



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«МИРЭА – Российский технологический университет»**

**РТУ МИРЭА**

Институт информационных технологий (ИИТ)

Кафедра математического обеспечения и стандартизации информационных технологий  
(МОСИТ)

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ**  
по дисциплине «Обоснование и разработка требований к программным  
системам»

**Практическое занятие № 7**

Вариант № 13. Обоснование и разработка требований к программной  
системе управления разработками цифровой компании

Студент группы  
ИКБО-65-23

*ИКБО-65-23, Олефиров ГГ*

\_\_\_\_\_  
(подпись)

Преподаватель

*Бирюкова Анна Алексеевна*

\_\_\_\_\_  
(подпись)

Отчет представлен

*«\_\_\_»\_\_\_\_202\_\_г.*

Москва 2025 г.

## Цель занятия

Научиться разрабатывать нефункциональные требования к системе.

### 1. Постановка задачи

Задачами работы являются:

- описание требований к качеству продукта;
- описание ограничений системы;
- описание характеристик внешнего интерфейса.

В третьей практической работе была разработана диаграмма вариантов использования, представленная на рисунке 1.

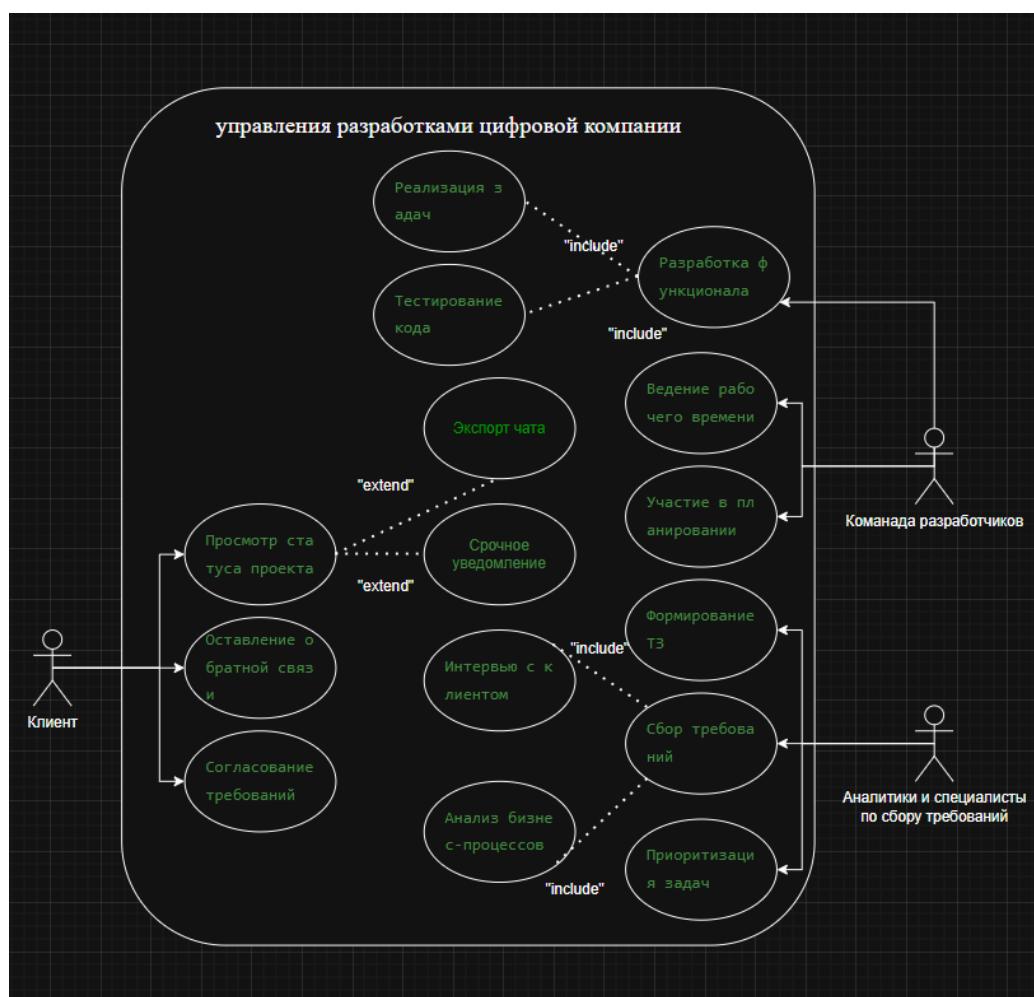


Рисунок 1 – Диаграмма вариантов использования для управления разработками цифровой компании

## **2. Результат работы**

### **2.1 Атрибуты качества**

В практической работе были определены атрибуты качества, необходимые для разработки системы управления разработками цифровой компании. Описанные атрибуты приведены в приложении 1.

