

## Этапы разработки веб-приложения

Подготовила:  
Таринская Т.Г.

## Оглавление

|  |   |
|--|---|
| Этапы разработки веб-приложения .....    | 3 |
| Предпроектное исследование .....         | 4 |
| Проектирование и разработка макета ..... | 5 |
| Разработка фронтенда и бэкенда .....     | 6 |
| Тестирование и отладка .....             | 7 |
| Развертывание и поддержка .....          | 8 |
| СПИСОК ЛИТЕРАТУРЫ .....                  | 9 |

## Этапы разработки веб-приложения

**Веб-приложение** — это интерактивный сайт, который выполняет определённую функцию. При этом часть кода, обрабатывающего бизнес-логику, выполняется на сервере.

**Разработка web-приложения (web-разработка)** — это комплекс мер и действий по планированию и созданию сайта в сети Internet в зависимости от поставленных целей и задач.

Прежде всего, приложения можно разделить на кастомные (написанные кодом) и ноукод (собранные в конструкторах). Современные ноукод-редакторы, скажем, Webflow или Bubble позволяют создавать интерактивные решения — к ним можно подключить платёжную систему и сделать работающий интернет-магазин. Ноукод выбирают, потому что это быстро и дешево. Производительность таких приложений ниже, чем у кастомных — большой трафик они не выдержат. Поэтому если нужен полноценный сервис, с прицелом на большую аудиторию, стоит остановиться на кастомных решениях.

### **Этапы:**

1. Предпроектное исследование, определение целей и требований
2. Разработка структуры веб-приложения, прототипы страниц
3. Проектирование (UX-дизайн, UI-дизайн) и разработка дизайн-макетов
4. Разработка, программирование
5. Тестирование
6. Релиз веб-сервиса и сопровождение

## Предпроектное исследование

Первым этапом разработки веб-приложения является проведение исследования и создание плана. В этом этапе команда разработчиков собирает всю необходимую информацию о бизнесе компании, ее целях, аудитории и требованиях. Это включает в себя анализ конкурентов, изучение потребностей пользователей и определение основных функций и возможностей, которые должны быть включены в веб-приложение.

Сначала нужно понять, какую проблему вы пытаетесь решить. Это может быть упрощение процесса заказа товара, улучшение коммуникации между сотрудниками или повышение эффективности работы с документами. Важно выбрать проблему, которая действительно актуальна для вашей целевой аудитории и которую ваше решение может решить.

Изучите рынок, чтобы понять, есть ли уже решения для выбранной вами проблемы. Если да, то что именно они делают хорошо, а что можно улучшить? Какие функции уже реализованы в существующих решениях, а каких не хватает? Это поможет вам оценить конкуренцию и понять, какие возможности для улучшения существуют.

Данный этап является очень важным, поскольку именно на этом этапе можно выявить требования к будущей web-системе. Как правило, это осуществляется за счет выявления пожеланий заказчика, исследования целевой аудитории или же интервьюирования потенциальных пользователей будущего Web-приложения.

## Проектирование и разработка макета

На втором этапе команда разработчиков создает дизайн и макет веб-приложения. Макет — визуальное оформление основных элементов будущего web-приложения. Как правило, создание макетов осуществляется в какой-либо из графических программ.

Это включает в себя разработку пользовательского интерфейса (UI) и пользовательского опыта (UX), определение структуры и навигации приложения, а также создание прототипа, который поможет понять, как будет выглядеть финальный продукт. В этом этапе команда также определяет используемые технологии и инструменты разработки.

## Разработка фронтенда и бэкенда

Третий этап - это разработка фронтенда и бэкенда веб-приложения.

