Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет Программной инженерии и компьютерной техники

**Лабораторная работа №1 по**

**Экономике программной инженерии**

Вариант: <https://kudago.com/>

Работу выполнили:

Барсуков М.А., группа P3415,

Ценеков Д.А., группа Р3410

Преподаватель:

Блохина Е. Н.

Санкт-Петербург,

2025

# **Содержание**

[**Текст задания 3**](#_hfzfco5669ds)

[**Выполнение 4**](#_qrnpekutwoun)

[Функциональные требования 4](#_o6d04kq1sulb)

[Оценка трудоемкости наивным методом 7](#_lesrdpgoboct)

[Оценка трудоемкости методом PERT 12](#_ltiv59mrny1x)

[Сетевая диаграмма 15](#_ii8fqct88q77)

[Метод критического пути 16](#_esoume6saaa)

[Оценка методом функциональных точек 18](#_devi1okhzy88)

[1. Определение типа оценки 18](#_njaj0zes8b3n)

[2. Определение области оценки и границ продукта 18](#_kwvdq4x70zqh)

[3. Расчет функциональных точек, связанных с данными 19](#_by6ds6h43by2)

[4. Расчет функциональных точек транзакций 20](#_41p6jd37530h)

[5. Определение суммы невыровненных ФТ (UFP) 20](#_9pxoh12h3esm)

[6. Определение фактора выравнивания (VAF) 21](#_teei99vcbruv)

[7. Расчет количества выровненных функциональных точек (AFP) 21](#_1tppw66lgnj2)

[Расчет трудоемкости методом COCOMO II 22](#_bl54232vbq59)

[1. Конвертация функциональных точек в SLOC 22](#_ahxipf8o7re9)

[2. Расчет факторов масштаба (SF) 22](#_6s1u9w3hev8s)

[3. Расчет множителей усилия (EM) 23](#_nwxkmo7ulwmv)

[4. Итоговый расчёт трудоёмкости 24](#_b94c03ono82d)

[5. Расчет сроков разработки 24](#_sadwdiglj1ge)

[6. Расчет команды 24](#_8gnzl4vluu7t)

[Итоговая оценка 25](#_7lk8shkcttw8)

[Метод оценки вариантов использования (Use Case Points) 26](#_ntcqtnqzke0)

[Оценка веса прецедентов (UUCW) 27](#_47psuha27bdz)

[Оценка веса акторов (UAW) 27](#_2jrdwtxv3c31)

[Оценка технических факторов (TCF) 28](#_ajtr5v7b58du)

[Оценка факторов окружения (ECF) 29](#_q1qy7e9m4ovu)

[Вычисление UCP 29](#_3174oynhdn11)

[Подсчет фактора продуктивности (PF) 30](#_25oiapqn0ulo)

[Применение фактора продуктивности (PF) 33](#_heobc9plzk7)

[Анализ результатов 34](#_cuu7898n6qnl)

[**Вывод 35**](#_ivkp8uwrw9kw)

# **Текст задания**

Для выданного веб-проекта:

1. Сформировать набор функциональных требований для разработки проекта.
2. Оценить трудоемкость разработки проекта наивным методом.
3. Оценить трудоемкость разработки проекта методом PERT (Project Evaluation and Review Technique**).** Нарисовать сетевую диаграмму взаимосвязи работ и методом критического пути рассчитать минимальную продолжительность разработки. Предложить оптимальное количество разработчиков и оценить срок выполнения проекта.
4. Оценить размер проекта методом функциональных точек, затем, исходя из предположения, что собранной статистики по завершенным проектам нет, рассчитать трудоемкость методом COCOMO II ([Обновленная таблица количества строк на точку для разных языков программирования](https://www.qsm.com/resources/function-point-languages-table))
5. Оценить размер проекта методом оценки вариантов использования (Use Case Points). Для расчета фактора продуктивности PF использовать любой свой завершенный проект с известными временными трудозатратами, оценив его размер методом UCP.
6. Сравнить полученные результаты и сделать выводы.

Веб-проект: <https://kudago.com/>

# **Выполнение**

## **Функциональные требования**

1. Управление контентом и событиями

* Система категоризации: поддержка основных категорий, подкатегорий и множественной категоризации для событий.
* Управление событиями: создание разовых, повторяющихся событий и событий с несколькими датами через календарный интерфейс.
* Управление местами: создание карточек мест с фото, адресом, контактами, галереей и привязкой к карте.
* Геолокация: привязка событий и мест к локациям, поддержка мультигородости и переключения между городами.
* Модерация контента: система статусов для событий и пользовательского контента (комментарии, отзывы) перед публикацией.
* Ролевая модель: система прав для редакторов и модераторов.

2. Пользовательский интерфейс и навигация

* Виджет выбора города: с сохранением предпочтений пользователя в cookies/сессии.
* Комплексная система фильтрации: фильтры по категориям, дате, цене, возрасту, меткам и району.
* Полнотекстовый поиск: с AJAX-обновлением результатов по мере ввода.
* Интерактивный календарь: просмотр событий по датам с подсветкой активных дней.
* Адаптивный дизайн: mobile-first верстка для мобильных устройств и планшетов.

3. Пользовательские аккаунты и персонализация

* Регистрация и аутентификация: через email и социальные сети.
* Личный кабинет: функционал «Избранное» для событий и мест, настройка напоминаний и управление email-рассылками.
* Пользовательский контент: система комментариев, отзывов, рейтингов и лайков с последующей модерацией.

4. Работа с местами (локациями)

* Детальные карточки мест с фото, адресом, контактами, галереей, привязкой к карте.
* Привязка событий к местам, отображение будущих событий в карточке места.

5. Дополнительный контент и функции

* Новостной раздел и статьи: CMS для публикации редакционных материалов.
* Редакционные подборки и гиды: создание тематических списков событий и мест.
* Система тегов: гибкая система тегов для событий и статей, формирование автоматических подборок.
* Интерактивные тесты и викторины: CMS для их создания и логика для подсчёта результатов.
* Аналитика контента: подсчёт просмотров, формирование пользовательских рейтингов и топа популярности.

6. Интеграции и монетизация

* REST API: предоставление данных партнёрским сервисам.
* Система бронирования и оплаты билетов:
  + Бронирование билетов на сайте с выбором места (если применимо).
  + Автоматическая отмена брони и возврат билетов в продажу, если оплата не была произведена в течение 15 минут.
  + Автоматическая отмена билета (аннулирование) в случае отмены денежного транзакции (chargeback) или возврата средств (refund).
* Реферальные ссылки: интеграция с билетными системами.
* Система баннерной рекламы: управление и ротация рекламных мест, возможность промотирования событий.
* Email-рассылки: целевые рассылки по категориям и городам с механизмом double opt-in.

7. Административная панель

* Управление контентом: комплексное управление событиями, местами, статьями и подборками через WYSIWYG-редактор.
* Модерация: управление пользовательским контентом (комментарии, отзывы), обработка жалоб и бан пользователей.

