

ОБ АВТОРЕ

Баланов Антон Николаевич имеет большой опыт руководства и консультирования в сфере ИТ-технологий. Работал топ-менеджером в крупных компаниях — таких, как Industrial and Commercial Bank of China (КНР), Caravan portal (ОАЭ), Банк ВТБ, Сбербанк России, VK; руководил разработками сервиса Gosuslugi.ru. Имеет степень MBA IT (CIA) и сертификации Microsoft, CompTIA, ISACA, PMI, SHRM, ПВА, HRCI, ISO, Six Sigma (Master Black Belt). Преподавал в следующих вузах и учебных центрах: Российском университете дружбы народов, СберУниверситете, Институте бизнеса и делового администрирования и Центре подготовки руководителей и команд цифровой трансформации (на базе Высшей школы государственного управления РАНХиГС). Автор десятков книг и научно-практических публикаций в профессиональных изданиях. Является советником Российской академии естественных наук.

Широкая эрудиция и глубокие профессиональные компетенции автора в сфере ИТ-технологий позволили ему создать книжную серию «Айтишный университет», один из выпусков которой находится перед вами.

ОГЛАВЛЕНИЕ

Глава 1. Планирование разработки	10
Введение	10
Значение планирования веб-разработки и ее этапы	11
Определение целей и требований проекта	14
Разработка плана проекта, включая задачи, сроки и ресурсы	16
Управление рисками и анализ зависимостей	18
Заключение	20
 Глава 2. Анализ требований и спецификаций	22
Введение	22
Сбор и анализ требований пользователей и заказчика	23
Создание функциональных и нефункциональных спецификаций	26
Проектирование структуры данных и базы данных	28
Установление контрольных точек и проверка соответствия требованиям	31
Заключение	32
 Глава 3. Проектирование пользовательского интерфейса	34
Введение	34
Исследование пользовательского опыта и дизайна	35
Разработка прототипов пользовательского интерфейса	38
Создание информационной архитектуры и навигационной структуры	40

8 Оглавление

Оптимизация интерфейса для удобства использования и доступности	42
Заключение	44
Глава 4. Разработка серверной части приложения	47
Введение	47
Выбор технологий и фреймворков для серверной разработки	48
Разработка архитектуры серверной части приложения	51
Создание API и обработка запросов от клиентов	53
Реализация бизнес-логики и взаимодействие с базой данных	56
Заключение	59
Глава 5. Разработка клиентской части приложения	61
Введение	61
Выбор технологий и фреймворков для разработки клиентской части	62
Разработка пользовательского интерфейса с использованием HTML, CSS и JavaScript	64
Реализация клиентской логики и обработка событий	68
Интеграция с серверной частью приложения через API	71
Заключение	73
Глава 6. Тестирование и отладка приложения	75
Введение	75
Планирование и разработка тестовых сценариев	76
Выполнение модульного, интеграционного и системного тестирования	78
Отладка и исправление ошибок в приложении	81
Проверка производительности и безопасности	84
Заключение	86
Глава 7. Развёртывание и поддержка приложений	88
Введение	88

Выбор платформы и инфраструктуры для развертывания приложений	89
Создание процесса развертывания и автоматизации.....	91
Поддержка и обслуживание приложений, включая резервное копирование и мониторинг	93
Заключение	96
 Глава 8. Управление изменениями и обновлениями	 98
Введение.....	98
Управление версиями и контроль изменений в коде.....	99
Планирование и реализация обновлений приложений.....	101
Управление рисками и минимизация проблем при обновлениях.....	105
Тестирование и проверка совместимости после обновлений.....	108
Заключение	111
 Глава 9. Оптимизация и масштабирование приложений.....	 113
Введение.....	113
Анализ и оптимизация производительности приложений....	114
Масштабирование инфраструктуры для обработки растущей нагрузки	116
Использование кэширования, шардинга и других методов оптимизации	119
Мониторинг и улучшение производительности приложений	123
Заключение	125
 Глава 10. Документация и обучение для разработчиков.....	 128
Введение.....	128
Создание документации для разработчиков и пользователей.....	129
Проведение обучающих сессий и тренингов	132
Обмен знаниями и передача опыта в команде разработки	135
Управление знаниями и обновление документации	138
Заключение	141

ГЛАВА 1

ПЛАНИРОВАНИЕ РАЗРАБОТКИ

ВВЕДЕНИЕ

Глава 1 посвящена планированию разработки веб-проектов. Планирование является важным этапом веб-разработки, поскольку оно позволяет нам определить цели проекта, установить требования и разработать стратегию выполнения задач.

Мы начнем с рассмотрения значения планирования веб-разработки и его роли в обеспечении успешного завершения проекта. Планирование помогает нам организовать работу, устанавливает рамки проекта и обеспечивает эффективное использование ресурсов. Мы изучим различные этапы планирования, включая определение целей и требований проекта, разработку плана проекта и управление рисками.