### **2.2 Ограничения**

Для системы управления разработками были определены следующие ограничения:

CON-1: Система должна соответствовать требованиям GDPR и Федерального закона №152-ФЗ «О персональных данных» при обработке данных пользователей (клиентов, разработчиков, аналитиков). [юридическое ограничение]

CON-2: Интерфейс должен быть реализован на React.js с использованием Tailwind CSS для обеспечения единообразного дизайна. [ограничение реализации]

CON-3: Система должна корректно работать на устройствах с разрешением экрана от 1024x768 и скоростью интернет-соединения не менее 2 Мбит/с. [аппаратное ограничение]

CON-4: Переключение между ролями (Клиент/Разработчик/Аналитик) должно осуществляться через единую панель выбора в верхней части интерфейса без перезагрузки страницы. [ограничение дизайна]

CON-5: Данные о задачах, код-ревью и учете времени должны храниться и передаваться в формате JSON. [ограничение данных]

CON-6: Все текстовые элементы интерфейса должны использовать русский язык с минимальным применением технической терминологии. [нормативное ограничение]

## 2.2.1 Пользовательский интерфейс в виде карты диалоговых окон

На рисунках 2-8 представлены макеты интерфейса системы управления разработками цифровой компании

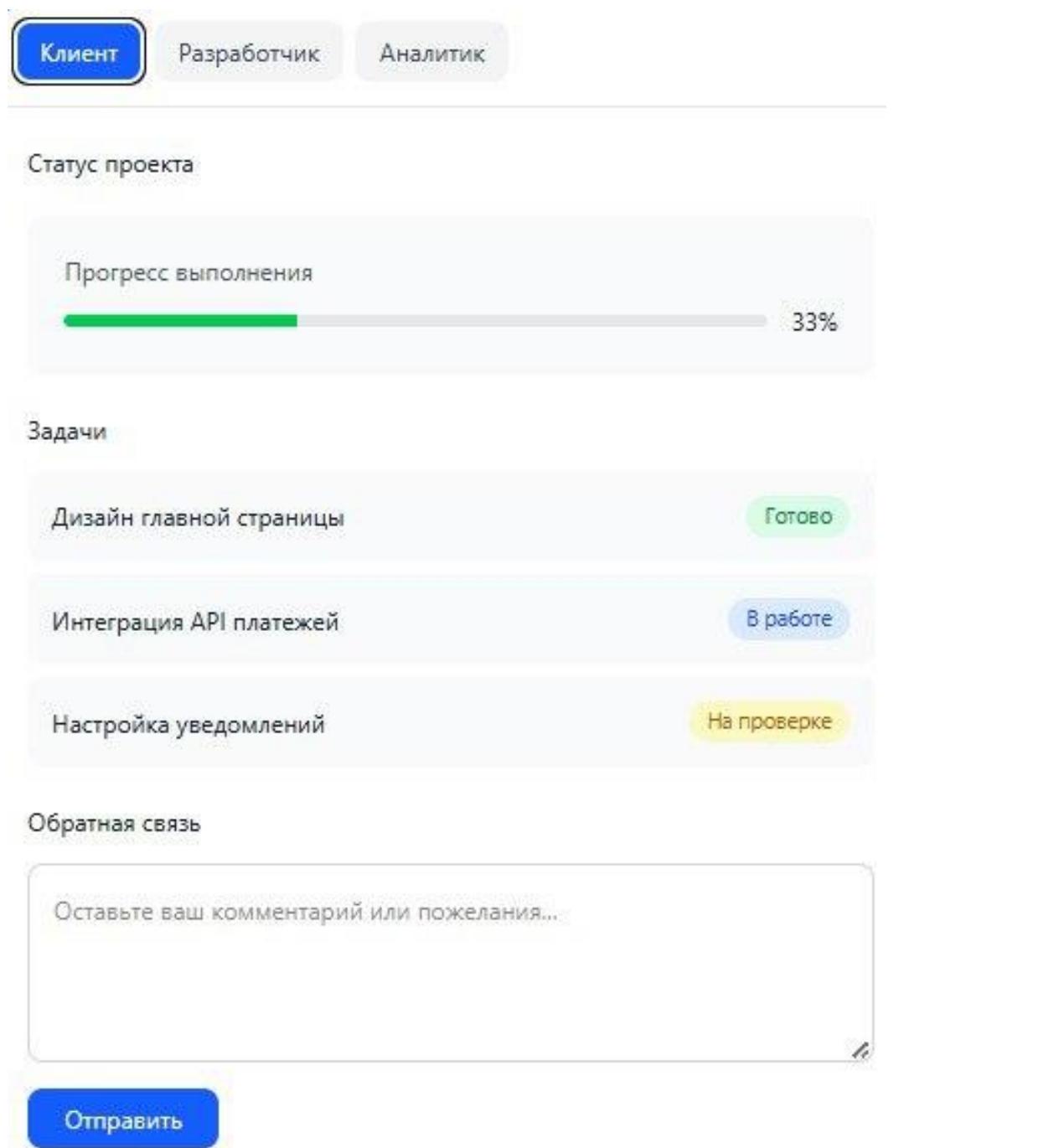


Рисунок 2 – Макет клиентской панели

Клиент

Разработчик

Аналитик

## Рабочая панель

Задачи

Время

Код-ревью

Документация

### Документация проекта



API Документация

Описание всех API эндпоинтов

Открыть →



Руководство по настройке

Инструкция по развертыванию проекта

Открыть →



Стандарты кода

Правила оформления и структура

Открыть →



Архитектура системы

Диаграммы и описание компонентов

Открыть →



Быстрая справка

`npm start` - Запуск dev сервера

`npm test` - Запуск тестов

`npm build` - Сборка проекта

Рисунок 3 – Рабочая панель разработчика

Клиент

Разработчик

Аналитик

## Рабочая панель

Задачи

Время

Код-ревью

Документация

### Мои задачи

Интеграция API платежей