Фронтенд (англ. frontend) — это разработка пользовательских функций и интерфейса. К ним относится всё, что пользователи видят на сайте или в приложении, и с чем можно взаимодействовать: картинки, выпадающие списки, меню, анимация, карточки товаров, кнопки, чекбоксы, интерактивные элементы. Фронтенд включает в себя разработку HTML, CSS и JavaScript. HTML и CSS — это не языки программирования, а языки разметки. Они «рассказывают» браузеру, как именно должна выглядеть страница: где расположены блоки, какого они цвета, какого размера шрифт и картинки. От языков программирования HTML и CSS отличаются тем, что в них нет никаких функций, подсчётов, сравнений и других действий — они статично описывают внешний вид страницы. JavaScript — язык программирования. Он содержит функции и методы, которые позволяют получать информацию от сервера, отправлять её назад и выводить для пользователя, создавать интерактивные элементы, например кнопки и формы. Для frontend-разработки подходят библиотеки и фреймворки — например, React, Angular, Vue.js, Svelte. С ними разработчик решает часть задач при создании продукта: это управление состоянием приложения, разбивка элементов интерфейса на компоненты.

Бэкенд является серверной частью приложения и обрабатывает логику, хранение данных, а также взаимодействие с базой данных. В этом этапе команда разработчиков создает и тестирует функциональность приложения, а также реализует систему безопасности и возможные интеграции с внешними сервисами. В отличие от фронтенда, бэкенд использует гораздо больше языков программирования. Он может быть написан на любом языке. Сейчас наиболее востребованы Python, Java, Go, всё ещё популярен PHP. Иногда бэкенд пишут на C# и Ruby. JavaScript тоже подходит для бэкенда — его часто выбирают для изучения те, кто перешёл из фронтенда.

## Тестирование и отладка

После завершения разработки фронтенда и бэкенда веб-приложения команда разработчиков проводит тестирование и отладку приложения. Это включает в себя проверку функциональности приложения, устранение ошибок и несоответствий, а также оптимизацию производительности. При тестировании приложения используются различные методы, включая ручное тестирование, автоматизированное тестирование, а также тестирование с использованием различных устройств и браузеров.

Тестирование приложения представляет собой проверку того, как работает приложение на различных браузерах и платформах. Основная цель тестирования - убедиться, что приложение работает так, как задумано, и если присутствует какая-либо ошибка, ее можно легко устранить. Наиболее распространенные типы тестирования : функциональное тестирование, безопасность, совместимость браузера, производительность, отказоустойчивость, а также интерактивное и интуитивно понятное тестирование.

Основными этапами в процессе отладки являются: сбор информации о проблеме, отслеживание источника этой проблемы, отслеживание проектных кодов и связанных с ними ошибок.

## Развертывание и поддержка

Последний этап разработки веб-приложения - это развертывание и поддержка приложения. На этом этапе команда разработчиков размещает приложение на сервере, конфигурирует его для работы в сети и обеспечивает его доступность и безопасность. После развертывания команда обеспечивает поддержку и обновление приложения, включая устранение ошибок, внесение изменений и добавление новых функций.

Безопасность является одним из самых важных аспектов поддержки веб-приложений. Веб-приложения часто содержат конфиденциальную информацию пользователей, поэтому необходимо предпринимать соответствующие меры для защиты этой информации.

Поддержка производительности является важной задачей для веб-приложений. Пользователи ожидают отзывчивости и быстрой работы приложения, а поиск именно вашего приложения будет снижаться, если оно работает медленно.

Поддержка веб-приложений также включает обновление и развитие приложения. Технологии веб-разработки постоянно развиваются, а пользователи ожидают от приложения новых функций и возможностей.



## СПИСОК ЛИТЕРАТУРЫ

1. Васильев В.В. "Тестирование программного обеспечения: Учебник для вузов". М.: Высшая школа, 2021. - 180 с.
2. Николаев Н.Н. "Поддержка и обслуживание веб-приложений". М.: Инфра-М, 2020. - 150 с.
3. Лебедев Л.Л. "Проектирование интерактивных веб-сервисов: от идеи до реализации". СПб: БХВ-Петербург, 2019. - 270 с.