Определение целей и требований проекта является важным шагом в планировании разработки. Мы узнаем, как проводить анализ потребностей пользователей, определять функциональные и нефункциональные требования, а также формулировать конкретные цели и ожидания проекта. Четкое понимание требований и целей помогает нам определить необходимые ресурсы и планировать задачи.

Разработка плана проекта является ключевым этапом планирования разработки веб-приложений. Мы обсудим методы создания плана проекта, включая определение задач, установление сроков выполнения и распределение ресурсов. Мы также рассмотрим важность управления зависимостями между задачами и разработку графика выполнения проекта.

Управление рисками и анализ зависимостей является неотъемлемой частью планирования разработки. Мы рассмотрим методы и инструменты для идентификации, оценки и управления рисками проекта. Кроме того, мы изучим анализ

зависимостей между задачами, чтобы эффективно планировать последовательность выполнения задач и управлять зависимостями.

Планирование разработки веб-проектов является важным этапом, который определяет успех проекта. Четкое определение целей, требований и разработка плана позволяют нам эффективно управлять ресурсами, сроками и рисками проекта. Понимание этапов планирования и использование соответствующих методов и инструментов помогут нам достичь желаемых результатов и успешно завершить веб-проекты.

ЗНАЧЕНИЕ ПЛАНИРОВАНИЯ ВЕБ-РАЗРАБОТКИ И ЕЕ ЭТАПЫ

Планирование является важным этапом веб-разработки, который помогает определить цели, требования и стратегию для успешного создания веб-приложения. Правильное планирование позволяет оптимизировать процесс разработки, улучшить качество и эффективность работы, а также уменьшить риски и непредвиденные затраты. Рассмотрим основное значение планирования веб-разработки и его этапы.

1. Определение целей и требований

Первый этап планирования — определение целей и требований проекта. Здесь необходимо ясно сформулировать, какие задачи должно решать веб-приложение и какие функциональные возможности оно должно предоставлять. Также важно учесть требования пользователей, бизнес-задачи, бюджет и сроки выполнения проекта.

2. Исследование и анализ

На этом этапе проводится исследование рынка, анализ конкурентов и аудитории. Это помогает определить особенности и требования целевой аудитории, а также выявить лучшие практики и тенденции веб-разработки. Исследование также включает анализ технических требований, выбор технологий и платформы разработки.

3. Создание структуры и макетов

На этом этапе создается структура веб-приложения и разрабатываются его макеты. Структура определяет иерархию страниц, навигацию и взаимосвязи между различными элементами приложения. Макеты позволяют визуализировать дизайн и расположение элементов на страницах.

4. Разработка функциональности

На этом этапе разрабатывается функциональность веб-приложения. Это включает программирование, создание баз данных, интеграцию с API и другие технические аспекты. Разработка функциональности основана на требованиях и спецификациях, определенных на предыдущих этапах планирования.

5. Тестирование и отладка

После разработки функциональности проводится тестирование и отладка веб-приложения. Это позволяет выявить и исправить ошибки, проверить работоспособность и соответствие приложения требованиям. Тестирование может включать модульные тесты, интеграционное тестирование, функциональное тестирование и тестирование производительности.

6. Запуск и мониторинг

После успешного тестирования веб-приложение готово к запуску. На этом этапе выполняется развертывание приложения на сервере и настройка его работы. Также необходимо установить механизмы мониторинга производительности приложения, чтобы обнаруживать и исправлять возможные проблемы в реальном времени.

Таблица 1.1

Этапы планирования веб-разработки

<i>Этап планирования</i>	<i>Описание</i>
Определение целей и требований	Определение целей проекта, бизнес-задач и требований к функциональности и дизайну веб-приложения.

<i>Этап планирования</i>	<i>Описание</i>
Исследование и анализ	Исследование рынка, анализ конкурентов и аудитории, выбор технологий и определение технических требований.
Создание структуры и макетов	Создание иерархии страниц, навигации и разработка макетов, визуализирующих дизайн и расположение элементов.
Разработка функциональности	Разработка программного кода, баз данных, интеграция с API и другие технические аспекты веб-приложения.
Тестирование и отладка	Проведение тестирования функциональности, отладка и исправление ошибок, проверка соответствия требованиям и работоспособности.
Запуск и мониторинг	Развертывание приложения, настройка его работы на сервере, установка механизмов мониторинга производительности.

Пример.

Предположим, что у вас есть проект по созданию онлайн-магазина. Ниже приведены примеры этапов планирования веб-разработки для этого проекта.

1. *Определение целей и требований.* Цель проекта — создать функциональный и привлекательный онлайн-магазин, где пользователи смогут просматривать и покупать товары. Требования включают возможность регистрации и авторизации пользователей, каталог товаров с фильтрами, корзину покупок, онлайн-оплату и удобную навигацию по сайту.