8. Релиз и сопровождение

* Документирование, A/B-тестирование.
* Развёртывание, аналитика, поддержка и мониторинг.

## **Оценка трудоемкости наивным методом**

| № | Название | Описание | Оптимистичная оценка | Пессимистичная оценка | Наиболее вероятная оценка |
| --- | --- | --- | --- | --- | --- |
| 0 | **Проектирование** | |  |  |  |
| 0.1 | Сбор требований | Интервью с заказчиком, анализ конкурентов (afisha.ru, timepad.ru, kassir.ru), сбор функциональных и нефункциональных требований. | 16 | 40 | 24 |
| 0.2 | Анализ требований | Формализация требований, выявление противоречий, приоритизация (MoSCoW), согласование с редакцией и бизнесом. | 16 | 32 | 24 |
| 0.3 | Прототипирование и составление SRS | Создание low-fi прототипов ключевых экранов (список событий, карточка, фильтры, админка). Написание Software Requirements Specification (SRS) по IEEE 830. | 32 | 80 | 48 |
| 0.4 | Выбор стека технологий | Оценка вариантов: Next.js vs Nuxt, PostgreSQL vs MongoDB, Docker/K8s, CI/CD. Согласование с командой и инфраструктурой. | 8 | 24 | 16 |
| 0.5 | Дизайн | Необходимо составить полный дизайн будущего сайта в Figma: все страницы (desktop + mobile), состояния (загрузка, ошибка, пустой результат), дизайн-система, компоненты. | 80 | 200 | 120 |
| 1 | **Управление контентом и событиями** | |  |  |  |
| 1.1 | Категоризация событий | Реализовать систему категорий и подкатегорий для событий (Выставки, Концерты, Театр и т.д.). Возможность присваивать несколько категорий одному событию. Админ-панель для управления деревом категорий. | 24 | 56 | 40 |
| 1.2 | Привязка к геолокации | Система мультигородости. Каждое событие и место привязано к городу. Возможность переключения между городами с изменением всего контента. Админка для управления списком городов. | 40 | 96 | 64 |
| 1.3 | Управление расписанием | Сложная система работы с датами: разовые события, повторяющиеся события, события с несколькими датами, режимы работы. Календарный интерфейс в админке. | 48 | 120 | 80 |
| 1.4 | Модерация и публикация | Workflow модерации контента перед публикацией. Система статусов событий ("Выбор редакции", "Бесплатно" и др.). Ролевая модель для редакторов. | 32 | 80 | 48 |
| 2 | **Пользовательский интерфейс и навигация** | |  |  |  |
| 2.1 | Выбор города | Виджет выбора города в шапке сайта. Сохранение выбора в cookies/сессии. Динамическое обновление контента при смене города. | 16 | 40 | 24 |
| 2.2 | Календарь событий | Интерактивный календарь для навигации по датам. Подсветка дат с событиями. Фильтрация событий по выбранной дате. | 24 | 64 | 40 |
| 2.3 | Система фильтров и поиска | Комплексная система фильтрации: по категориям, дате, цене, возрастному ограничению, меткам, району. Полнотекстовый поиск. AJAX-обновление результатов. | 64 | 160 | 96 |
| 2.4 | Адаптивный дизайн | Responsive верстка для мобильных устройств и планшетов. Адаптация всех компонентов интерфейса. Mobile-first подход. | 80 | 200 | 120 |
| 3 | **Пользовательские аккаунты и персонализация** | |  |  |  |
| 3.1 | Регистрация и аутентификация | Система регистрации через email и социальные сети (VK, OK и др.). Email подтверждение. Восстановление пароля. | 40 | 96 | 64 |
| 3.2 | Личный кабинет | Функционал "Избранного" для событий и мест. Система напоминаний о событиях. Настройки email-рассылок. | 48 | 112 | 72 |
| 3.3 | Обратная связь | Система комментариев и отзывов к событиям и местам. Модерация комментариев. Рейтинги и лайки. | 32 | 80 | 48 |
| 4 | **Работа с местами (локациями)** | |  |  |  |
| 4.1 | Карточка места | Детальные карточки мест с фото, адресом, контактами, описанием. Галерея изображений. Привязка к карте. | 40 | 96 | 64 |
| 4.2 | Привязка событий к местам | Связь между событиями и местами проведения. В карточке места - список будущих событий. | 16 | 40 | 24 |
| 5 | **Дополнительный контент и функции** | |  |  |  |
| 5.1 | Новостной раздел и статьи | CMS для редакционных материалов. Различные типы контента: новости, статьи, обзоры, рецензии. Категоризация и тегирование. | 48 | 120 | 80 |
| 5.2 | Система тегов (меток) | Гибкая система тегов для событий и статей. Создание автоматических подборок на основе тегов. Облако тегов. | 24 | 64 | 40 |
| 5.3 | Рейтинги и просмотры | Система подсчета просмотров событий и статей. Пользовательские рейтинги. Топы и рейтинги популярности. | 24 | 56 | 40 |
| 5.4 | Интерактивные тесты и викторины | CMS для создания многовариантных тестов с вопросами, вариантами ответов, логикой подсчёта результатов и отображением персонализированного финального экрана. Поддержка тегов, категорий и привязки к городам. | 32 | 80 | 48 |
| 5.5 | Редакционные подборки и гиды | Инструмент для создания тематических списков («14 вещей, ради которых нужно приехать в Петербург»), с возможностью встраивания карточек событий/мест, изображений и описаний. | 24 | 64 | 40 |
| 6 | **Интеграции и монетизация** | |  |  |  |
| 6.1 | Интеграция с партнерами | REST API для партнеров. Реферальные ссылки для покупки билетов. Система отслеживания переходов. | 64 | 144 | 96 |
| 6.2 | Размещение рекламы | Система баннерной рекламы. Промотируемые события с маркировкой "Реклама". Ротация баннеров. | 40 | 96 | 64 |
| 6.3 | Подписка на рассылки | Система email-рассылок по категориям и городам. Double opt-in подписка. Шаблоны писем. | 32 | 80 | 48 |
| 7 | **Административная панель** | |  |  |  |
| 7.1 | Управление контентом | Комплексная админ-панель для управления всеми типами контента. WYSIWYG редактор. Массовые операции. | 80 | 200 | 128 |
| 7.2 | Модерация пользовательского контента | Интерфейс модерации комментариев и отзывов. Жалобы пользователей. Бан пользователей. | 24 | 64 | 40 |
| 8 | **Технические требования** | |  |  |  |
| 8.1 | Производительность | Оптимизация работы с большими объемами данных. Кэширование. Индексация. Оптимизация запросов. | 48 | 120 | 80 |
| 8.2 | Безопасность | Защита от XSS, CSRF, SQL-инъекций. Валидация данных. Безопасное хранение паролей. HTTPS. | 40 | 96 | 64 |
| 9 | **Тестирование и обеспечение качества** | |  |  |  |
| 9.1 | Юнит-тестирование | Написание и поддержка unit-тестов для критически важных модулей backend и frontend (логика фильтрации, расчёта дат, авторизации). Покрытие ≥ 70%. | 24 | 56 | 40 |
| 9.2 | Функциональное и интеграционное тестирование | Автоматизированное тестирование сквозных сценариев: поиск → фильтрация → просмотр карточки; публикация события → отображение на сайте. Использование Playwright / Cypress. | 40 | 96 | 64 |
| 9.3 | Тестирование UI/UX и кросс-браузерность | Проверка корректного отображения на всех поддерживаемых устройствах и браузерах (Chrome, Safari, Firefox, Edge, мобильные). Валидация адаптивной верстки. | 24 | 64 | 40 |
| 9.4 | Приемочное тестирование (UAT) | Проверка готовности релиза совместно с бизнес-аналитиками и редакцией. Соответствие требованиям и ожиданиям заказчика. | 16 | 32 | 24 |
| 9.5 | Нагрузочное и стресс-тестирование | Проверка устойчивости сайта при пиковых нагрузках (праздники, анонсы крупных событий). Тестирование времени отклика API и времени загрузки страниц. | 24 | 64 | 40 |
| 10 | **Релиз** | |  |  |  |
| 10.1 | Документирование | Процесс закрепления и систематизации всех процедур, где отражены все данные и правила. | 90 | 160 | 100 |
| 10.2 | Альфа- и бета-тестирование | Внутреннее альфа-тестирование с участием редакторов и QA. Бета-тестирование новых функций (например, тестов или подборок) на ограниченной аудитории через feature flags. | 16 | 40 | 24 |
| 10.3 | A/B-тестирование | Инфраструктура для проведения A/B-тестов: заголовки, кнопки, расположение блоков «Выбор редакции», формы подписки. Интеграция с аналитикой. | 32 | 80 | 48 |
| 10.4 | Поддержка | Комплексная работа над стабильностью, безопасностью и масштабируемостью системы в условиях реального использования | 70 | 130 | 115 |
| 10.5 | Развёртывание | Процесс размещения программного обеспечения на сервере или другом целевом окружении для его использования конечными пользователями. | 160 | 200 | 170 |
| 10.6 | Аналитика | Регулярный анализ поведения пользователей и улучшение продукта. | 40 | 80 | 66 |
| Итого: | | | 1672 | 3802 | 2515 |