В работе

Высокий приоритет

Коммит

На проверку

Настройка уведомлений

На проверке

Низкий приоритет

Коммит

На проверку

Рисунок 4 – Рабочая панель разработчика

Клиент

Разработчик

Аналитик

## Рабочая панель

Задачи

Время

Код-ревью

Документация

### Учет времени

Часы

Выберите задачу

Записать

### История

Интеграция API платежей

4.5 ч

Сегодня, 14:30

Настройка уведомлений

3 ч

Вчера, 16:00

Дизайн главной страницы

6 ч

2 дня назад

Всего за неделю

13.5 часов

Рисунок 5 – Рабочая панель разработчика

Клиент

Разработчик

Аналитик

## Рабочая панель

Задачи

Время

Код-ревью

Документация

### Учет времени

Часы

Интеграция API платежей

Записать

Выберите задачу

Дизайн главной страницы

Интеграция API платежей

История

Интеграция API

Сегодня, 14:30

4.5 ч

Настройка уведомлений

3 ч

Настройка уведомлений

Вчера, 16:00

6 ч

Дизайн главной страницы

2 дня назад

Всего за неделю

13.5 часов

Рисунок 6 – Рабочая панель разработчика

Клиент

Разработчик

Аналитик

## Рабочая панель

Задачи

Время

Код-ревью

Документация

### Код-ревью

Добавить валидацию форм

Ожидает

Автор: Иван

3 комментариев

Отклонить

Одобрить

Просмотр

Оптимизация запросов БД

Одобрено

Автор: Мария

5 комментариев

Просмотр

Рефакторинг компонентов

Ожидает

Автор: Петр

1 комментариев

Отклонить

Одобрить

Просмотр

+ Создать Pull Request

Рисунок 7 – Рабочая панель разработчика

Клиент

Разработчик

Аналитик

### Управление требованиями

#### Новое требование

Название требования

Описание требования

Приоритет

Добавить

#### Приоритизация задач

#1 Дизайн главной страницы



Высокий

#2 Интеграция API платежей



Высокий

#3 Настройка уведомлений



Низкий

#### Анализ требований

Всего требований

3

Высокий приоритет

2

Рисунок 8 – Рабочая панель аналитика

## 2.2.2 Общая схема общего взаимодействия прототипов экранных форм

На рисунке 6 представлена диаграмма состояний [2], отражающая взаимосвязь между элементами интерфейса.

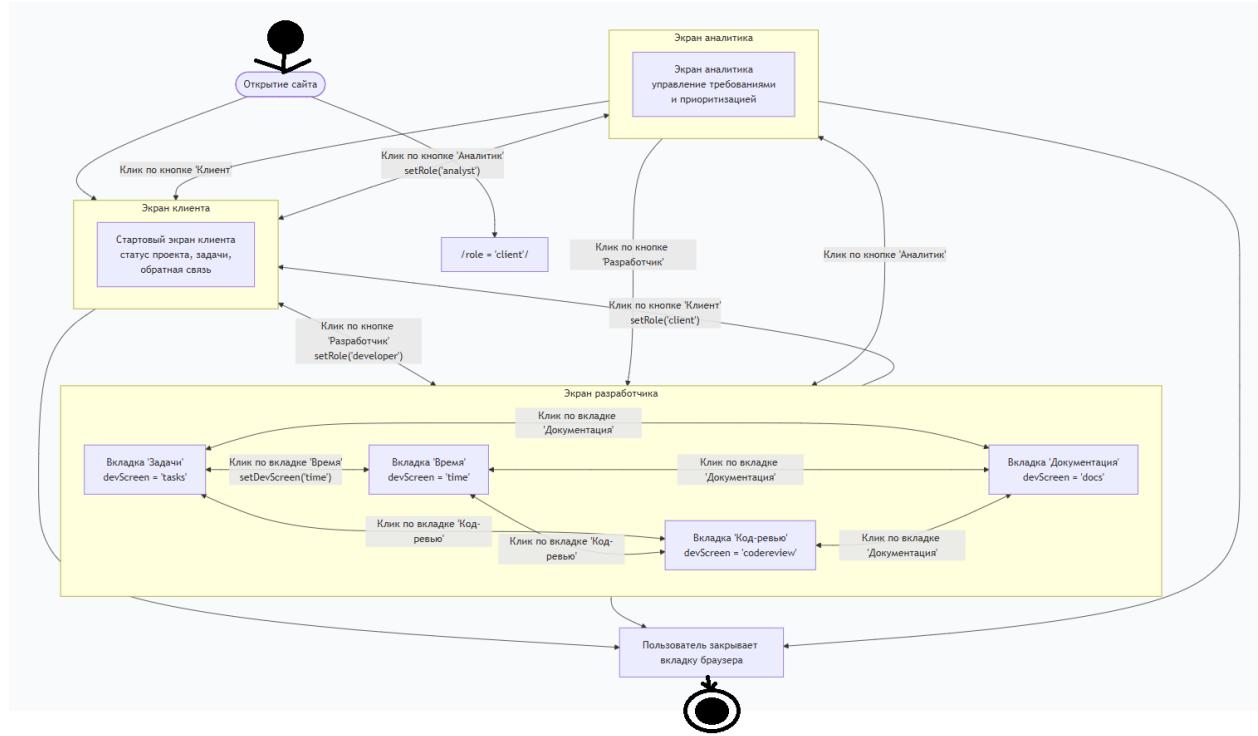