2. *Исследование и анализ.* Исследование рынка показывает, что существует большое количество конкурирующих онлайн-магазинов, поэтому важно создать уникальный и привлекательный дизайн. Анализ аудитории выявляет, что

большинство пользователей используют мобильные устройства для покупок, поэтому веб-приложение должно быть адаптировано для мобильных платформ.

3. *Создание структуры и макетов.* Создается структура сайта с разделами, такими как «Главная страница», «Каталог товаров», «Корзина», «Оплата» и др. Разрабатываются макеты страниц, отображающие расположение элементов и общий дизайн.

4. *Разработка функциональности.* Программируются функции регистрации и авторизации пользователей, создается база данных для хранения информации о товарах и заказах. Интегрируется платежная система для онлайн-оплаты заказов.

5. *Тестирование и отладка.* Проводятся тесты для проверки функциональности сайта, такие как регистрация, добавление товаров в корзину, оформление заказа и оплата. Ошибки исправляются, а функциональность проверяется повторно.

6. *Запуск и мониторинг.* Онлайн-магазин разворачивается на веб-сервере, выполняются настройки для обеспечения безопасности и производительности. Устанавливаются механизмы мониторинга, чтобы отслеживать работу приложения и быстро реагировать на возможные проблемы.

В результате этих этапов планирования вы создаете качественное веб-приложение, которое соответствует требованиям и целям вашего проекта, а также обеспечивает хороший пользовательский опыт и функциональность для ваших клиентов.

ОПРЕДЕЛЕНИЕ ЦЕЛЕЙ И ТРЕБОВАНИЙ ПРОЕКТА

Определение целей и требований проекта является важным этапом веб-разработки. На этом этапе определяются основные цели, задачи и требования, которые должны быть выполнены при разработке веб-приложения. Рассмотрим процесс определения целей и требований проекта.

1. Определение целей проекта

Цели проекта определяют конечные результаты, которых необходимо достичь. Они должны быть ясными, измеримыми,

достижимыми, релевантными и ограниченными по времени (SMART-критерии).

Пример.

Цель проекта. Разработать веб-приложение для онлайн-магазина, которое позволит пользователям просматривать и покупать товары, администраторам управлять каталогом товаров и обрабатывать заказы.

2. Определение требований проекта

Требования проекта определяют функциональные и нефункциональные характеристики, которые должны быть реализованы в веб-приложении. Они описывают, что приложение должно делать и каким образом.

Пример.

Требования проекта.

- Функциональные требования.
- Регистрация и аутентификация пользователей.
- Просмотр каталога товаров с возможностью фильтрации и сортировки.
- Добавление товаров в корзину и оформление заказа.
- Администрирование каталога товаров и управление заказами.
- Нефункциональные требования.
- Быстрая загрузка страниц и отзывчивый интерфейс.
- Безопасность данных пользователей.
- Совместимость с различными браузерами и устройствами.

Определение целей и требований проекта является фундаментальным этапом веб-разработки. Цели проекта определяют конечные результаты, которые нужно достичь, в то время как требования проекта определяют функциональные и нефункциональные характеристики приложения. Они должны быть ясно сформулированы и предоставлять четкие указания для разработчиков.

В таблице ниже приведены примеры для определения целей и требований проекта.

Таблица 1.2

Элемент	Описание	Пример
Цели проекта	Конечные результаты, которые нужно достичь	Разработать веб-приложение для онлайн-магазина
Требования проекта	Функциональные и нефункциональные характеристики приложения	Просмотр каталога товаров с возможностью фильтрации и сортировки

РАЗРАБОТКА ПЛАНА ПРОЕКТА, ВКЛЮЧАЯ ЗАДАЧИ, СРОКИ И РЕСУРСЫ

Разработка плана проекта является важным этапом в процессе управления проектом. План определяет задачи, сроки и ресурсы, необходимые для успешной реализации проекта.

1. Задачи проекта

Первый шаг в разработке плана проекта — определение задач, которые должны быть выполнены. Задачи могут включать в себя различные аспекты проекта, такие как анализ требований, разработка, тестирование и внедрение. Примеры задач для веб-приложения могут выглядеть следующим образом.

Таблица 1.3

Задача	Описание
Анализ требований	Изучение и документирование требований к проекту
Проектирование интерфейса	Создание макетов и визуальных прототипов интерфейса
Разработка бэкенда	Создание серверной части приложения и API
Разработка фронтенда	Создание клиентской части приложения и интерфейса

Задача	Описание
Тестирование и отладка	Проведение функционального и интеграционного тестирования
Внедрение и запуск	Установка приложения на сервер и запуск
Документация и поддержка	Создание документации и предоставление поддержки

2. Сроки проекта

Планирование сроков проекта важно для управления временными ресурсами и выполнения задач вовремя. В таблице ниже представлен пример графика сроков для различных задач проекта.