**Оценка наивным методом**: 2515 чел/час.

## **Оценка трудоемкости методом PERT**

− пессимистичная оценка;

− оптимистичная оценка;

− наиболее вероятная оценка;

− оценка средней трудоемкости;

− общая оценка статически независимых работ;

− среднеквадратичное отклонение;

− среднеквадратичное отклонение для оценки суммарной трудоемкости.

| № | Название | Оптимистичная оценка (O) | Пессимистичная оценка (P) | Наиболее вероятная оценка (M) |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | **Проектирование** | | | | | |
| 0.1 | Сбор требований | 16 | 40 | 24 | 25,33 | 4,00 |
| 0.2 | Анализ требований | 16 | 32 | 24 | 24,00 | 2,67 |
| 0.3 | Прототипирование и составление SRS | 32 | 80 | 48 | 50,67 | 8,00 |
| 0.4 | Выбор стека технологий | 8 | 24 | 16 | 16,00 | 2,67 |
| 0.5 | Дизайн | 80 | 200 | 120 | 126,67 | 20,00 |
| 1 | **Управление контентом и событиями** | | | | | |
| 1.1 | Категоризация событий | 24 | 56 | 40 | 40,00 | 5,33 |
| 1.2 | Привязка к геолокации | 40 | 96 | 64 | 65,33 | 9,33 |
| 1.3 | Управление расписанием | 48 | 120 | 80 | 81,33 | 12,00 |
| 1.4 | Модерация и публикация | 32 | 80 | 48 | 50,67 | 8,00 |
| 2 | **Пользовательский интерфейс и навигация** | | | | | |
| 2.1 | Выбор города | 16 | 40 | 24 | 25,33 | 4,00 |
| 2.2 | Календарь событий | 24 | 64 | 40 | 41,33 | 6,67 |
| 2.3 | Система фильтров и поиска | 64 | 160 | 96 | 101,33 | 16,00 |
| 2.4 | Адаптивный дизайн | 80 | 200 | 120 | 126,67 | 20,00 |
| 3 | **Пользовательские аккаунты и персонализация** | | | | | |
| 3.1 | Регистрация и аутентификация | 40 | 96 | 64 | 65,33 | 9,33 |
| 3.2 | Личный кабинет | 48 | 112 | 72 | 74,67 | 10,67 |
| 3.3 | Обратная связь | 32 | 80 | 48 | 50,67 | 8,00 |
| 4 | **Работа с местами (локациями)** | | | | | |
| 4.1 | Карточка места | 40 | 96 | 64 | 65,33 | 9,33 |
| 4.2 | Привязка событий к местам | 16 | 40 | 24 | 25,33 | 4,00 |
| 5 | **Дополнительный контент и функции** | | | | | |
| 5.1 | Новостной раздел и статьи | 48 | 120 | 80 | 81,33 | 12,00 |
| 5.2 | Система тегов (меток) | 24 | 64 | 40 | 41,33 | 6,67 |
| 5.3 | Рейтинги и просмотры | 24 | 56 | 40 | 40,00 | 5,33 |
| 5.4 | Интерактивные тесты и викторины | 32 | 80 | 48 | 50,67 | 8,00 |
| 5.5 | Редакционные подборки и гиды | 24 | 64 | 40 | 41,33 | 6,67 |
| 6 | **Интеграции и монетизация** | | | | | |
| 6.1 | Интеграция с партнерами | 64 | 144 | 96 | 98,67 | 13,33 |
| 6.2 | Размещение рекламы | 40 | 96 | 64 | 65,33 | 9,33 |
| 6.3 | Подписка на рассылки | 32 | 80 | 48 | 50,67 | 8,00 |
| 7 | **Административная панель** | | | | | |
| 7.1 | Управление контентом | 80 | 200 | 128 | 132,00 | 20,00 |
| 7.2 | Модерация пользовательского контента | 24 | 64 | 40 | 41,33 | 6,67 |
| 8 | **Технические требования** | | | | | |
| 8.1 | Производительность | 48 | 120 | 80 | 81,33 | 12,00 |
| 8.2 | Безопасность | 40 | 96 | 64 | 65,33 | 9,33 |
| 9 | **Тестирование и обеспечение качества** | | | | | |
| 9.1 | Юнит-тестирование | 24 | 56 | 40 | 40,00 | 5,33 |
| 9.2 | Функциональное и интеграционное тестирование | 40 | 96 | 64 | 65,33 | 9,33 |
| 9.3 | Тестирование UI/UX и кросс-браузерность | 24 | 64 | 40 | 41,33 | 6,67 |
| 9.4 | Приемочное тестирование (UAT) | 16 | 32 | 24 | 24,00 | 2,67 |
| 9.5 | Нагрузочное и стресс-тестирование | 24 | 64 | 40 | 41,33 | 6,67 |
| 10 | **Релиз** | | | | | |
| 10.1 | Документирование | 90 | 160 | 100 | 108,33 | 11,67 |
| 10.2 | Альфа- и бета-тестирование | 16 | 40 | 24 | 25,33 | 4,00 |
| 10.3 | A/B-тестирование | 32 | 80 | 48 | 50,67 | 8,00 |
| 10.4 | Поддержка | 70 | 130 | 115 | 110,00 | 10,00 |
| 10.5 | Развёртывание | 160 | 200 | 170 | 173,33 | 6,67 |
| 10.6 | Аналитика | 40 | 80 | 66 | 64,00 | 6,67 |
| Итого: | | 1672 | 3802 | 2515 | 2 589,00 | 62,04 |

Суммарная трудоемкость проекта может быть рассчитана по формуле:

А среднеквадратичное отклонение для оценки суммарной трудоемкости будет составлять:

Суммарная трудоемкость проекта с вероятность 95%:

.

## **Сетевая диаграмма**

Нарисуем сетевую диаграмму взаимосвязи работ, как показано на рисунке 1:

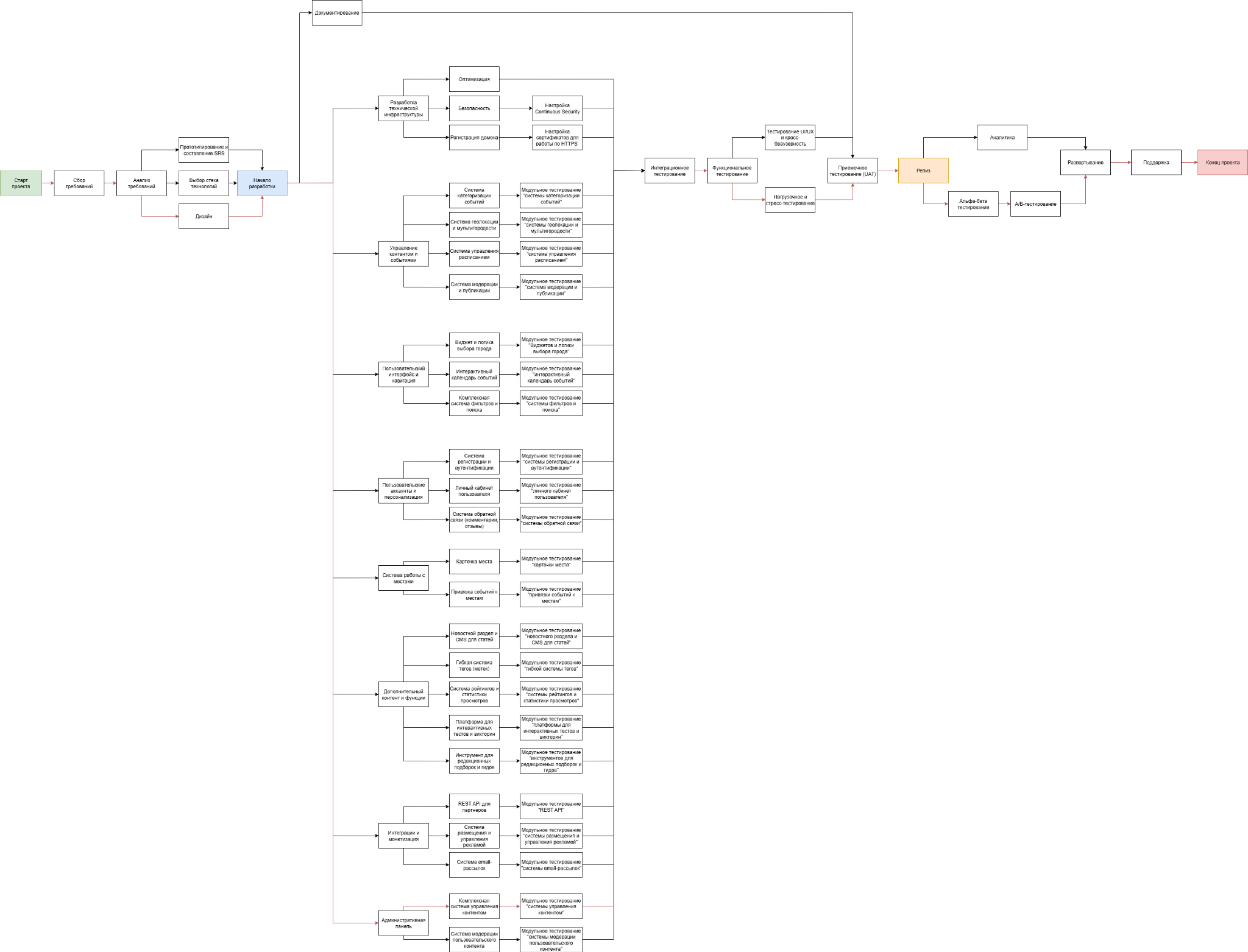


Рисунок 1 – Сетевая диаграмма взаимосвязи работ

Полная диаграмма: [Сетевая диаграмма.drawio](https://drive.google.com/file/d/1bXWiJkNsACg4k996eLoamzyFhQssXJrI/view?usp=drive_link).

## 

## **Метод критического пути**

Как видно на сетевой диаграмме, критический путь – это Старт проекта → Сбор требований → Анализ требований → Дизайн → Начало разработки → Административная панель → Комплексная система управления контентом → Модульное тестирование "системы управления контентом" → Интеграционное тестирование → Функциональное тестирование → Нагрузочное и стресс-тестирование → Приемочное тестирование (UAT) → Релиз → Аналитика → Развертывание → Поддержка → Конец проекта:

* Критический путь: 838 ч./ч.
* СКО критического пути: 34,3 ч./ч.
* Длинный путь: 2 515 ч./ч.

**Оценим время разработки по критическому пути:**

Рассчитаем минимальную продолжительность разработки:

* Рабочий день: 12 часов.
* Рабочая неделя: 6 дней = 72 часов на человека.

838 / 12 ≈ 69 дня (~2.3 месяцев)

**Оценим срок выполнения проекта:**

Рекомендуемый состав команды:

| **Роль** | **Количество** | **Обоснование** |
| --- | --- | --- |
| Frontend-разработчик | 2 | Сложный адаптивный UI, интерактивные компоненты. |
| Backend-разработчик | 2 | Ядро: события, места, пользователи, расписание, модерация, API для партнёров, email-рассылки. |
| DevOps / SRE-инженер | 1 | Настройка CI/CD, HTTPS, мониторинг, развёртывание. |
| QA-инженер | 1 | Юнит-, интеграционное, функциональное, нагрузочное тестирование. |
| UX/UI-дизайнер | 1 | Необходим для задачи “Дизайн”. |
| Технический писатель / PM | 1 | Документирование, координация, сбор требований. Совмещен с тимлидом. |

Оценка по критическому пути минимальный срок:

При команде из 5 человек (дизайнер и DevOps — на частичной/этапной занятости), работающих параллельно:

Эффективность параллелизма ≈ 70–80% (из-за коммуникаций, зависимостей, интеграций).

Реалистичная оценка с параллелизмом: (25.33 + 24.00 + max(126.67, 16.00, 50.67/3) + max(108.33, 1456/2 + 40/2 + 65.33 + max(41.33, 41.33) + 24.00) + max(64.00, 25.33 + 50.67) + 173.33 + 110.00) \* 0.75 = **1060.4925 часов** (≈ 89 рабочих дней ≈ 15 рабочих недель ≈ **3.68 месяца**).

При хорошей организации (Agile, 2-недельные спринты, раннее тестирование) можно уложиться в 12–14 недель.

## 

## **Оценка методом функциональных точек**

Метод функциональных точек (Function Point Analysis, FPA) позволяет оценить размер программного продукта с точки зрения функциональности, предоставляемой пользователю, независимо от используемых технологий

При анализе методом функциональных точек надо выполнить следующую последовательность шагов:

1. Определение типа оценки.
2. Определение области оценки и границ продукта.
3. Подсчет функциональных точек, связанных с данными.
4. Подсчет функциональных точек, связанных с транзакциями.
5. Определение суммарного количества не выровненных функциональных точек (UFP).
6. Определение значения фактора выравнивания (FAV).
7. Расчет количества выровненных функциональных точек (AFP).

### **1. Определение типа оценки**

Оценивается сложность уже существующего продукта — информационно-развлекательного портала KudaGo.

### **2. Определение области оценки и границ продукта**

Границы системы определены следующим образом:

* Внутри системы: все данные и функции, управляющие афишей событий, местами проведения, поиск и фильтрацию, систему комментариев, пользователями, контентом (статьи, тесты, подборки), админ-панель и интеграциями.
* Внешние сущности: пользователи (гости и зарегистрированные), платёжные системы (для покупки билетов), партнёрские API (для интеграции событий), сервисы аналитики и картографии (Google/Yandex Maps).

### **3. Расчет функциональных точек, связанных с данными**

Согласно методике, для определения сложности файлов используются параметры DET (Data Element Type) и RET (Record Element Type). Внутренние логические файлы (ILF) — данные, управляемые системой:

| № | Название ILF | RET | DET | Сложность | UFP |
| --- | --- | --- | --- | --- | --- |
| 1 | Событие | 3 (основное, расписание, категории) | 25 (название, описание, дата, время, цена, возраст, теги, фото, гео и др.) | High | 15 |
| 2 | Место проведения | 2 (адрес, контакты) | 12 | Average | 10 |
| 3 | Пользователь | 2 (профиль, настройки) | 10 | Average | 10 |
| 4 | Избранное | 1 | 5 | Low | 7 |
| 5 | Комментарий/отзыв | 1 | 6 | Low | 7 |
| 6 | Статья/новость | 2 (текст, метаданные) | 10 | Average | 10 |
| 7 | Город | 1 | 5 | Low | 7 |
| 8 | Категория/тег | 1 | 4 | Low | 7 |
| 9 | Подборка/гид | 2 | 12 | Average | 10 |
| 10 | Тест/викторина | 3 | 20 | High | 15 |
| Итого ILF |  |  |  |  | 91 |

Внешние интерфейсные файлы (EIF) — данные, на которые система ссылается, но не управляет:

| № | Название EIF | RET | DET | Сложность | UFP |
| --- | --- | --- | --- | --- | --- |
| 1 | Партнёрские билетные системы | 1 | 8 | Low | 5 |
| 2 | Сервисы геокодирования (карты) | 1 | 6 | Low | 5 |
| 3 | Соцсети (для авторизации/шаринга) | 1 | 4 | Low | 5 |
| Итого EIF |  |  |  |  | 15 |

### **4. Расчет функциональных точек транзакций**

Транзакции классифицируются на входные (EI), выходные (EO) и запросы (EQ).

| № | Название транзакции | Тип | FTR | DET | Сложность | UFP |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Поиск событий | EQ | 3 | 8 | Average | 4 |
| 2 | Фильтрация (категория, дата, цена) | EQ | 4 | 12 | High | 6 |
| 3 | Регистрация/вход | EI | 2 | 6 | Average | 4 |
| 4 | Добавление в избранное | EI | 2 | 3 | Low | 3 |
| 5 | Публикация события (админ) | EI | 4 | 15 | High | 6 |
| 6 | Просмотр карточки события | EO | 3 | 20 | High | 7 |
| 7 | Подписка на рассылку | EI | 2 | 4 | Low | 3 |
| 8 | Комментирование | EI | 2 | 5 | Low | 3 |
| 9 | Управление подборкой (админ) | EI | 3 | 10 | Average | 4 |
| 10 | Генерация RSS/экспорт | EO | 2 | 8 | Average | 5 |
| Итого | | | | | | 45 |

### **5. Определение суммы невыровненных ФТ (UFP)**

Суммируем все компоненты:

UFP = ILF + EIF + Транзакции = 91 + 15 + 45 = **151**.

### **6. Определение фактора выравнивания (VAF)**

Фактор выравнивания учитывает 14 общих системных характеристик (GSC). Для KudaGo, как мультигородского агрегатора с высокой интерактивностью, оценки DI (Degree of Influence) следующие:

| № | Параметр | DI |
| --- | --- | --- |
| 1 | Обмен данными | 4 |
| 2 | Распределённая обработка | 3 |
| 3 | Производительность | 4 |
| 4 | Аппаратные ограничения | 1 |
| 5 | Транзакционная нагрузка | 4 |
| 6 | Взаимодействие с пользователем | 5 |
| 7 | Эргономика | 4 |
| 8 | Изменение данных | 3 |
| 9 | Сложность обработки | 4 |
| 10 | Повторное использование | 2 |
| 11 | Удобство установки | 2 |
| 12 | Администрирование | 3 |
| 13 | Портируемость | 3 |
| 14 | Гибкость | 3 |
| TDI (ΣDI) |  | 45 |

TDI = ΣDI = 45

Рассчитываем VAF по формуле:

VAF = 0.65 + (TDI × 0.01) = 0.65 + 0.45 = 1.10

### **7. Расчет количества выровненных функциональных точек (AFP)**

AFP = UFP × VAF = 151 × 1.10 ≈ 166 FP

## **Расчет трудоемкости методом COCOMO II**

Метод COCOMO II требует перевода функциональных точек в строки кода (SLOC), а затем применения регрессионной модели для оценки трудоемкости разработки программного продукта на основе его размера в тысячах строк исходного кода и ряда корректирующих факторов, отражающих сложность проекта, квалификацию команды и особенности окружения

### **1. Конвертация функциональных точек в SLOC**

Для современного веб-стека (предположим Python/Django для бэкенда и React/TypeScript для фронтенда) используем коэффициенты из отраслевой статистики:

* Backend (Python): ~45 SLOC/FP
* Frontend (React): ~55 SLOC/FP
* Средневзвешенное значение: 50 SLOC/FP

Общий объём кода: SLOC = 166 FP × 50 = 8 300 строк = 8.3 KSLOC

### **2. Расчет факторов масштаба (SF)**

Факторы масштаба определяют нелинейность зависимости трудоёмкости от размера проекта. Для портала KudaGo выбраны следующие уровни:

| Фактор | Уровень | Значение |
| --- | --- | --- |
| PREC (Прецедентность) | Nominal | 3.72 |
| FLEX (Гибкость процесса) | High | 2.48 |
| RESL (Архитектура и риски) | High | 2.83 |
| TEAM (Сработанность команды) | Nominal | 3.29 |
| PMAT (Зрелость процессов) | Nominal | 4.68 |
| ΣSF |  | 17.00 |

Показатель масштаба E рассчитывается по формуле:

E = 0.91 + 0.01 × ΣSF = 0.91 + 0.17 = 1.08

### **3. Расчет множителей усилия (EM)**

Множители усилия отражают влияние характеристик проекта и команды на трудоемкость. Для KudaGo выбраны следующие значения на основе анализа функциональности и архитектуры:

| Фактор | Уровень | Множитель |
| --- | --- | --- |
| Требуемая надёжность (RELY) | High | 1.10 |
| Размер БД (DATA) | High | 1.14 |
| Сложность продукта (CPLX) | High | 1.17 |
| Ограничения времени выполнения (TIME) | Nominal | 1.00 |
| Ограничения памяти (STOR) | Nominal | 1.00 |
| Изменчивость окружения (PVOL) | Nominal | 1.00 |
| Время восстановления (RUSE) | Low | 0.95 |
| Аналитическая способность (ACAP) | High | 0.85 |
| Опыт в приложениях (APEX) | Nominal | 1.00 |
| Опыт в платформе (PLEX) | Nominal | 1.00 |
| Опыт в языке (LTEX) | High | 0.91 |
| Использование методов (PCON) | High | 0.87 |
| Использование инструментов (TOOL) | High | 0.90 |
| Распределённая разработка (SITE) | Nominal | 1.00 |
| Требуемое соблюдение графика (SCED) | Nominal | 1.00 |

Произведение всех множителей:

EM = 1.10 × 1.14 × 1.17 × 0.95 × 0.85 × 0.91 × 0.87 × 0.90 ≈ 0.89

### **4. Итоговый расчёт трудоёмкости**

Базовая формула COCOMO II для предварительной оценки:

PM = A × (KSLOC)^E × EM

Где:

* PM - человеко-месяцы
* A = 2.94 (для современных языков)
* E = 1.08 (произведение множителей усилия)
* EM = 0.89

Подставляем значения:

PM = 2.94 × (8.3)^1.08 × 0.89 ≈ 2.94 × 9.05 × 0.89 ≈ 23.7 чел/мес

Переводим в человеко-часы (160 часов в месяце):

Трудоемкость = 23.7 × 160 ≈ **3 792 чел/час**

### **5. Расчет сроков разработки**

Формула длительности (TDEV) по COCOMO II:

TDEV = 3 × (PM)^(0.33+0.002×(E)),

где:

* PM = 23.7 чел×мес
* E = 1.08 (показатель масштаба)

TDEV=3×23.7^(0.33+0.002×1.08) = 3×23.7^0.33216 ≈ 3×2.89 ≈ 8.67 месяцев

Срок разработки **≈ 8.7 месяцев**.

### **6. Расчет команды**

Средний размер команды = PM / TDEV = 23.7 / 8.67 ≈ **2.73 человека**.

Округляем с учётом реалистичной загрузки и необходимости покрытия всех ролей (backend, frontend, тестирование, интеграции):

Рекомендуемая команда: 3 человека

* 1 backend-разработчик
* 1 frontend-разработчик
* 1 full-stack / QA / DevOps (на частичной занятости или в ротации)

## **Итоговая оценка**

| Метрика | Значение |
| --- | --- |
| Функциональные точки | 166 FP |
| Оценочный объем кода | 8,300 SLOC |
| Трудоемкость | 23.7 чел·мес  (≈ 3 792 чел·часа) |
| Срок разработки | 8.7 месяцев |
| Рекомендуемая команда | 3 человека |

*Примечание:* Оценка не включает время на проектирование архитектуры, DevOps-работы, создание дизайна, интенсивное тестирование и управление проектом. Эти этапы могут увеличить общую трудоемкость на 25–40%.

## **Метод оценки вариантов использования (Use Case Points)**

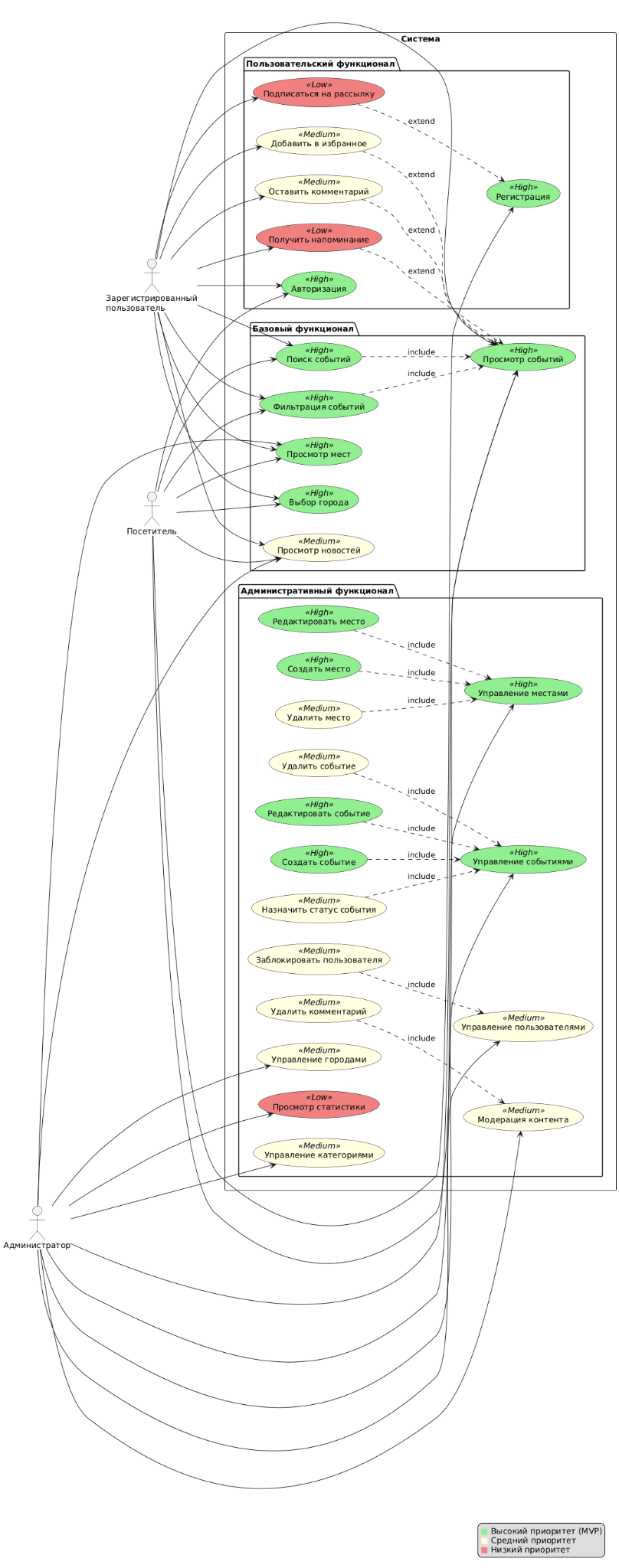


Рисунок 2 – Use-Case диаграмма

## **Оценка веса прецедентов (UUCW)**

| Сложность | Вес | Количество | Затраты |
| --- | --- | --- | --- |
| Low | 5 | 3 | 15 |
| Medium | 10 | 12 | 120 |
| High | 15 | 13 | 195 |
| Итого |  | 28 | 330 |

UUCW = 330

## **Оценка веса акторов (UAW)**

| Сложность | Вес | Актор | Количество | Затраты |
| --- | --- | --- | --- | --- |
| Low | 1 | Платежная система | 1 | 1 |
| Medium | 2 | Партнерская система | 1 | 2 |
| High | 3 | Редактор/Администратор | 1 | 3 |
| Итого |  |  | 3 | 6 |

UAW = 6

## **Оценка технических факторов (TCF)**

| № | Фактор | Вес (W) | Значение (F) | Значение с учетом веса |
| --- | --- | --- | --- | --- |
| T1 | Распределенная система | 2 | 3 | 6 |
| T2 | Высокая пропускная способность | 1 | 4 | 4 |
| T3 | Работа пользователей онлайн | 1 | 5 | 5 |
| T4 | Сложная обработка данных | 1 | 4 | 4 |
| T5 | Повторное использование кода | 1 | 3 | 3 |
| T6 | Простота установки | 0,5 | 2 | 1 |
| T7 | Простота использования | 0,5 | 5 | 2,5 |
| T8 | Переносимость | 2 | 3 | 6 |
| T9 | Простота внесения изменений | 1 | 3 | 3 |
| T10 | Параллелизм | 1 | 3 | 3 |
| T11 | Требования безопасности | 1 | 4 | 4 |
| T12 | Доступ внешних пользователей | 1 | 5 | 5 |
| T13 | Требования к обучению | 1 | 2 | 2 |
| Итого |  |  |  | 48,5 |

TCF = 0.6 + (48.5/100) = 1.085

## **Оценка факторов окружения (ECF)**

| Фактор | Вес | Влияние | Затраты |
| --- | --- | --- | --- |
| Знакомство с моделью проекта | 1.5 | 3 | 4.5 |
| Частичная занятость | -1 | 1 | -1 |
| Аналитические способности команды | 0.5 | 4 | 2 |
| Опыт применения подобных систем | 0.5 | 3 | 1.5 |
| Опыт в веб-разработке | 1 | 4 | 4 |
| Мотивация | 1 | 3 | 3 |
| Сложные технологии | -1 | 2 | -2 |
| Неизменность требований | 2 | 2 | 4 |
| Итого |  |  | 16 |

ECF = 1.4 + (-0.03 × 16) = 0.92

## **Вычисление UCP**

UCP = (UUCW + UAW) × TCF × ECF  
UCP = (330 + 6) × 1.085 × 0.92 = 336 × 1.085 × 0.92 = 335.3952

## 

## **Подсчет фактора продуктивности (PF)**

Выбранный проект для сравнения: курсовая работа «ТЕМАТИЧЕСКИЙ АУДИОПЛЕЕР», как показано на рисунке 3:

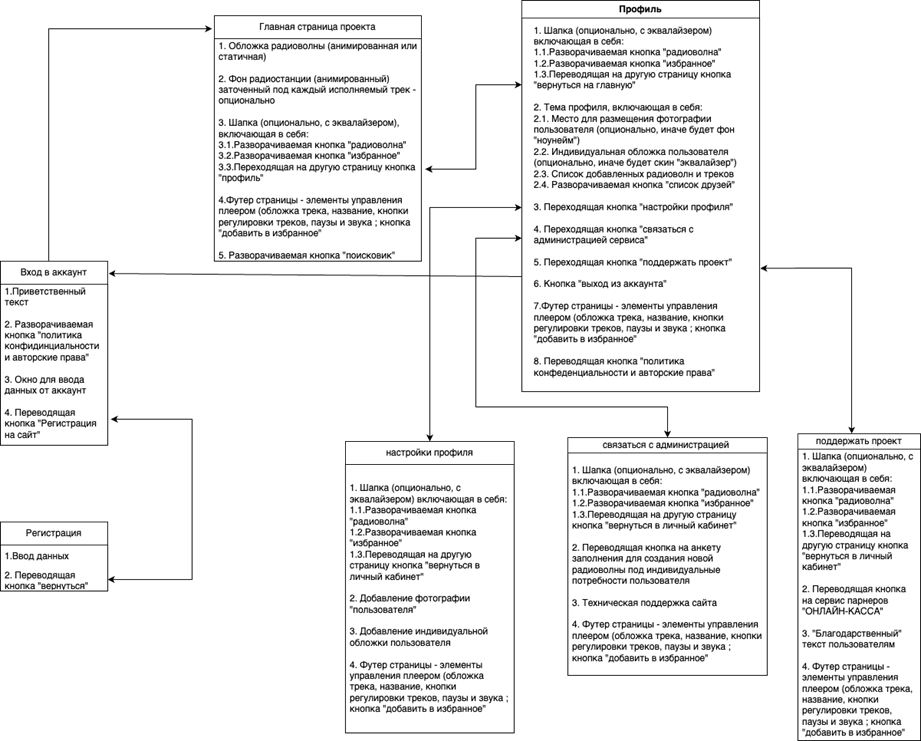


Рисунок 3 – USER-FLOW диаграмма проекта

Вес прецедентов (UUCW, Unadjusted Use Case Weight):

| Сложность | Вес (UUCW) | Количество прецедентов | Затраты = Вес × Кол-во |
| --- | --- | --- | --- |
| low | 5 | 3 | 15 |
| medium | 10 | 4 | 40 |
| high | 15 | 0 | 9 |
|  | | | **55** |

Таблица весов UUCW по сложности

Вес актера без поправок (Unadjusted Actor Weight, UAW):

| Сложность | Вес (UAW) | Кол-во | Затраты = Вес×Кол-во |
| --- | --- | --- | --- |
| Low (Simple) |  |  |  |
| Medium (Average) | 2 | 1 | 1 |
| High (Complex) |  |  |  |
|  | | | **2** |

Оценка технических факторы (TF, TCF):

| **№** | **Фактор** | **Вес** | **Оценка (0–5)** |
| --- | --- | --- | --- |
| T1 | Распределённость системы | 2.0 | 2 |
| T2 | Производительность/требования по времени | 1.0 | 3 |
| T3 | Сложность обработки данных | 1.0 | 2 |
| T4 | Сложность пользовательского ввода | 0.5 | 3 |
| T5 | Частота транзакций | 1.0 | 2 |
| T6 | Онлайн-обновления | 0.5 | 2 |
| T7 | Эффективность обработки | 0.5 | 2 |
| T8 | Переиспользуемость | 1.0 | 2 |
| T9 | Сложность преобразований | 1.0 | 2 |
| T10 | Удобство инсталляции | 0.5 | 3 |
| T11 | Удобство эксплуатации | 0.5 | 3 |
| T12 | Множественные сайты / разные площадки | 2.0 | 1 |
| T13 | Лёгкость изменений | 1.0 | 3 |
|  | | | **26** |
| TCF = 0.6 + 0.02 \* TF | | | **0.86** |

Оценка факторов окружения (EF, ECF):

| **№** | **Фактор** | **Вес** | **Оценка (0–5)** |
| --- | --- | --- | --- |
| E1 | Опыт команды с подобными проектами | 1.5 | 3 |
| E2 | Опыт работы с предметной областью | 0.5 | 2 |
| E3 | Опыт работы с технологиями (JS/React) | 1.0 | 4 |
| E4 | Способности аналитиков | 0.5 | 3 |
| E5 | Мотивация команды | 1.0 | 4 |
| E6 | Стабильность требований | 2.0 | 3 |
| E7 | Использование специалистов-контракторов | –1.0 | 1 |
| E8 | Сложность и распределённость команды | –1.0 | 1 |
|  | | | 19 |
|  | | | 0,83 |

Итоговый расчет Use Case Points курсовой работы «тематический аудиоплеер»:

* UCP прошлого проекта = 40.7 чч
* Затраченное время = 200 чч
* PF = 200 / 40.7 = 4.91

## **Применение фактора продуктивности (PF)**

Общее время разработки = UCP × PF + CONST

Где CONST включает:

* Настройка инфраструктуры: 40 чч
* DevOps работы: 30 чч
* Создание дизайна: 60 чч
* Тестирование: 50 чч
* Управление проектом: 40 чч
* Итого CONST = 220 чч

Общая оценка трудозатрат:  
E = 335.3952 × 4.91 + 220 = 1646.79043 + 220 = 1866.79043 чч

## 

## **Анализ результатов**

| Метод оценки | Затраты (чч) |
| --- | --- |
| Наивный | 2515 |
| PERT | 1060 |
| МФТ + СОСОМО-II | 3792 |
| UCP | 1866 |

Наивный метод (2515) и метод PERT (1060) являются самыми неточными, так как это упрощенные подходы, которые не учитывают все функции, риски, сложность интеграции, непредвиденные работы и другие факторы, неизбежно возникающие в реальных проектах. Наши оценки можно считать не очень релевантными, так как у нас нет представления, насколько трудоемкий процесс разработки большого продукта.

Метод функциональных точек + COCOMO II (3792) даёт в 2 раза больший результат, чем UCP (1866). Это связано с тем, что COCOMO II учитывает не только функциональный объем системы, но и такие параметры, как сложность архитектуры, требования к надёжности, размер базы данных и квалификацию команды. В то же время метод UCP сильно зависит от выбранного коэффициента продуктивности (PF), который в нашем случае был рассчитан на основе небольшого учебного проекта и может быть занижен для enterprise-уровня. Кроме того, UCP фокусируется преимущественно на пользовательских сценариях и может недооценивать сложность backend-логики, административной панели и интеграций — ключевых компонентов KudaGo. Таким образом, наиболее объективной представляется оценка по методу МФТ + COCOMO II, поскольку она опирается на отраслевую статистику и учитывает как функциональную сложность, так и характеристики процесса разработки.

Проект является достаточно сложным и требует тщательного планирования. Разброс в оценках (более 2000 единиц) подчеркивает важность использования нескольких методов оценки для формирования сбалансированного понимания сроков.

## **Вывод**

В результате выполнения лабораторной работы мы сравнили методы оценки затрат человеко-часов на разработку заданного по варианту продукта, а также же разобрались, какие величины учитывают эти методы и какие недостатки у них есть. Были применены четыре различных метода оценки трудоёмкости разработки веб-проекта KudaGo: наивный метод, метод PERT, метод функциональных точек в сочетании с моделью COCOMO II, а также метод оценки по вариантам использования (Use Case Points). Полученные результаты существенно различаются: от 1060 человеко-часов по методу PERT до 3792 человеко-часов по COCOMO II. Такой разброс объясняется различиями в подходах — от субъективных экспертных оценок до формализованных моделей, учитывающих архитектурную сложность, характеристики команды и технические ограничения. Наиболее реалистичной представляется оценка, полученная с помощью метода функциональных точек и модели COCOMO II, поскольку она основана на отраслевых нормативах и учитывает не только функциональные требования, но и нефункциональные аспекты проекта: надёжность, производительность, безопасность и опыт команды. В то же время метод UCP, хоть и дал меньшую оценку, может быть полезен на ранних этапах планирования, особенно при наличии релевантного опыта и корректно подобранного фактора продуктивности. Методы наивной оценки и PERT оказались наименее точными в данном контексте, так как сильно зависят от субъективного восприятия объёма работ и не учитывают сложность интеграций, административной части и инфраструктурных задач. Тем не менее, они позволяют быстро сформировать первоначальное представление о масштабе проекта.Таким образом, для повышения точности оценки рекомендуется комбинировать несколько методов, использовать исторические данные по схожим проектам и учитывать как функциональную, так и техническую сложность системы. Это особенно важно при работе над крупными, многокомпонентными продуктами, подобными KudaGo, где значительную долю усилий составляют не только реализация пользовательских сценариев, но и поддержка масштабируемости, безопасности и удобства администрирования.