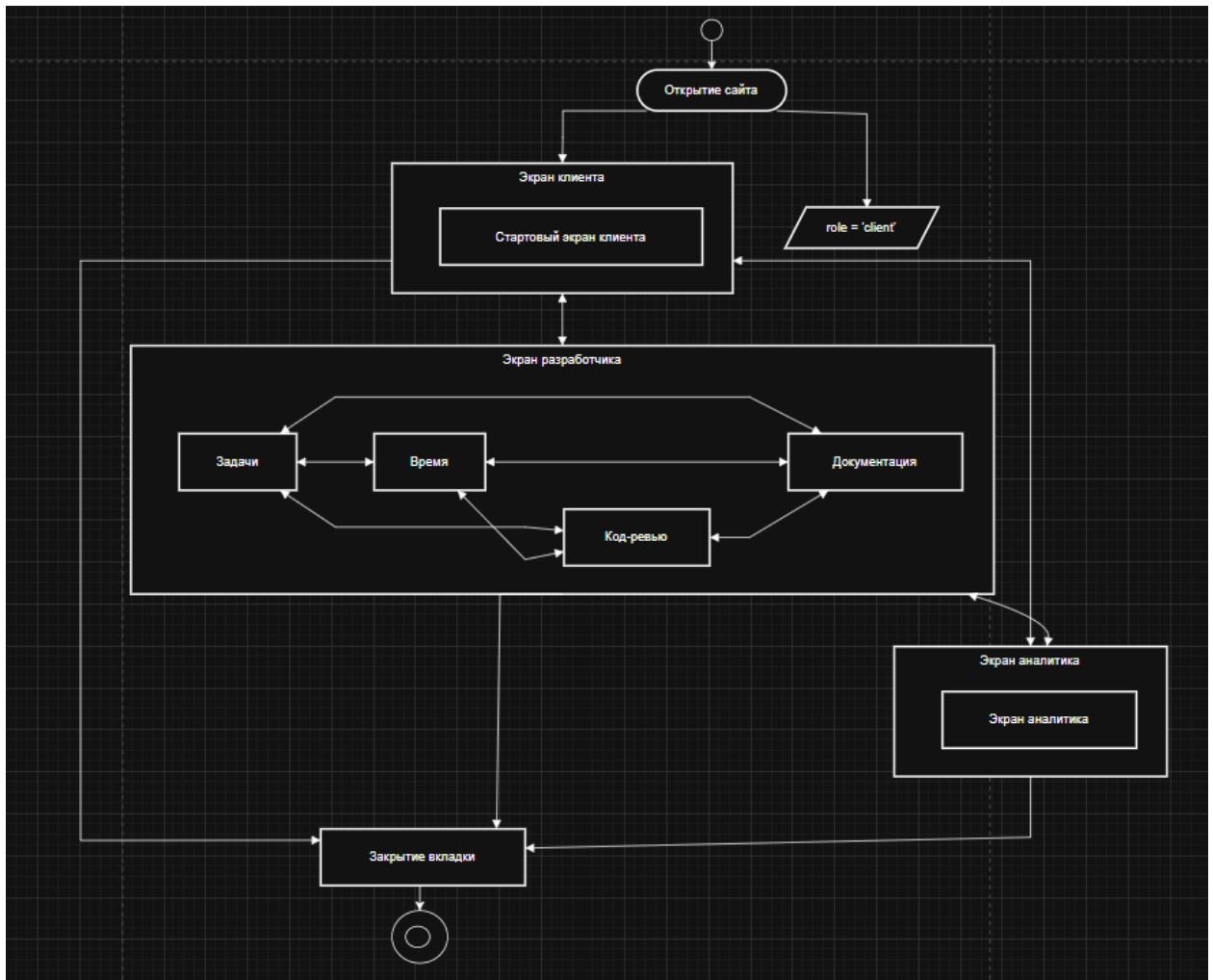