Таблица 1.4

Задача	Начало	Завершение
Анализ требований	01.08.2025	15.08.2025
Проектирование интерфейса	16.08.2025	30.08.2025
Разработка бэкенда	01.09.2025	30.09.2025
Разработка фронтенда	01.10.2025	31.10.2025
Тестирование и отладка	01.11.2025	15.11.2025
Внедрение и запуск	16.11.2025	30.11.2025
Документация и поддержка	01.12.2025	15.12.2025

3. Ресурсы проекта

Определение ресурсов проекта важно для эффективного использования бюджета и распределения задач между командой. Ресурсы могут включать в себя персонал, оборудование, программное обеспечение и другие ресурсы, необходимые для успешного завершения проекта. Пример таблицы с ресурсами может выглядеть следующим образом.

Таблица 1.5

<i>Ресурс</i>	<i>Доступные количество/возможности</i>
Разработчики	2
Дизайнеры	1
Тестировщики	1
Сервер	1
Бюджет	10 000 000 ₽

Разработка плана проекта позволяет управлять процессом выполнения задач и определить сроки, необходимые для достижения целей проекта. Определение задач, сроков и ресурсов помогает организовать работу команды и использовать ресурсы эффективно. Примеры таблиц и примеров, представленных выше, помогают наглядно представить структуру и компоненты плана проекта.

Важно помнить, что разработка плана проекта является итеративным процессом, который может требовать дальнейшей детализации и корректировки в ходе выполнения проекта.

УПРАВЛЕНИЕ РИСКАМИ И АНАЛИЗ ЗАВИСИМОСТЕЙ

Управление рисками и анализ зависимостей являются важными аспектами управления проектом. Рассмотрим примеры, чтобы помочь вам в управлении рисками и анализе зависимостей в проекте.

1. Управление рисками

Управление рисками включает идентификацию, анализ и управление потенциальными рисками, которые могут возникнуть в проекте. Пример таблицы для управления рисками может выглядеть следующим образом (см. Табл. 1.6).

Управление рисками в проекте включает оценку потенциальных проблем и разработку планов действий для их устранения или смягчения. Пример таблицы с рисками помогает отслеживать статус каждого риска и иметь план действий для управления ими.

Таблица 1.6

Риск	Вероятность	Влияние	Статус	План действий
Задержка в поставке	Средняя	Высокое	Активный	Согласовать новые сроки с поставщиком
Отсутствие ключевого ресурса	Низкая	Среднее	Мониторинг	Искать замену или альтернативные решения
Технические проблемы	Высокая	Высокое	Активный	Провести дополнительные тесты и исправления
Изменение требований	Средняя	Среднее	Мониторинг	Проанализировать влияние и обсудить с заказчиком

2. Анализ зависимостей

Анализ зависимостей позволяет определить связи между различными элементами проекта и понять, как изменения в одном элементе могут повлиять на другие. Пример таблицы для анализа зависимостей может выглядеть следующим образом.

Таблица 1.7

Задача	Зависит от	Влияет на
Анализ требований	-	Проектирование, Разработка
Проектирование интерфейса	Анализ требований	Разработка, Тестирование

<i>Задача</i>	<i>Зависит от</i>	<i>Влияет на</i>
Разработка бэкенда	Анализ требований	Разработка фронтенда, Тестирование
Разработка фрон-тенда	Анализ требований, Проектирование ин-терфейса, Разработка бэкенда	Тестирование, Внедре-ние и запуск
Тестирование и отладка	Разработка бэкенда, Разработка фронтенда	Внедрение и запуск, Документация и под-держка
Внедрение и запуск	Разработка фронтенда, Тестирование	-
Документация и поддержка	Тестирование и отлад-ка, Внедрение и запуск	-

Анализ зависимостей помогает в планировании и управлении проектом, учитывая связи между различными задачами. Пример таблицы с зависимостями позволяет наглядно представить связи между задачами и понять, какие задачи зависят от других и какие задачи влияют на другие.

Важно проводить регулярное обновление таблиц и анализировать риски и зависимости в процессе выполнения проекта. Это поможет управлять рисками и ресурсами более эффективно, а также принимать своевременные меры для устранения проблем и обеспечения успешной реализации проекта.

ЗАКЛЮЧЕНИЕ

Глава 1 предоставила нам важное понимание роли и значения планирования веб-разработки. Мы изучили различные этапы планирования, которые помогают обеспечить успешное выполнение проекта и достижение его целей.

Одним из первостепенных аспектов планирования является определение целей и требований проекта. Это позволяет разработчикам и заказчикам иметь ясное представление о том,

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru