, Behat, Codeception, WebDriverIO, GraphQL, REST, React JS, Bootstrap, RabbitMQ, Gearman, Gulp.js, Deployer, Sphinx, git и другие. Рассмотрим их более подробно. Для создания проекта необходимо владеть более продвинутыми языками программирования, например php - это распространенный язык программирования общего назначения с открытым исходным кодом. PHP сконструирован специально для ведения Web-разработок и его код может внедряться непосредственно в HTML. Node.js - программная платформа, основанная на движке V8 которая транслирует JavaScript в машинный код. В основе Node.js лежит событийно-ориентированное и асинхронное (или реактивное) программирование с неблокирующим вводом/выводом. Go - компилируемый многопоточный язык программирования, разработанный компанией Google. Язык Go разрабатывался в качестве замены C: его высокая производительность почти сопоставима с языком Си, но более простой синтаксис дает возможность разрабатывать приложения гораздо быстрее. Процесс разработки предполагает наличие ошибок, и чтобы их вовремя устранить необходимо проводить тестирование кода, в этом помогут Behat - это фреймворк для тестирования, помогающий осуществлять разработку согласно концепции BDD. Codeception - фреймворк для тестирования. Есть интеграция с Symfony2, Lavarel4, Yii, Phalcon, Zend Framework. WebDriverIO — это клиент для сервера Selenium, работающий на Node.js и предоставляющий JavaScript API.

Для взаимодействия кода с другими средствами используются API, примером которых могут послужить такие API как GraphQL — это стандарт декларирования структуры данных и способов получения данных, предложенный и реализованный Facebook. REST - это архитектурный стиль, применяемый при разработке веб-сервисов основанный на протоколе http.

В процессе роста во многих проектах появляется необходимость решения ряда задач, связанных с очередями. Часто очереди сообщений используют как связующее звено между различными внутренними подсистемами. Несколько классических примеров:

- отложенная обработка пользовательских данных;
- передача статистики;
- сглаживание нагрузки на относительно медленные системы;
- выполнение периодических задач.

Существует несколько подходов к организации очередей [4]:

- использовать реляционные базы данных;
- применить существующие решения (RabbitMQ - платформа, реализующая систему обмена сообщениями между компонентами программной системы на основе стандарта AMQP. Gearman — сервер очередей и т.п.);
- написать свой код.

Так же не стоит забывать о системе поиска, для этого можно использовать например Sphinx - система полнотекстового поиска. Отличительной особенностью является высокая скорость индексации и поиска, а также интеграция с существующими СУБД.

После окончания создания проекта, происходит деплой и сборка проектов, в этом могут помочь Gulp.js - это потоковый сборщик проектов на JS. Это инструмент, который помогает автоматизировать рутинные задачи веб-разработки. Deployer — удобный и гибкий деплой приложений написанный на PHP. По итогу разработки всегда есть куда стремиться, а значит и проект будет развиваться, но чтобы не запутаться в различных версиях проекта можно воспользоваться системами управления версиями, например git - распределённая система управления версиями. Проект был создан Линусом Торвальдсом для управления разработкой ядра Linux, первая версия выпущена 7 апреля 2005 года.

Однако список на этом не заканчивается. Во фронтэнде есть четкий регламент, где HTML, CSS Javascript подчинены своим определенным функциям и не нуждаются в замене. В бэкэнде предоставляется большая свобода выбора, проблемы появляются редко, только тогда, когда выбранный язык по каким-то причинам не совмещается с языком расположения сайта. Также будут необходимо умение работать с фреймворками.

В заключении статьи хотелось бы отметить следующее: современные сайты должны быть в первую очередь удобными для пользователя, многофункциональными и иметь приятный дизайн, а будущее разработки сайтов напрямую зависит от развития веб-технологий и своевременного применения их на сайтах. Современные веб-технологии предоставляют разработчикам неограниченные возможности для реализации идей. Существуют автоматизированные системы создания сайта в виде WYSIWYG редакторов, но

они не способны раскрыть весь потенциал современных web-технологий, такие системы больше подходят для создания простых web-ресурсов. Для более сложных web-ресурсов, применяются иные, инструменты, позволяющие раскрыть весь потенциал web-технологий. Основными этапами разработки являются Front-end и Back-end разработка, при которых в ходе последовательного выполнения отдельных шагов разработки и тестирования, осуществляется полное завершение проекта. Проведя анализ современных технологий, применяемых при разработке сайтов, следует сделать вывод о том, что разработка сайта состоит из определенных этапов, что подразумевает собой использование различных инструментов, включая концептуальное проектирование, построение архитектуры проекта, дизайн, и маркетинг.

Современное программирование идет вперед, и роль кодов постепенно снижается, так как многие механизмы маршрутизируются. Вследствие этого и появляется спрос на работников, разбирающихся в работе более крупных систем, а не мелких, заменяемых машинами. В любом случае, профессиональные программисты рекомендуют своей головой разобраться во всех этапах написания служебной части сайта, более конкретно – сначала определенно стоит написать полный документ нужных команд «от руки». Это помогает вникнуть во все тонкости и добавляет опыта, который непременно поможет в случае непредвиденных ситуаций.

Список литературы:

1. Киселев, П.В. Прогрессивные веб-приложения: объединяющая технология для веб- и нативных приложений / П.В. Киселев // Политехнический молодежный журнал. -2020. -№ 2 (43). - С. 3.
2. Кузнецова, Л.В. Лекции по современным веб-технологиям [Электронный ресурс] / URL: <http://www.iprbookshop.ru/52151.html> (дата обращения: 05.01.2021).
3. Савельев, А.О. Проектирование и разработка веб-приложений на основе технологий Microsoft/ Савельев А.О., Алексеев А.А. – М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2019.— 419 с.
4. Сервер очередей [Электронный ресурс] / URL: <https://habr.com/ru/company/mailru/blog/216363/> (дата обращения: 05.01.2021).
5. Тузовский, А. Ф. Проектирование и разработка web-приложений: учебное пособие для среднего профессионального образования / А. Ф. Тузовский. — М.: Юрайт, 2019. — 218 с.
6. Что такое frontend разработка [Электронный ресурс] / URL: <https://webshake.ru/post/chto-takoe-frontend-razrabotka> (дата обращения: 05.01.2021).