Рисунок 9 – Диаграмма состояний, отражающая взаимодействие элементов интерфейса программной системы управления разработками цифровой компании.



### 2.2.3 Диаграмма компонентов

Для дальнейшей реализации системы были выявлены и описаны компоненты сайта, визуализация представлена на рисунке 7, отображающем диаграмму компонентов [2].

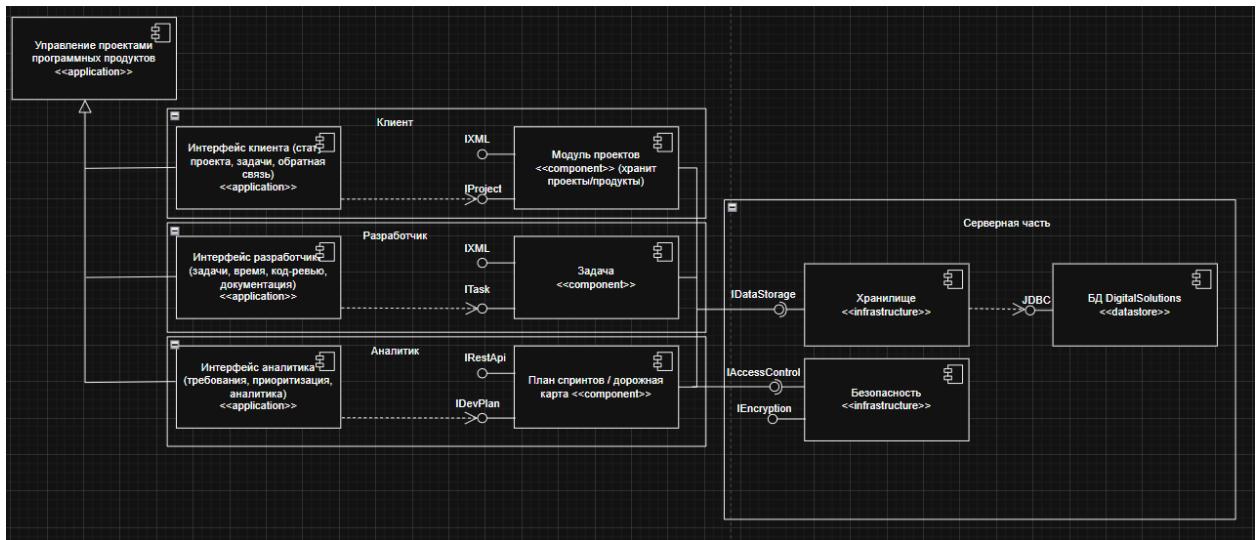


Рисунок 11 – Диаграмма компонентов, отражающая взаимодействие компонентов системы

#### 2.2.4 Диаграмма размещения

Для отображения физической архитектуры системы была создана диаграмма размещения [2], представленная на рисунке 8.

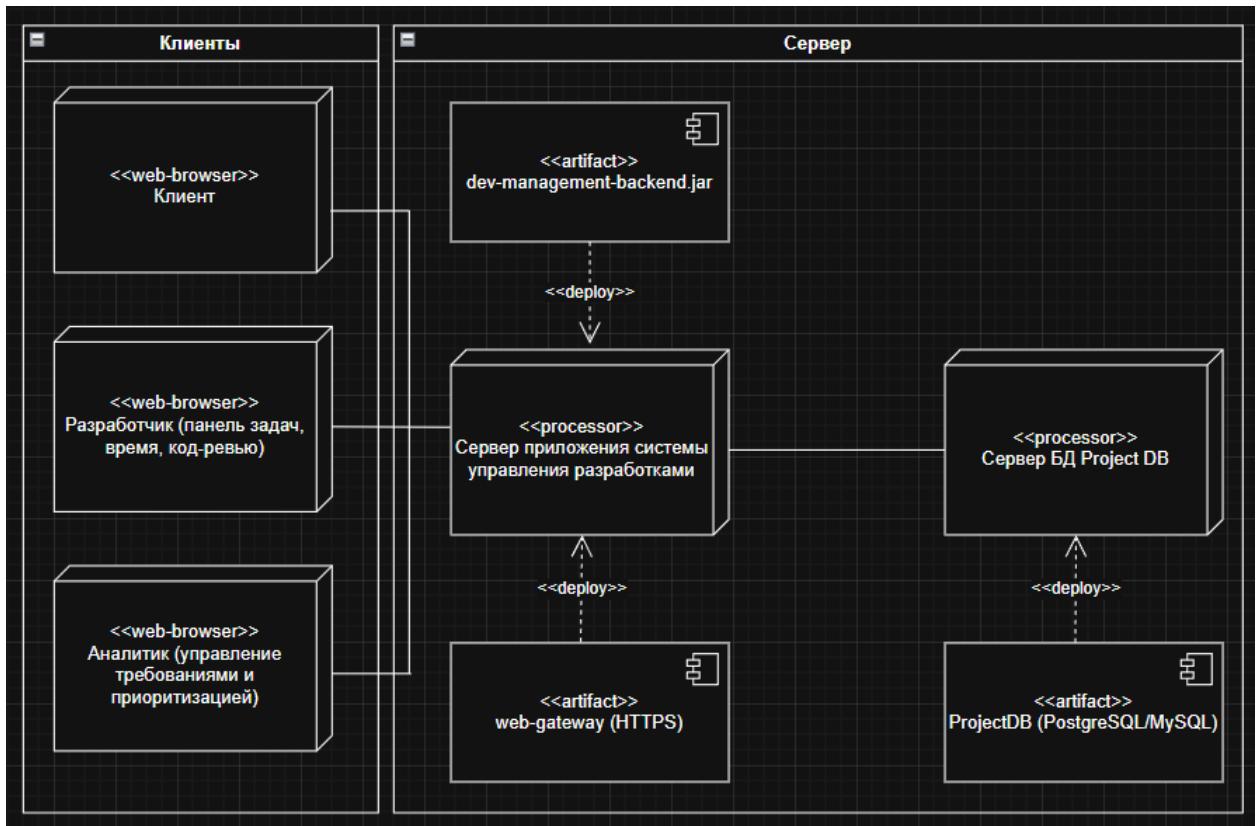


Рисунок 12 – Диаграмма размещения программной системы управления разработками цифровой компании

## **2.2.5 Коммуникационные интерфейсы**

### **1.1.1 Коммуникационные интерфейсы**

При реализации программной системы управления разработками цифровой компании используются следующие коммуникационные интерфейсы:

HTTP/HTTPS – между веб-браузером пользователя (клиент, разработчик, аналитик) и веб-сервером системы, для передачи запросов к интерфейсу управления проектами, задачами, код-ревью и требованиями, а также для отображения веб-страниц.

WebSocket (или long polling HTTP) – между веб-браузером и сервером приложения, для передачи событий в реальном времени (обновление статуса задач, прогресса проекта, уведомлений о код-ревью).

SMTP – между сервером приложения и почтовым сервером, для отправки почтовых уведомлений участникам проекта о новых задачах, изменениях статусов и результатах ревью.

TLS/SSL – между веб-браузером и веб-сервером, для шифрования всего трафика системы (автентификация пользователей, передача данных задач и требований).

Коннекторы к СУБД (например, JDBC-драйвер PostgreSQL/MySQL) – между сервером приложения и сервером базы данных Project DB, для выполнения запросов на чтение и запись данных по проектам, задачам, временным логам, требованиям и пользователям.

## **Список использованных источников и литературы**

1. Ахмедова, Х. Г. Обоснование и разработка требований к программным системам : учебно-методическое пособие / Х. Г. Ахмедова, А. В. Овсянникова, А. А. Бирюкова. — Москва : РТУ МИРЭА, 2023. — 68 с.  
— ISBN 978-5-7339-2010-8. —  
URL: <https://e.lanbook.com/book/398132> (дата обращения: 15.11.2025). —  
Режим доступа: для авториз. пользователей.
2. Завьялов, А. В. Диаграммы UML для анализа и проектирования информационных систем : учебно-методическое пособие / А. В. Завьялов. — Москва : РТУ МИРЭА, 2021. — 65 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/218630> (дата обращения: 15.11.2025). —  
Режим доступа: для авториз. пользователей.

## Приложения

### Приложение 1. Атрибуты качества

#### Внешние атрибуты качества

##### Требования к доступности

AVL-1: Веб-система управления разработками должна быть доступна не менее чем 96,5% времени в течение календарного месяца.

AVL-2: Время восстановления системы после критического сбоя (полная недоступность интерфейса для всех ролей) не должно превышать 20 минут.

##### Требования к установке

INS-1: Развёртывание новой версии фронтенда и backend-сервиса по CI/CD-пайплайну должно занимать не более 30 минут при стандартной нагрузке.

INS-2: Процесс установки должен автоматически проверять версии Node.js, СУБД и наличие необходимых переменных окружения, отказывая в установке при несоответствии.

##### Требования к целостности

INT-1: Все данные задач, требований и записей учета времени, отправленные через веб-формы, должны сохраняться в базе данных без потери и искажения.

INT-2: Резервное копирование базы данных проекта и файлов артефактов (документация, вложения) должно выполняться автоматически не реже одного раза в сутки.

INT-3: Система должна защищать данные задач, требований и пользователей от неправомерного добавления, удаления или изменения через интерфейс или API.

#### Требования к совместимости

IOP-1: Веб-интерфейс должен корректно отображаться и функционировать в актуальных версиях браузеров Google Chrome, Mozilla Firefox, Yandex Browser, Microsoft Edge и Opera на момент внедрения.

IOP-2: Интерфейс панели клиента, разработчика и аналитика должен быть адаптивным и поддерживать корректное отображение на устройствах с шириной экрана от 360 до 1920 пикселей.

#### Требования к производительности

PER-1: Время полной загрузки основной панели (для любой роли) при подключении от 5 Мбит/с не должно превышать 2 секунд для десктопной версии и 3 секунды для мобильных устройств.

PER-2: Система должна поддерживать одновременную работу не менее 200 активных пользователей (клиенты, разработчики, аналитики) без заметной деградации времени отклика (до 3 секунд на основные операции).

#### Требования к надежности

REL-1: Частота критических сбоев (полная недоступность интерфейса более чем на 30 минут) не должна превышать одного случая в месяц.

REL-2: Система должна корректно обрабатывать некорректный ввод в формах задач, требований и учета времени без падения сервера и с выдачей информативных сообщений об ошибках.

REL-3: При ошибке на стороне сервера пользователю должно отображаться понятное сообщение о технической проблеме с возможностью повторить попытку операции.

#### Требования к устойчивости

ROB-1: При кратковременной недоступности базы данных система должна сохранять доступ к статическим страницам и отображать сообщение о временной недоступности функционала работы с задачами и требованиями.

#### Требования к защите ресурсов

SAF-1: Одновременный запуск веб-клиента не должен приводить к использованию более 40 МБ оперативной памяти в браузере при стандартном сценарии работы (просмотр задач и код-ревью).

#### Требования к безопасности

SEC-1: Пароли всех учетных записей (включая администраторов и разработчиков) должны храниться только в хешированном виде с использованием современного устойчивого алгоритма (например, bcrypt/argon2) с солью.

SEC-2: Пользовательские сессии должны иметь ограниченное время жизни (не более 12 часов бездействия) и уничтожаться после выхода пользователя из системы.

SEC-3: Вся информация, вводимая в формах (логины, комментарии, названия задач), должна валидироваться на стороне клиента и сервера для предотвращения внедрения SQL- и XSS-атак.

SEC-4: Пользователь, получивший временный пароль от администратора, обязан изменить его при первом успешном входе в систему.

#### Требования к удобству использования

USE-1: Клиент должен иметь возможность посмотреть статус проекта и список своих задач не более чем за два клика с момента открытия главной страницы.

USE-2: Формы создания задач, требований и записей учета времени должны быть интуитивно понятными, содержать подсказки по обязательным полям и мгновенную валидацию ввода.

USE-3: Основные разделы интерфейса («Статус проекта», «Рабочая панель», «Управление требованиями») должны быть доступны из верхнего меню переключения ролей или главной панели навигации.

#### Внутренние атрибуты качества

#### Требования к эффективности

EFF-1: Средняя загрузка CPU сервера при работе 100 одновременных пользователей не должна превышать 60% при штатном сценарии (просмотр и редактирование задач, код-ревью, учет времени).

EFF-2: Для статики (JS-бандлы, стили, изображения) должно использоваться

кеширование на уровне веб-сервера или CDN для снижения нагрузки на приложение.

#### Требования к возможности модификации

MOD-1: Основные настройки интерфейса (цветовая схема, логотип, базовые шрифты) должны конфигурироваться через единый файл конфигурации или тему без изменения бизнес-логики.

MOD-2: Вложенность вызываемых функций в модулях фронтенда и backend не должна превышать двух–трех уровней, чтобы упростить сопровождение и тестирование.

#### Требования к обслуживанию

SUP-1: Система должна вести журналы (логи) всех ошибок, неуспешных попыток авторизации и критических событий (сбои БД, падение сервисов) с возможностью последующего анализа администраторами.

SUP-2: Кодовая база должна быть документирована минимум на уровне модулей и основных компонентов (описание назначения, входных и выходных данных, возможных ошибок).

#### Требования к переносимости

POR-1: Конфигурация сервера (строка подключения к БД, параметры очередей, внешние API) должна храниться во внешних конфигурационных файлах или переменных окружения, а не быть жестко прописанной в исходном коде.

## Требования к повторному использованию

REU-1: Модули интерфейса задач, код-ревью и управления требованиями должны быть реализованы как переиспользуемые компоненты, пригодные для включения в другие внутренние системы компании.

REU-2: Стили UI-компонентов (кнопки, формы, таблицы) должны быть унифицированы и оформлены в виде общей дизайн-системы для повторного использования в будущих проектах.

## Требования к масштабируемости

SCA-1: Архитектура системы должна позволять горизонтальное масштабирование backend-сервиса (добавление экземпляров приложения за балансировщиком) без изменений в коде.

SCA-2: Хранилище данных должно поддерживать увеличение объема (дискового пространства) без остановки системы и миграции на другую платформу.

## Требования к проверяемости

VER-1: Для ключевых модулей (авторизация, управление задачами, код-ревью, учет времени) должны быть разработаны модульные и интеграционные тесты с автоматическим прогоном в конвейере сборки.

VER-2: Конфигурация среды тестирования должна быть максимально приближена к конфигурации промышленной среды (версии СУБД, ОС, runtime), чтобы минимизировать невоспроизводимые дефекты.

VER-3: Процесс развертывания новой версии должен включать автоматический запуск набора регрессионных тестов и контроль успешного завершения перед выкладкой в продуктив.